IMPUTAÇÃO MULTIVARIADA: UMA ABORDAGEM EM CASCATA

Claudia Ferlin

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Prof. Luis Alfredo Vidal de Carvalho, D. Sc.

Prof. Geraldo Bonorino Xexéo, D. Sc.

Prof. Nelson Francisco Favilla Ebecken, D. Sc.

Profª. Flavia Maria Santoro, D. Sc.

Prof. Basílio de Bragança Pereira, Ph. D.

RIO DE JANEIRO, RJ – BRASIL

AGOSTO DE 2008

FERLIN, CLAUDIA

Imputação Multivariada: Uma Abordagem em Cascata [Rio de Janeiro] 2008

XI, 244 p. 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia de Sistemas e Computação, 2008)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Pré-Processamento de Dados 2. Imputação em Cascata 3. Imputação Multivariada

I. COPPE/UFRJ II. Título (série)

.

Aos meus queridos pais Dirceu e Dorothea,

meu carinho e minha gratidão em reconhecimento ao amor incondicional e pelo exemplo de vida que guiam meus passos.

Aos meus amados Luis Yale, Pablo e Sofia,

meu amor e gratidão em reconhecimento ao apoio, compreensão, alegria e carinho fundamentais à elaboração deste trabalho.

AGRADECIMENTOS

Inicialmente eu gostaria de fazer uma breve homenagem a minha incomparável mãe (in memoriam) que se foi muito cedo, mas que sempre está na minha lembrança. Sem dúvidas, muito deste trabalho se deve ao seu carinho, seu apoio, seu exemplo e seus ensinamentos, destacando sempre a importância da educação.

Meus afetivos agradecimentos e amor ao grande companheiro de todas as horas, Luís Yale, que não só abdicou do tempo de convívio mas também o multiplicou, substituindo-me em todas as tarefas do dia a dia em prol da realização deste trabalho encorajando-me sempre a prosseguir. Com a mesma intensidade agradeço aos meus filhos Pablo e Sofia, ela sempre presente, que souberam esperar, incentivar e transmitir seu amor de um modo abnegado. A presença de vocês é responsável pela minha saúde afetiva.

Ao meu pai, pela sólida formação e apoio contínuo, que me permitiram dar continuidade nos estudos até a chegada a este doutorado, meus eternos agradecimentos. Agradeço, também, a meus irmãos, Lisette e Beclar, pela torcida, incentivo e carinho que nos momentos difíceis foram fundamentais e às minhas sobrinhas Mariana e Elisa, que também na torcida, me alegraram e apoiaram. Ao meu sogro, Luis Carlos, de forma silenciosa, agradeço o apoio também espiritual.

Em particular, gostaria de agradecer ao professor Luís Alfredo Vidal de Castro, que efetivamente é um orientador, não só por sua capacidade técnica, mas por ser amigo e humano, propiciando oportunidades únicas aos seus alunos que viabilizam a conclusão de suas metas. Somente devido sua grande compreensão esta tese pode ser finalizada.

À banca avaliadora deste trabalho, os professores Basílio Bragança, Flávia Santoro, Geraldo Xexéo e Nelson Ebecken, meus sinceros agradecimentos, não só pela paciência e compreensão na elasticidade de alguns prazos, mas principalmente pela grande oportunidade recebida de contar com suas contribuições e reflexões

A Jorge de Abreu Soares, insubstituível e grande responsável pela existência desta tese, meu grande amigo, co-orientador, que me deu a honra de ser sua comadre,

muito obrigada por sua amizade, carinho, sensibilidade, contribuições, garra e suporte em todas as horas.

Do mesmo modo agradeço ao também co-orientador Ronaldo Goldschmidt, cujos vastos conhecimentos foram fonte de sugestões decisivas que enriqueceram meu trabalho, pela presença contínua e por ser um amigo querido, calmo, dedicado, competente, cujo tempo usufruí em demasia.

Ao meu amigo Rafael Castaneda, cuja participação foi muito além da esperada, a quem considero como um filho, obrigada por suas noites, seu suporte, suas contribuições, sua delicadeza e paciência em atender e providenciar tantas solicitações.

Os três são responsáveis por calorosas discussões teóricas e metodológicas (chamando-me muitas vezes à razão) e por bons momentos de descontração.

Aos meus amigos e colegas, Carmen Lúcia Asp de Queiroz, grande especialista em diagramas de atividades, André Avelino Sobral e Fernando Pina, efusivos agradecimentos pela importante ajuda e cobertura. Sem o apoio de vocês e suas horas extras certamente a realização deste trabalho teria sido muito mais pesada. À Isabel Fernandes, sempre amiga, agradeço o apoio espiritual e as palavras de carinho.

Agradeço também ao corpo administrativo do Programa de Engenharia de Sistemas e Computação da COPPE, as funcionárias Solange, Cláudia, Lúcia e Sônia pela delicadeza e destreza no atendimento e esclarecimentos.

Aos meus amigos da PUC-Rio e da UniverCidade agradeço as palavras de incentivo e o afeto.

Sem dúvidas, a elaboração desta tese foi um produto coletivo. Até o stress foi compartilhado com várias pessoas . A todos estes grandes amigos, no sentido amplo da palavra, registro minha gratidão.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D. Sc.)

IMPUTAÇÃO MULTIVARIADA: UMA ABORDAGEM EM CASCATA

Claudia Ferlin

Agosto/2008

Orientador: Luis Alfredo Vidal de Carvalho

Programa: Engenharia de Sistemas e Computação

As aplicações atuais e a evolução tecnológica vêm promovendo a produção e o armazenamento de um grande volume de dados. Este cenário faz com que a existência de valores ausentes em registros das bases de dados inevitavelmente aumente. Estas lacunas prejudicam a análise dos dados, além de dificultar ou mesmo inviabilizar o processo de abstração de conhecimento a partir deles.

Desta forma, este trabalho tem por objetivo propor uma abordagem em cascata para tratar a imputação multivariada com reutilização dos valores imputados, bem como avaliar o impacto da ordem no processo de imputação e da reutilização dos valores imputados na correlação original da base de dados. Nesta abordagem o processo de imputação é precedido pela tarefa de agrupamento usando como critério a morfologia da ausência. Os casos incompletos são distribuídos em grupos considerando como critério de pertinência o conceito de morfologia da ausência neles existentes. A morfologia de ausência é um conceito aqui proposto para descrever a distribuição de valores presentes e ausentes nos atributos de um conjunto de casos. Portanto, os grupos são formados por casos similares quanto à forma de distribuição de seus atributos não preenchidos.

Os resultados experimentais mostram melhora da qualidade dos dados sugeridos pela imputação sequencial em cascata quando comparada com a imputação sequencial com e sem reutilização dos valores imputados.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D. Sc.)

MULTIVARIATE IMPUTATION: A CASCADE APPROACH

Claudia Ferlin

August/2008

Advisor: Luis Alfredo Vidal de Carvalho

Department: Computer and Systems Engineering

Nowadays applications and technological evolution have caused the production and storage of huge volumes of data. This scenario facilitated the increased occurrence of missing values in data sets. Missing data is harmful for statistical analysis, complicating or even not allowing the process of extracting knowledge from these non preprocessed data.

Hence, this work aims to propose a cascade approach to the problem of multivariate imputation of missing values. Introduce the idea of clustering using the morphology of the missingness before the imputation and analyze the effects of the order in sequential imputation as well as the correlation in data sets.

Experimental results illustrate the comparision between this approach and sequential imputation with and without reuse. They indicated that cascade imputation achieves quality improvement of imputed data.

Índice

CAPÍTULO 1

INTRODUÇÃO

1.1.Considerações Iniciais

A automatização de processos unida ao avanço tecnológico dos computadores têm como uma de suas conseqüências, o armazenamento de uma imensa quantidade de dados em meios digitais. Estes dados "escondem" um verdadeiro tesouro quando devidamente tratados e interpretados: o conhecimento. A descoberta de conhecimento em bases de dados (*Knowledge Discovery in Databases* ou KDD) visa buscar padrões desconhecidos existentes nas massas de dados e envolve diversas áreas de conhecimento, tais como a Estatística, Inteligência Artificial, Aprendizado de Máquina, Banco de Dados, Reconhecimento de Padrões, Armazém de Dados (*Data Warehousing*), Visualização de Dados entre outras. Este "garimpo" de informações valiosas escondidas nos registros dá suporte à gestão do conhecimento permitindo identificar tendências, apoiar decisões, prever cenários futuros, descobrir padrões desconhecidos, realizar associações, formular teorias, entre tantas outras contribuições para o meio comercial e científico. O macro-processo para que este conhecimento, a princípio invisível, possa emergir, envolve três grandes etapas. Inicialmente, há a preparação dos dados (1ª etapa) que serão explorados por meio de diferentes técnicas (2ª etapa) para posterior construções de modelos ou padrões que devem ser validados e/ou verificados (3ª etapa). Uma vez que os dados é a fonte primária do processo deve-se tomar cuidado com eles, lembrando o famoso axioma da computação: "garbage in garbage out". Se dados inválidos forem manipulados as saídas também serão inválidas levando a conclusões inválidas.

A segunda etapa, onde os dados são minerados, pressupõe uma base de dados sem erros ou ruídos, limpa, com todos os registros completos e, preferencialmente bem balanceados. No entanto, em bases reais, este é um contexto quase utópico. Os dados, em geral, em algum grau e de algum modo, estão corrompidos, seja pela ausência de valores, seja pela presença de valores incorretos. Os bancos de dados modelam o nosso conhecimento do mundo real e portanto são repletos de incertezas com as quais deve-se lidar (MOTRO 1995).

Sendo assim, a primeira etapa da Descoberta de Conhecimento em Bases de Dados, conhecida como pré-processamento de dados, é muito relevante para o sucesso do processo. É ela a responsável pelo tratamento dos dados, que os organiza e representa na forma adequada aos algoritmos de mineração, ou seja, a preparação dos dados envolve uma seqüência de operações destinadas a converter os dados originais em um formato adequado para as tarefas de processamento (PYLE, 1999). Pode incluir tarefas como a seleção (automática e/ou manual) de atributos relevantes, amostragem, transformações de representação, identificação e descrição de valores discrepantes (*outliers*), tratamento de valores ausentes (que é o objeto desta tese) entre outras. Na prática, presume-se que 80% do tempo gasto no processo de KDD envolva a etapa de pré-processamento (PYLE,1999).

1.2.Motivação

A Descoberta de Conhecimento a partir de casos armazenados em base de dados vêm contribuindo muito para o crescimento da ciência auxiliando na compreensão e aquisição de novos conhecimentos e apoiando processos decisórios. Áreas como a Medicina, a BioInformática, e as Ciências Sociais têm nesta tecnologia uma grande aliada. No entanto, de nada adianta extrair conhecimento de uma base que não é confiável, cujos dados não são consistentes.

O grande volume de dados gerados pela automatização de processos, somado a problemas inerentes à integração de diferentes fontes, de problemas técnicos ou simplesmente por esquecimento humano, levam a dados "poluídos", com valores desconhecidos, atributos com valores incorretos ou imprecisos (ruídos), ou mesmo ausentes. Portanto, a mineração dos dados pode ficar bastante comprometida se não for precedida por uma fase de limpeza das bases.

Uma abordagem bastante simplista e presente nos pacotes estatísticos é a eliminação de todo o caso ou de atributos que não apresente originalmente a qualidade desejada. Esta solução pode gerar problemas até maiores já que a amostra resultante pode ficar tendenciosa e preciosas informações podem ser perdidas.

Portanto, há uma tendência de recuperar estas informações. Em pequenos volumes de dados, este processo pode ser realizado de forma manual, mas na grande maioria dos problemas reais, o volume de dados é intratável pelo ser humano, necessitando do apoio

de processos automatizados. Soluções automáticas vêm sendo propostas, integrando diferentes áreas do saber. Porém, esta é uma tarefa complexa muito dependente do domínio de aplicação e como de se esperar, sem uma solução universal. Conforme afirmação de Junninen et al (2004), métodos de imputação não podem ser considerados um tipo de alquimia estatística onde a informação é gerada a partir do nada. Sendo assim, técnicas de aprendizado de máquina estão sendo ajustadas para este fim, ou seja, para descobrir o valor mais similar ao que supostamente estaria armazenado em tais atributos. Portanto, esta tese é motivada exatamente por este desafio: como imputar bons valores para preencher os diferentes atributos de bases de dados utilizando técnicas de Inteligência Artificial bem como analisar o impacto da ordem de imputação e da reutilização destes valores na imputação de valores conseguintes,.

1.3.Objetivo

O objetivo central da tese é propor uma nova abordagem para imputação multivariada: a Imputação em Cascata. O termo imputação pode ser usado como sinônimo de complementação de dados e o termo multivariada significa em muitas variáveis. Logo, o que está sendo proposto é uma estratégia, denominada Imputação em Cascata, para a complementação de dados em tabelas compostas por atributos numéricos em cujas tuplas podem haver vários atributos com valores ausentes simultaneamente.

Nesta abordagem híbrida, o processo de imputação é precedido pela tarefa de agrupamento. Os casos incompletos são distribuídos em grupos considerando como critério de pertinência o conceito de morfologia da ausência neles existentes. A morfologia de ausência é um conceito aqui proposto para descrever a distribuição de valores presentes e ausentes nos atributos de um conjunto de casos. Portanto, os grupos são formados por casos similares quanto à forma de distribuição de seus atributos não preenchidos. O objetivo do agrupamento e o critério são inéditos, pois os trabalhos existentes na literatura que fazem uso de um prévio agrupamento para a complementação de dados utilizam como critério a quantidade de valores ausentes ou características relevantes do domínio ou conjuntos nebulosos ou entropia, mas não analisam a distribuição espacial da ausência e a razão para agrupar, técnica conhecida por imputação *hot-deck* (FORD, 1983, FULLER, KIM, 2001), tem como objetivo dividir o conjunto original em grupos para que o processo de imputação seja

influenciado apenas pelos objetos do conjunto de dados que possuam alguma relação com aqueles incompletos.

Para que os valores dos atributos não influenciem o método de agrupamento, o subconjunto da base que contém casos incompletos é inicialmente binarizado, A binarização consiste em substituir valores presentes por um e valores ausentes por zero. Dado o viés em Inteligência Computacional, para agrupar os casos binarizados, utiliza-se uma rede SOM (s*elf-organizing map).* Após o agrupamento, os casos são restaurados e aplica-se a imputação sequencial com realimentação de valores para predizer os valores ausentes dos atributos de cada grupo. Para regressão de cada célula ausente, utiliza-se a média dos valores dos atributos dos casos selecionados pelo algoritmo dos *k*-vizinhos mais próximos (K-NN)

Nesta abordagem, há a possibilidade de também analisar outras variáveis que não apenas o desempenho do método tais como: topologias da rede, número de vizinhos, critérios de distância, critérios de ordenação entre grupos e intra-grupo. Para isso, foi construída uma plataforma, *workflow-like*, em Java, que gera, segundo parâmetros definidos, a seqüência de experimentos desejada. Para avaliar os resultados, usa-se como métrica uma medida de erro normalizada que constata o quão distante o resultado gerado está do real, normalizado pelo espectro de valores do atributo. Devido as controvérsias da literatura quanto à reutilização de valores, é desejado, também, avaliar o impacto da reutilização dos valores na correlação original das variáveis e a influência da ordem de imputação dos atributos.

Resultados foram gerados em cinco bases do repositório de aprendizado de máquina da Universidade da Califórnia, Irvine (MERZ 1998). Este repositório possui diversas bases de dados com as mais diferentes características, e serve como *benchmark* para diversos trabalhos na área de descoberta de conhecimento de bases de dados. As bases *Íris Plants*, *Pima Indians, Wisconsin Breast Cancer, Wine e Hardware Machine* foram escolhidas para os experimentos, pois são formadas por casos originalmente completos, condição necessária para a avaliação. Todos os atributos utilizados, à exceção da classe, são numéricos. A ausência dos dados foi gerada artificialmente, adotando-se o mecanismo de ausência completamente aleatório (MCAR - *Missing Completely At Random*), com diferentes percentuais de ausência. Estes índices variaram de 10% a 30%, com saltos de 10%. Para cada base foram criadas três versões sujas para

cada um dos percentuais citados, totalizando 45 bases. Os resultados gerados foram comparados com os gerados pela imputação seqüencial sem o prévio agrupamento, com e sem reutilização de valores. Algumas considerações sobre os parâmetros ajustáveis nos algoritmos foram também tecidas.

1.4.Organização do texto

Este documento está organizado do seguinte modo. No capítulo 2 são tratados os fundamentos teóricos deste trabalho. Os principais conceitos relacionados à tarefa de Pré-Processamento de Dados no contexto da Descoberta de Conhecimento são apresentados de modo a contextualizar a imputação de valores ausentes. No capítulo 3, a etapa de Complementação de Valores Ausentes é detalhada descrevendo conceitos e algumas tarefas, técnicas e algoritmos e soluções para o tratamento de dados ausentes. Neste capítulo também são citados os trabalhos relacionados presentes na literatura. O capítulo 4 introduz a Imputação em Cascata, solução aqui proposta para a o problema da ausência de valores em diferentes atributos de uma base de dados, ou seja, para imputação multivariada. Esta abordagem agrupa, por meio de uma rede neuronal do tipo SOM (*Self-Organizing Map),* os casos incompletos das bases de dados utilizando como critério a morfologia desta ausência, para então estimar os valores. Defende a reutilização dos valores previamente imputados e a importância da ordem que os mesmos são gerados. O capítulo 5 analisa experimentalmente o método proposto sobre cinco bases do repositório da Universidade da Califórnia, Irvine, conhecido como *UCI*: *Iris Plants*, *Wisconsin Breast Câncer*, *Pima Indians Diabetes*, *Hardware Machine e Wine*. O capítulo 6, encerra esta tese, apresentando conclusões e discussões finais bem como indicando os trabalhos futuros.

CAPÍTULO 2

A ETAPA DE PRÉ-PROCESSAMENTO DE DADOS NO CONTEXTO DO PROCESSO DE DESCOBERTA DE CONHECIMENTO EM BASES DE DADOS

2.1 Introdução

A descoberta de conhecimento em bases de dados (*Knowledge Discovery in Databases* ou KDD) visa buscar padrões desconhecidos existentes nas massas de dados e envolve diversas áreas de conhecimento, tais como a Estatística, Inteligência Artificial, Aprendizado de Máquina, Banco de Dados, Reconhecimento de Padrões, Armazém de Dados (*Data Warehousing*), Visualização de Dados, entre outras. Este "garimpo" de informações valiosas escondidas nos registros dá suporte à gestão do conhecimento permitindo identificar tendências, apoiar decisões, prever cenários futuros, descobrir padrões desconhecidos, realizar associações, formular teorias, entre tantas outras contribuições para o meio comercial e científico.

Uma das definições mais usuais para o processo de descoberta de conhecimento em bases de dados é a formulada por FAYYAD, PIATETSKY-SHAPIRO e SMYTH (1996a):

> *"KDD é um processo, de várias etapas, não trivial, interativo e iterativo, para identificação de padrões compreensíveis, válidos, novos e potencialmente úteis a partir de grandes conjuntos de dados".*

GOLDSCHMIDT e PASSOS (2005) analisam a definição acima, extraindo da mesma as principais características deste processo:

- Sua decomposição em etapas (sequencialmente executadas).
- O termo *interativo* indica que o homem é o responsável pelo controle do processo.
- O termo *iterativo* mostra a possibilidade de repetições de alguns passos do processo (refinamentos sucessivos) com intuito de se aprimorar os resultados obtidos.

- A expressão *não trivial* alerta para o grau de complexidade inerente ao processo, que envolve algum mecanismo de busca ou inferência, não se resumindo a apenas a um processo de totalização.
- A expressão *padrão válido* indica que o conhecimento deve ser verdadeiro (pelo menos com algum grau de certeza) e adequado ao contexto da aplicação de KDD.
- Um *padrão compreensível* indica que o mesmo deve estar representado de forma passível de ser interpretada pelo homem.
- Um *padrão novo* indica que o resultado deve acrescentar novos conhecimentos aos conhecimentos existentes no domínio da aplicação.
- Um *padrão útil* indica que os novos padrões devem ter aplicabilidade, trazendo algum benefício ao contexto da aplicação de KDD.

O macro-processo para que este conhecimento, a princípio invisível, possa emergir, envolve três grandes etapas. Inicialmente, há a preparação dos dados (1ª etapa) que serão explorados por meio de diferentes técnicas (2ª etapa) para posterior construção de modelos ou padrões que devem ser validados e/ou verificados (3ª etapa). Uma vez que os dados se constituem na fonte primária do processo, deve-se tomar cuidado com eles, lembrando o famoso axioma da computação: “garbage in garbage out”. Se dados inválidos forem manipulados, as saídas também serão inválidas, levando, por conseguinte, a conclusões inválidas.

A segunda etapa, onde os dados são minerados, pressupõe uma base de dados sem erros ou ruídos, limpa, com todos os registros completos e, preferencialmente, bem balanceados. No entanto, em bases de dados reais, este é um contexto quase utópico. Os dados, em geral, estão “poluídos”, com valores incorretos ou imprecisos (ruídos), ou mesmo com valores ausentes. Os bancos de dados procuram modelar o nosso conhecimento do mundo real e, portanto, são repletos de incertezas com as quais um processo de análise de dados deve lidar. (MOTRO 1995)

Sendo assim, a primeira etapa da Descoberta de Conhecimento em Bases de Dados, conhecida como pré-processamento de dados, é muito relevante para o sucesso do processo. É ela a responsável pelo tratamento dos dados, organizando-os e representando-os na forma adequada aos algoritmos de mineração. Para tanto, a preparação dos dados envolve uma seqüência de operações destinadas a converter os

dados originais em um formato adequado para as tarefas de processamento (PYLE, 1999). Como exemplos destas tarefas, podem ser citadas: a seleção (automática e/ou manual) de atributos relevantes, amostragem, transformações de representação, identificação e descrição de valores discrepantes (*outliers*), tratamento de valores ausentes (que é o objeto desta tese), entre outras. Na prática, presume-se que 80% do tempo gasto no processo de KDD envolva a etapa de pré-processamento dos dados (PYLE,1999). No entanto, mesmo considerando que sem dados de boa qualidade o resultado da mineração é pobre, a etapa de pré-processamento não recebe dos pesquisadores o mesmo investimento que a etapa de mineração.

O processo de KDD, proposto por FAYYAD e seu grupo em 1996 (FAYYAD et al, 1996b), pode ser resumido conforme ilustrado na figura 2.1.

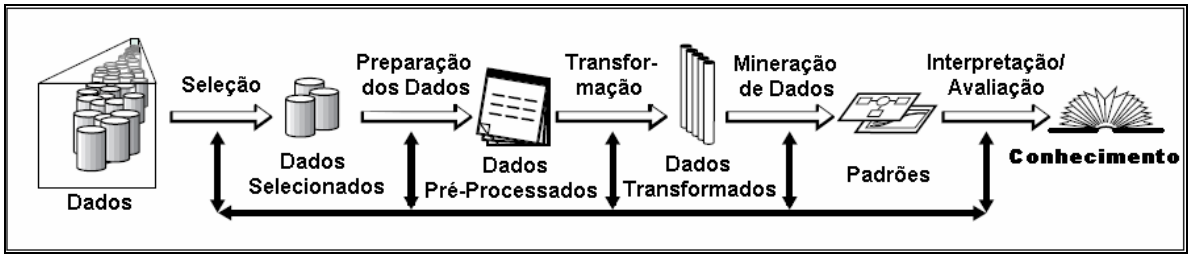


Figura 2.1: Visão geral do processo de KDD.
Fonte: Adaptado de (*FAYYAD* et al*, 1996b apud SOARES, 2007)*

Embora a granularidade das etapas difira entre os pesquisadores da área, todos concordam com há necessidade de: compreender o problema, determinando os objetivos do processo de KDD, preparar os dados para garantir a qualidade dos dados, extrair os padrões ou o modelo de comportamento a partir dos dados preparados e, por último, apresentar estes resultados em um modo compreensível ao ser humano.

O foco desta tese está no tratamento de valores ausentes para o processo de KDD. Por ser uma das tarefas de pré-processamento, esta etapa do processo de KDD encontra-se sucintamente descrito na próxima seção.

2.2 O Pré-Processamento dos Dados no Processo de Descoberta de Conhecimento em Bases de Dados (KDD)

2.2.1 Introdução

Nesta fase, os dados são preparados com intuito de aprimorar sua qualidade para posterior mineração. Algumas das principais atividades realizadas são (GOLDSCHMIDT, PASSOS, 2005, ZHANG *et al*, 2003):

- Seleção de Dados
- Agrupamento de dados
- Coleta e Integração
- Codificação
- Construção de Atributos
- Correção de Prevalência
- Discretização
- Limpeza dos Dados
- Normalização de Dados
- Partição dos Dados

2.2.2 Seleção de Dados

Esta atividade consiste em determinar o subconjunto de atributos relevantes ao processo de KDD.

2.2.2.1 Caracterização dos dados

O processo de descoberta de conhecimento é realizado sobre um conjunto de dados ou objetos (também chamados de observações, padrões, exemplos, registros ou tuplas) que descrevem entidades, operações, experiências idéias ou conceitos do mundo real. A descrição de um caso é representada pelas diversas características ou atributos destes objetos, armazenadas em variáveis, cujos valores expressam a respectiva medida de um caso. São descritos genericamente como um vetor $X = (x_1,..., x_n)$, onde X é o caso, e x_i, i=1,...,n, são os atributos (CIOS *et al*, 1998, PYLE, 1999).Os atributos ou variáveis podem ser classificadas em (PYLE, 1999):

- Nominais ou Categóricas – São variáveis utilizadas para nomear ou atribuir rótulos a objetos. O domínio dos valores que estas variáveis podem assumir é um conjunto finito e pequeno de estados possíveis. Como exemplo pode-se citar o tipo de residência do aluno: própria, alugada, de parentes, funcional ou outros. Nas variáveis nominais não há um ordenamento de seus valores. Não se pode dizer que "própria" é menor que "alugada", por exemplo. Os valores de variáveis nominais podem ser representados por tipos de dados alfanuméricos.
- Discretas – assemelham-se às variáveis nominais, mas, os valores (estados) que elas podem assumir possuem um ordenamento, e este possui algum significado. O dia da semana é um bom exemplo deste tipo de variável, onde a "segunda-feira" vem após o "domingo" e antes da "terça-feira", ou nível de curso, pois "mestrado" é posterior à "graduação" e anterior a "doutorado". Podem ser partições ou intervalos de um domínio contínuo de valores. Por exemplo: faixa de despesa e faixa etária, dentre outros. As variáveis discretas também podem ser representadas por tipos de dados alfanuméricos.
- Contínuas – são variáveis que representam quantidades e há uma relação de ordem entre os valores. O conjunto de valores de uma variável contínua pode ser finito ou infinito. Renda de uma pessoa ou percentual de desconto são exemplos de variáveis contínuas. Em geral os valores das variáveis contínuas são representados por um tipo de dado numérico.

Já GOWDA e DIDAY (1992) classificam os atributos nos seguintes tipos:

1) Atributos Qualitativos ou Categorizados
   a. Ordinais
   b. Nominais
2) Atributos Quantitativos ou Numéricos
   a. Valores contínuos
   b. Valores discretos
   c. Intervalo de valores

Nos atributos **ordinais,** os valores do domínio expressam alguma ordem. Por exemplo, respostas de pesquisas de opinião onde há associado a alguma escala de intensidade (1- Discordo 2-Concordo Parcialmente 3- Concordo 4- Concordo Plenamente) ou *Estatura* (1- Baixo, 2-Médio, 3-Alto). Este tipo de dado é também

conhecido como escala de temperatura, ou *likert data* (JÖNSSON, WOHLIN, 2006). Já os atributos **nominais**, assumem valores possíveis do domínio de aplicação sem nenhuma noção de precedência, ou seja, são não numéricos e não ordenados. Por exemplo, considere o atributo *mobiliário*, podendo assumir os valores *cadeira, poltrona, sofá* ou o atributo *cores básicas*, podendo assumir os valores *branco, preto, azul, amarelo ou vermelho*. Não há como expressar ordem nestes conjuntos de valores.

Os atributos quantitativos podem ser medidos utilizando-se alguma escala. A natureza do conjunto de valores desta escala é que os diferencia. Os atributos **contínuos** sempre comportam um terceiro valor entre dois outros (qualquer atributo cujo domínio é um número real, por exemplo, a altura de uma pessoa). Os possíveis valores dos atributos discretos não admitem a propriedade acima, (por exemplo, os números inteiros). Nos dois casos, não existe limite inferior ou superior. Quando eles existirem, dizemos que estamos tratando de um **intervalo de valores** (por exemplo, o valor do percentual de bolsa concedido a um aluno pode variar de 0 a 100%).

2.2.2.2 Seleção de Dados

Esta atividade, também conhecida como Redução de Dados, preocupa-se em extrair os atributos que devem ser efetivamente considerados no processo de KDD. Por exemplo, caso o objetivo do processo de KDD seja prever a apólice de seguro de vida de novos clientes, o consumo de energia elétrica é uma informação irrelevante mas torna-se relevante caso o processo tenha como meta prever as necessidades de investimento em infra-estrutura básica de um bairro. A seleção dos dados pode ocorrer de dois modos distintos: pela escolha de atributos (redução de dados vertical) ou pela escolha de registros (redução de dados horizontal).

A **redução de dados horizontal** consiste em escolher os registros da base que satisfazem o processo de KDD. É utilizado quando a base tem muita informação redundante ou quando há registros que estão fora do escopo do processo. Entre as operações de redução de dados horizontal podem ser citadas:

- Amostragem aleatória: consiste em selecionar um número pré-estabelecido de registros de forma que o conjunto resultante possua menos registros do que o conjunto original;

- Segmentação do banco de dados: a partir da escolha de um ou mais atributos norteadores do processo, seleciona-se os registros que satisfaçam a condições determinadas sobre os mesmos. Por exemplo, deseja-se avaliar o desempenho escolar dos alunos do ensino básico de uma escola que atua em todo o ensino fundamental. Neste exemplo, apenas os registros dos alunos que freqüentam do primeiro ao quinto ano devem ser considerados;

- Eliminação direta de casos: esta operação é similar à anterior, mas aqui são especificados os casos a serem eliminados e não os casos que devem permanecer na análise;

- Agregação de informações: consiste em consolidar os dados, diminuindo o detalhamento, de forma a reduzir o conjunto de dados original. Por exemplo: somar o número de faltas do aluno no ano.

A **redução de dados vertical**, também chamada de redução de dimensão, é implementada pela eliminação ou pela substituição dos atributos de um conjunto de dados. Deste modo, visa encontrar um conjunto mínimo de atributos de tal forma que a informação original seja preservada. Entre as principais motivações para a aplicação da redução de dados vertical podem ser citadas: (a) um conjunto de atributos bem selecionado pode conduzir a modelos de conhecimento mais concisos e com maior precisão; (b) se o método de seleção dos atributos for rápido, o tempo de processamento necessário para utilizá-lo e, em seguida, aplicar o algoritmo de mineração de dados em um subconjunto dos atributos, pode ser inferior ao tempo de processamento para aplicar o algoritmo de mineração sobre todo o conjunto de atributos; (c) a eliminação de um atributo é muito mais significativa em termos de redução do tamanho de um conjunto de dados do que a exclusão de um registro (GOLDSCHMIDT, PASSOS, 2005).

Há duas abordagens básicas para a redução vertical (FREITAS,2002, GOLDSCHMIDT, PASSOS, 2005):

- *Filter* (abordagem independente de modelo): consiste em analisar previamente o conjunto de dados para selecionar os atributos relevantes. A seleção de atributos é realizada desconsiderando o algoritmo de mineração que será aplicado aos atributos;

- *Wrapper (*abordagem dependente de modelo) – Esta abordagem consiste em iterativamente escolher um subconjunto de atributos, aplicar o algoritmo de mineração de dados para este conjunto de atributos e avaliar os resultados obtidos, buscando o subconjunto com melhor desempenho. Este ciclo de avaliação de subconjuntos de atributos candidatos continua até atingir algum critério de parada pré-estabelecido.

Embora a abordagem *wrapper* produza um subconjunto de atributos mais adequado para melhorar o desempenho dos algoritmos de mineração de dados, ela apresenta duas desvantagens: a de ser dependente do algoritmo, pois o sub-conjunto ideal para um algoritmo não necessariamente o é para outro, e ser muito dispendiosa, visto que realiza uma "busca cega" no espaço de soluções.

Há diversas estratégias interessantes para implementar as abordagens acima e escolher o conjunto de atributos a ser utilizado (AHA e BANKERT, 1995, BALA et al, 1995, HAN, KAMBER, 2001, FREITAS, 2002). Algoritmos Genéticos, por exemplo, são bastante utilizados no processo de otimização do conjunto de atributos. Algoritmos para indução de Árvores de Decisão, tais como ID3 e C4.5, também podem ser aplicados para selecionar atributos em problemas de classificação. Neste caso, os atributos não incluídos na árvore de decisão criada são eliminados. Han e Kamber (2005) mencionam uma técnica de seleção de atributos chamada *discrete wavelet transform (DWT)*, baseada no processamento linear de sinais, que, quando aplicada a um vetor *D*, transforma-o em um novo vetor *D'*, de mesma dimensão, com coeficientes *wavelet*. A redução da dimensão do vetor é obtida pela truncagem do conjunto *D'* modificado, considerando-se apenas os coeficientes maiores do que um valor especificado como parâmetro.

Uma técnica estatística amplamente aplicada, principalmente na área de processamento de sinais e imagens, estudada por Pearson em 1901 e consolidada por Hotteling em 1933, é chamada de análise de componentes principais (PCA), cujo objetivo é reduzir os atributos (ou variáveis no contexto estatístico) quantitativos.

Esta técnica multivariada consiste, basicamente, em analisar um conjunto de variáveis numéricas com alta dimensionalidade de representação, "compactando-as" de modo a reduzir o número de variáveis mas mantendo a máxima variabilidade dos dados originais. Deste modo, após a compactação o poder de informação mantém-se quase

igual ao original. O processo de redução do conjunto de variáveis originais consiste em encontrar combinações lineares destas variáveis, que irão gerar um outro conjunto de variáveis com novas coordenadas e não correlacionadas entre si. Geometricamente, as componentes principais representam um novo sistema de coordenadas, obtidas por uma rotação do sistema original, que fornece as direções de máxima variabilidade, e proporciona uma descrição mais simples e eficiente da estrutura de covariância dos dados. (JOHNSON, WICHERN, 2002)

2.2.3 Agrupamento de Dados

O agrupamento de dados é usado tanto como uma tarefa na mineração de dados como no pré-processamento dos dados. Visa estruturar os dados em grupos (*clusters*), alocando em um mesmo grupo os elementos semelhantes e elementos diferentes em grupos distintos (BERRY e LINOFF, 2000). Esta segmentação utiliza alguma medida de similaridade (vide apêndice I) A figura 2.2 demonstra como é realizado o agrupamento.

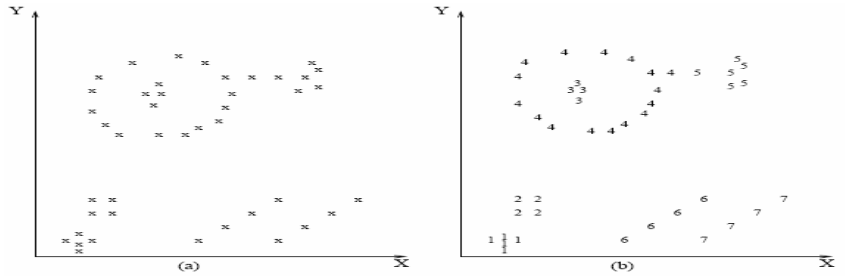


Figura 2.2 Exemplo de agrupamento de dados em sete grupos, onde o rótulo sobre o elemento indica a qual grupo ele pertence. Fonte: JAIN et al (1999)

Esta técnica tem ampla aplicação em diversas áreas, incluindo descoberta de conhecimento, análise estatística, compressão de dados, quantização vetorial, reconhecimento e classificação de padrões. Uma das aplicações comuns é a construção automatizada de categorias ou taxonomias (BERRY e LINOFF, 1997)

Há diversas técnicas de agrupamento, mas dois aspectos as definem como tal (CIOS et al., 1998):

a) os grupos não são pré-conhecidos e com isto não há supervisão no processo de divisão dos elementos. Estes são organizados de acordo com um critério definido pelo método de agrupamento;

b) a noção de similaridade (ou distância) entre dois elementos guia o processo. O algoritmo de agrupamento implementa uma função de distância capaz de

quantificar a similaridade. Em geral, quanto menor a distância, maior a similaridade, sendo que os elementos são iguais se a distância for 0.

Portanto, não há pré-classificação dos dados ou distinção entre variáveis dependentes ou independentes. Aplicam-se algoritmos de aprendizagem não supervisionada que se caracterizam por lidar diretamente com os dados, buscando encontrar padrões que determinam a estrutura destes elementos (BERY e LINOFF, 2000; CIOS et al., 1998).

Na etapa de pré-processamento o agrupamento de dados é bastante utilizado para eliminar o excesso de informações, compactando a base e simplificando a compreensão dos padrões. Por ter sido utilizado nesta tese como uma tarefa da etapa de preparação dos dados para posterior imputação, alguns métodos que implementam agrupamento de dados encontram-se descritos no apêndice III e IV.

2.2.4 Coleta e Integração

Esta atividade é responsável por coletar os valores dos atributos que serão minerados. No cenário atual, é comum que os dados para a mineração sejam provenientes de diferentes bases de dados. Descobrir onde estes dados estão, como coletá-los e integrá-los pode ser tornar uma tarefa árdua e demorada. O tempo de existência e a freqüência da carga da base de dados aumentam a chance de haver inconsistências e valores ausentes nos bancos de dados. Além disso, os dados podem não estar num formato adequado para serem utilizados. Portanto deve-se encontrar e fundir os dados provenientes de diferentes fontes em uma única fonte coerente.

Alguns desafios desta fase, segundo Pyle (1999) são: (a) problemas legais e/ou éticos, por exemplo, o valor do imposto de renda de um cliente de um banco ou a identificação de pacientes na área médica (b) motivos estratégicos, por exemplo, a proporção de operações fraudulentas e não fraudulentas de uma administradora de cartões é uma informação considerada estratégica e, portanto, sigilosa (c) razões políticas (d) formato dos dados, por exemplo, o formato da data ou código usado (EBCDIC, ASCII) (e) bancos e aplicações obsoletas, por exemplo, o cálculo de impostos que não se aplicam mais – (IPMF) (f) granularidade, ou nível de detalhe que a informação está armazenada.

Segundo Han e Kamber (2005) vários aspectos devem ser considerados no processo de integração de dados: a *integração de esquemas,* a *eliminação de redundâncias* e a *detecção e correção de dados com valores conflitantes*.

2.2.5 Codificação

Aqui, a meta é transformar a representação dos valores de um atributo para adaptá-lo às limitações do algoritmo de mineração escolhido. Na etapa de pré-processamento, isto pode acontecer de duas diferentes formas (GOLDSCHMIDT, PASSOS,2005):

- codificação numérica-categórica: os valores dos atributos originalmente numéricos são mapeados em categorias e por elas substituídos.

- codificação categórica-numérica**:** mapeia atributos qualitativos em quantitativos, ou seja, os k valores do domínio de um atributo são transformados na representação binária de números discretos.

2.2.6 Construção de Atributos

Em geral esta operação envolve incluir no conjunto de dados atributos derivados a partir dos já existentes, ou seja, construir novos atributos relevantes para a descrição de um conceito. Um exemplo bastante simples é a inclusão de um atributo *idade* em uma base que possui o campo *data de nascimento.* Esta tarefa é relevante, pois a maioria dos algoritmos de mineração não lida de forma adequada com o formato data.

2.2.7 Correção de Prevalência

Muitas vezes as bases de dados estão desbalanceadas, ou seja, há muitos registros representantes de alguma classe e poucos de outras e este desequilíbrio pode levar a resultados tendenciosos ou esconder determinados padrões. Pode-se solucionar este problema utilizando *amostragem estratificada,* onde são selecionados da base original quantidades semelhantes de registros para as classes envolvidas ou *replicação aleatória de registros* na qual são selecionados aleatoriamente registros das classes menos freqüentes e replicados na base. Esta abordagem pode levar a resultados tendenciosos (GOLDSCHMIDT ,PASSOS, 2005) .

2.2.8 Limpeza de Dados

A limpeza dos dados é uma das atividades essenciais no pré-processamento, pois nesta atividade, os erros existentes nas bases devem ser detectados e corrigidos, os valores ausentes devem ser complementados, e os valores discrepantes devem ser identificados, entre outros. Nas aplicações reais, é usual os dados se encontrarem incompletos, inconsistentes ou com ruídos. Os dados são considerados incompletos quando não estão suficientemente detalhados ou estão ausentes. Dados ruidosos são aqueles inválidos ou com valores atípicos (*outliers)*. Nos dados considerados inconsistentes há discrepância semântica. A detecção de ruídos visa ajustar valores atípicos ou com características bastante distintas do usual. Portanto, são sub-tarefas da limpeza:

- Complementação de Dados Ausentes
- Detecção de Ruídos
- Eliminação de Dados Inconsistentes

A complementação de dados ausentes, também conhecida como imputação, pode ser vista como o ato de preencher as lacunas geradas pela ausência de valores em atributos de bases de dados. A imputação é responsável em resgatar tais atributos, atribuindo-lhes valores que embora estimados sejam os mais semelhantes possíveis aos originais. Há diferentes técnicas para realizar esta complementação. Elas dependem da natureza do atributo e em geral valem-se dos valores existentes na base original para completar os casos incompletos. No entanto, todo o cuidado é necessário durante esta atividade porque se a mesma for realizada de forma inadequada, características distintas às originais podem ser introduzidas na base, prejudicando a análise realizada.

A complementação de dados ausentes é o foco desta tese e será tratada com mais detalhes no capítulo 3.

Para tratar *outliers* costuma-se utilizar técnicas de agrupamento ou inspeção. A inspeção é uma combinação da análise humana e computadorizada: valores que destoam muito da tendência geral dos dados de um conjunto podem ser identificados por algum algoritmo e analisados pelo especialista do domínio. No caso de tratar a identificação de *outliers* por técnicas de agrupamento, inicialmente reúne-se os valores em grupos e os valores isolados podem ser considerados atípicos. A figura 2.3 ilustra esta idéia:

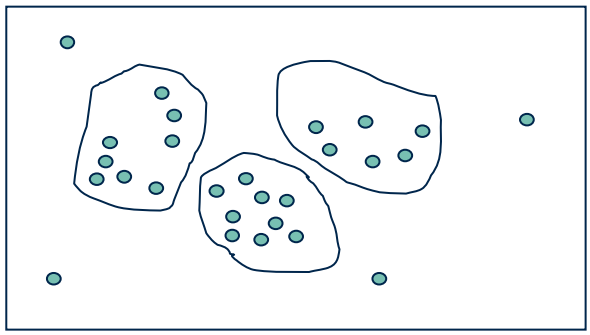


Figura 2.3: Detecção de "outliers" por agrupamento
Fonte: Adaptado de (CARVALHO, 2002)

Para eliminação de valores inconsistentes é usual utilizar métodos de suavização ou regressão. Uma das técnicas de suavização, conhecida como *encaixotamento* consiste em suavizar um conjunto de dados ordenados observando sua vizinhança. Os valores ordenados são distribuídos em um número de "compartimentos", também denominados *bins*. Na *regressão* os dados podem ser suavizados aplicando-se uma função matemática. *A figura 2.4* exemplifica esta idéia com duas variáveis e regressão linear.

A inconsistência nos dados pode ser introduzida por diversas causas e em diferentes momentos. Pode ocorrer tanto no momento da carga dos dados, como na integração de várias bases. Neste caso, é comum ter-se o mesmo atributo com diferentes codificações ou a duplicação de objetos. Erros desta natureza podem ser corrigidos manualmente (pela observação do analista de dados) ou automaticamente por ferramentas de Sistemas Gerenciadores de Banco de Dados ou métodos de Mineração de Dados.

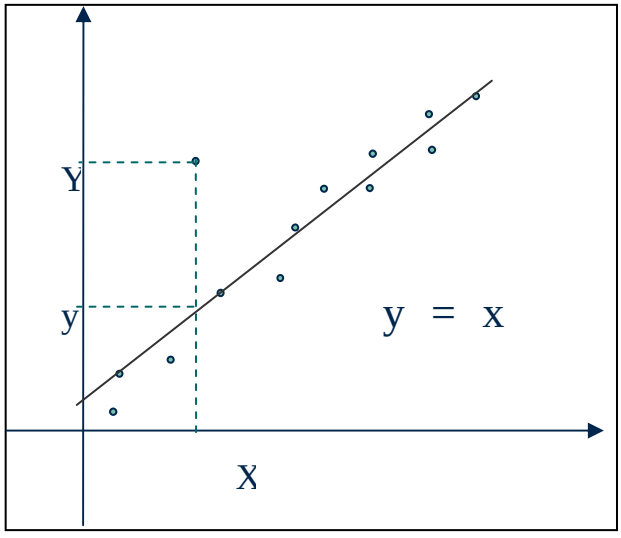


Figura 2.4: Suavização por regressão. Fonte: Adaptado de (CARVALHO, 2002)

2.2.9 Normalização de Dados

Como a natureza dos atributos que definem os objetos pode diferir muito, bem como a grandeza de seus valores, muitas vezes é necessário transformar os valores para uma escala de menor abrangência, evitando, assim, padrões tendenciosos. Uma das maneiras de obter-se este ajuste de escala é por meio de métodos de normalização onde estes valores são mapeados para um dos intervalos: [0.0, 1.0] ou [-1.0, 1.0]. (vide apêndice I)

2.2.10 Criação de Partições dos Dados

A qualidade dos modelos extraídos dos dados na etapa de mineração precisa de algum modo ser mensurada e testada. Para esta avaliação ser fidedigna dados que não participaram da criação destas abstrações são necessários. Portanto, os dados originais devem ser divididos em dois conjuntos: um conjunto de treinamento, cujos registros são utilizados para a construção do modelo de conhecimento e um conjunto de testes, cujos registros são usados na avaliação do modelo resultante. Esta divisão do conjunto de dados deve ser realizada com muito cuidado para que o modelo construído reflita realmente os padrões existentes nos dados originais.

A partição de dados pode ser feita utilizando várias abordagens (GOLDSCHMIDT, PASSOS, 2005):

- *Holdout:* este método divide aleatoriamente os registros em uma percentagem fixa p para treinamento e $(1 - p)$ para teste, considerando normalmente $p>½$. Esta abordagem é muito utilizada quando desejamos produzir um único modelo de conhecimento a ser aplicado posteriormente em algum sistema de apoio à decisão.
- Validação Cruzada com K Conjuntos (*K-Fold Cross Validation*): divide aleatoriamente o conjunto de dados com N elementos em K subconjuntos disjuntos (*folds*), com aproximadamente o mesmo número de elementos (N/K). Cada um dos K subconjuntos é utilizado como conjunto de teste e os ($K-1$) demais subconjuntos são reunidos em um conjunto de treinamento. O processo é repetido K vezes, sendo gerados e avaliados K modelos de conhecimento.

- Validação Cruzada com K Conjuntos Estratificada (*Stratified K-Fold Cross Validation*)*:* aplicável em problemas de classificação, este método é similar à *Validação Cruzada com K Conjuntos*, sendo que na geração dos subconjuntos mutuamente exclusivos, a proporção de exemplos em cada uma das classes é considerada durante a amostragem. Por exemplo, se o conjunto de dados original possui duas classes com distribuição de 40% e 60%, cada subconjunto também deverá conter aproximadamente esta mesma proporção de classes.

- *Leave-One-Out*: este método é um caso particular da *Validação Cruzada com K Conjuntos*, em que cada um dos *K* subconjuntos possui um único registro. É computacionalmente dispendioso e freqüentemente usado em pequenas amostras.

- *Bootstrap*: O conjunto de treinamento é gerado a partir de *N* sorteios aleatórios com reposição a partir do conjunto de dados original (contendo *N* registros). O conjunto de teste é composto pelos registros não sorteados do conjunto original para o conjunto de treinamento.

  Este método consiste em gerar os conjuntos, abstrair e avaliar o modelo de conhecimento um número repetido de vezes, a fim de estimar uma média de desempenho do algoritmo de Mineração de Dados.

CAPÍTULO 3

TRATAMENTO DE VALORES AUSENTES EM BASES DE DADOS

3.1.Introdução

Uma das principais e mais desafiadoras tarefas da etapa de pré-processamento de bases de dados é a complementação de valores que por alguma razão não se encontram disponíveis nas tabelas de um conjunto de dados. O processo de Descoberta de Conhecimento ou a Análise Estatística de Bases de Dados podem ficar bastante comprometidos com a incompletude dos dados já que, por conta desta ausência, os padrões gerados por alguns destes algoritmos são pouco esclarecedores ou tendenciosos.

A forma de tratar os dados incompletos é a chave do sucesso para seu posterior aproveitamento. Esta escolha diferencia um estudo tendencioso de um não tendencioso (TWALA *et al*, 2005). Porém, não existe um método único que atue bem em todos os tipos de dados ausentes. Diferentes situações exigem diferentes soluções (CRÉMILLEUX, RAGEL, BOSSON, 1999, RAGEL, CRÉMILLEUX, 1999, TSENG, WANG, LEE, 2003, MAGNANI, 2004). Seja qual for a técnica aplicada para complementar os valores ausentes as características originais da amostra, entre elas podemos citar a variância e a correlação dos atributos entre outros, não devem ser distorcidas (HRUSCHKA *et al*, 2003a).

O desafio da imputação de dados se torna ainda maior quando diferentes atributos em uma mesma base de dados apresentam valores ausentes. Esta é uma característica presente nas bases reais. No entanto, não se encontram disponíveis na literatura muitos trabalhos com este enfoque o que estimula o estudo e desenvolvimento de métodos de complementação com enfoque multivariado. Como um dos objetivos centrais desta tese é sugerir uma nova abordagem chamada de Imputação em Cascata, que aplica técnicas de Inteligência Computacional na tarefa de complementação de dados em cenários mutivariados, este capítulo aborda algumas das principais causas para ausência de dados, conceitos relevantes para o tratamento de valores ausentes, tais como o

mecanismo de ausência e os padrões de ausência e formas de lidar com a ausência de dados. Apresenta, também, algumas soluções e técnicas usuais para tratar a ausência de dados bem como taxonomias existentes para organizá-las. O capítulo é finalizado com trabalhos relacionados.

3.2.Possíveis causas da ausência de dados

Existem inúmeras razões para a existência de dados desconhecidos em um conjunto de dados. Em bases de dados que armazenam resultados de pesquisas de opinião, por exemplo, a ausência pode ser causada por desinteresse humano, como falta de tempo ou de comprometimento dos entrevistados, falta de um treinamento adequado por parte dos coletores de dados, razões técnicas, razões políticas, entre outros (CARTWRIGHT *et al*, 2003). É comum, também, os valores de um atributo estarem preenchidos com valores inconsistentes como, por exemplo, um atributo que indique o número de vezes que um paciente homem engravidou (ONISKO *et al*, 2002).

BROWN e KROS (2003 apud SOARES 2007), enumeram as seguintes razões para atributos de registros de tabelas não estarem preenchidos:

1) *Fatores operacionais*: erros na entrada de dados, estimativas erradas, remoção acidental de campos de tabelas, entre outras.

2) *Recusa na resposta em pesquisas*: entrevistados não respondem as questões por se sentirem constrangidos em responder (perguntas como idade, renda ou orientação sexual), ou por não conhecerem o assunto (por exemplo, índices de crescimento da economia);

3) *Respostas não aplicáveis*: por vezes, questões apresentadas em questionários não se aplicam aos entrevistados (por exemplo, perguntas envolvendo freqüência de empréstimos bancários a menores de idade ou questões direcionadas a fumantes aplicadas a não fumantes)

3.3.Mecanismos de ausência de dados

O mecanismo que causou a ausência de dados é extremamente relevante na escolha do método para complementar tais informações, sendo considerados em quase todos os trabalhos presentes na literatura. Dividem-se em três tipos: completamente aleatórios, aleatórios e não aleatórios.

Para apresentá-los de um modo mais formal, considera-se *D* um conjunto de dados, que inclui a variável dependente *Y* e as variáveis explanatórias *X*: *D* = {*Y* , *X}*. *D* pode ser subdividido em D_{pres} (o subconjunto de dados com valores presentes) e D_{aus} (o subconjunto com valores ausentes), e *R* um conjunto indicador de respostas (R_i = 1 indica que o i-ésimo elemento de D está preenchido, e R_i = 0 revela que o valor é ausente). Diz-se que o mecanismo é:

- Completamente Aleatório (MCAR – *Missing Completely At Random)* quando a probabilidade de um valor estar ausente independe de qualquer outra característica observada ou não observada da amostra, ou seja, a razão é desconhecida. *R* é independente de *D*: P(*R/ D)* = P*(*R). Isso significa que a probabilidade de encontrar um valor ausente é a mesma para qualquer atributo. Uma ilustração deste mecanismo pode ser o fato de cada entrevistado jogar um dado antes de responder uma questão e não respondê-la caso apareça o número "4". Esta condição é extremamente restritiva e necessária para tornar válida a exclusão do caso, mas os valores ausentes são raramente MCAR (RUBIN, 1976).

- Aleatório (*MAR – Missing At Random):* quando os valores faltantes podem ser completamente explicados por valores observados na amostra.. Por exemplo, em uma Pesquisa de Indicadores Sociais, a diferença na taxa de não-resposta entre brancos e negros pode mostrar que a questão sobre "ganhos" não é completamente aleatória ou em um questionário uma pessoa idosa não responde uma questão porque, pela diminuição de sua memória, tem dificuldade em lembrar de um evento. Observando-se as respostas de pesquisas encontra-se, na maioria das vezes, uma razão para a ausência, isto é, ela é independente de valores não observados. Logo, *R* é independente de $D_{aus,}$ ou seja: P(*R/ D)* = P*(*R/ D_{pres}). Deste modo, o termo MAR, ausência aleatória, é ambíguo e não muito adequado pois a ausência do dado não pode ser caracterizada como randômica. Ela é condicionada por outra variável X observada no conjunto de dados embora não na variável objetivo Y (SCHAFER, 1997). A maior parte das técnicas de imputação exige que os dados ausentes sigam mecanismos do tipo MCAR ou MAR.

- Não aleatório (*NMAR – Not Missingt At Random)*: quando a ausência está relacionada com os valores ausentes em vez de valores não observados, ou seja, *R* não é independente de D: P(*R/ D)* e não pode ser simplificada. Ocorre quando um mecanismo de ausência depende do valor real do dado ausente. Por exemplo, o peso

de uma pessoa não foi registrado porque era superior ao valor máximo da balança utilizada ou em um questionário, o entrevistado responde “não sei”, ou recusa-se a responder a questão porque a resposta é socialmente indesejável, como “bebe muito”. É a condição mais difícil de modelar e, na maioria dos casos, não é possível distinguir entre mecanismos MAR e NMAR. (LITTLE, RUBIN, 1987).

Para exemplificar os conceitos acima, considere a seguinte tabela com os dados de trabalhos impressos em uma gráfica:

Id	Data	Papel	Orientação	Num.de Pág	Impressora	Tipo	Qualidade
1	05/05	A4	Retrato	1	HP	Escolar	Rascunho
2	05/05	A4	Paisagem	40	HP	Escolar	Alta
3	06/05	A5	Retrato	30	Epson	Profissional	Normal
4	06/05	A6	Paisagem	20	HP	Profissional	Normal
5	06/05	A5	Retrato	12	Epson	Profissional	Alta
6	07/05	A6	Paisagem	1	Epson	Profissional	Rascunho

Tabela 3.1 - Tabela com dados de trabalhos impressos

Caso o número de páginas estivesse ausente nos registros 1 e 6, porque o profissional da gráfica simplesmente esqueceu de anotar a quantidade de páginas, o mecanismo de ausência é considerado MCAR. No entanto, se estes valores estivessem ausentes porque as impressoras não fornecem o total de páginas na qualidade Rascunho, o mecanismo de ausência é considerado MAR. Agora o mecanismo é NMAR se o contador de números de páginas ao ser ligado demora 20 minutos antes de começar a registrar o número de folhas e estes atendimentos foram realizados antes deste intervalo.

Os mecanismos de ausência afetam o desempenho dos métodos estatísticos de inferência (NIRELLI et al, 2003). Para ilustrar esta afirmação, considere X e Y duas variáveis da amostra e que como meta deseja-se avaliar três distribuições: a distribuição marginal de X (f(X)), a distribuição marginal de Y (f(Y)) e a distribuição condicional de Y dado X (f(Y|X)). Na amostra, X está completamente observada mas alguns valores de Y estão ausentes. Como X está totalmente preenchida, a inferência de sua distribuição marginal não é afetada por dados ausentes e a aplicação de métodos tradicionais geram inferências válidas. Agora inferências sobre a distribuição de Y podem ser potencialmente afetadas pela razão da ausência de alguns valores de Y:

- Se Y é MCAR, então a razão dos dados estarem ausentes são ignorados. Como resultado, inferências de f(Y) e f(Y|X) usando somente os casos completos (onde X e Y estão observados) não serão tendenciosas pela ausência dos valores de Y. Podem apenas ser menos precisas pela redução do tamanho da amostra.
- Se Y é MAR e a probabilidade de Y estar ausente depende de X, ao usar apenas os casos completos, a inferência de f(Y) pode ser tendenciosa. Imaginando o cenário no qual X e Y são positivamente correlacionadas e registros com valores maiores em X são mais prováveis de ter Y ausente, ter-se-á como conseqüência uma média de valores observados de Y menor do que a real. No entanto, a variável X pode ser útil na estimativa da média da variável Y, principalmente se X é conhecida em mais casos que Y, eliminando a análise tendenciosa (LITTLE e RUBIN, 2003).
- Agora se o mecanismo de ausência é NMAR, a análise de f(Y) ou f(Y|X) sem modelar o mecanismo leva a inferências tendenciosas, pois o ajuste da análise de Y em função de X não é suficiente para remover todas as tendências (LITTLE e RUBIN, 2003).

Em resumo, conforme já afirmado, o padrão de ausência é um fator determinante na escolha de como devem ser tratados os valores desconhecidos. Segundo Little e Rubin (1987), os mecanismos exigem estratégias de análise e preenchimento específicas. Se os dados são MCAR, ou mesmo MAR, o mecanismo de ausência pode ser ignorado e as técnicas usuais de imputação podem ser aplicadas. No entanto, quando os dados são NMAR, o mecanismo não pode ser ignorado, pois a amostra pode se tornar seriamente tendenciosa e técnicas não tradicionais devem ser aplicadas (LEEUW,2001). Por sua vez, BATISTA e MONARD (2003a) alegam que, na maioria dos casos, os atributos não são independentes. Descobrindo-se a relação entre eles, um modelo representando esta relação pode ser criado sendo que os valores ausentes são então obtidos em função dos demais atributos da relação. Por outro lado, TWALA, CARTWRIGHT e SHEPPERD (2005) afirmam que é impossível tratar os dados como MAR e NMAR sem considerações adicionais pois na maioria das vezes a conjunção de fatores que causa a ausência é desconhecida. Desta forma, a menos que a razão da ausência seja completamente conhecida, é razoável tratar os dados como MCAR.

Considerando o fato de conhecer ou não o motivo da ausência, WAYMAN (2003) apud GRAHAM e DONALDSON (1993) classifica os mecanismos de ausência em:

acessíveis e inacessíveis. O mecanismo é considerado *acessível* quando a causa da ausência pode ser explicada. Incorpora, portanto, dados completamente aleatórios (MCAR) e dados aleatórios (MAR). Por outro lado, o mecanismo é *inacessível* quando sua razão é desconhecida. Aqui, se pode incluir os dados que não são aleatórios (NMAR) ou dados aleatórios (MAR) onde a causa da ausência é conhecida, mas não medida.

3.4. Padrões de ausência de dados

Para alguns pesquisadores, entre eles SCHAFER e GRAHAM (2002) e TWALA, CARTWRIGHT e SHEPPERD (2005), a escolha do método de imputação depende do mecanismo de ausência e também do **padrão de ausência**. O padrão de dado ausente está relacionado com a forma como os dados são obtidos. Estes, por sua vez, são categorizados em *gerais* (ou *não monotônicos* ou *arbitrários)*, e *específicos*.

No padrão de ausência arbitrário ou não monotônico, o valor de qualquer atributo de qualquer registro pode estar ausente.

Os padrões de ausência específicos ainda podem ser sub-divididos em *univariados* e *monotônicos.* Nos padrões univariados apenas uma variável apresenta ausência de valores. Um padrão é considerado monotônico se é possível reorganizar o conjunto de atributos A_i $1 \leq i \leq n$, da tabela de tal modo que não haja dados nos atributos A_j, A_{j+1}, A_{j+2}, ... , A_p, $j < p$. A figura 3.1 ilustra graficamente os conceitos apresentados. Métodos mais simples de imputação podem ser usados em padrões monotônicos.(HORTON e LIPSITZ,2001)

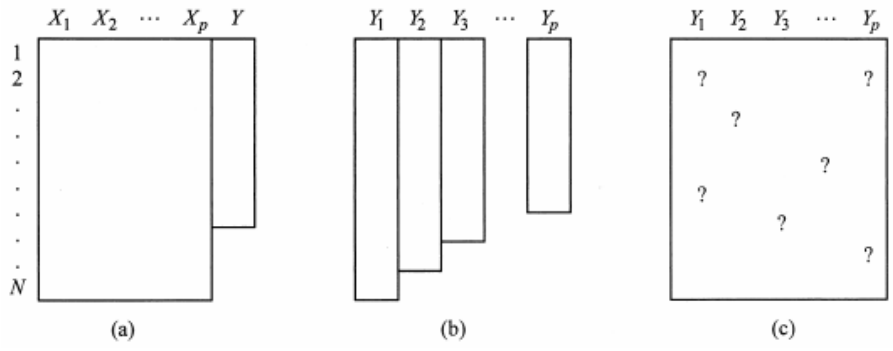


Figura 3.1 Padrões de ausência de respostas em um conjunto de dados retangulares: (a) padrões univariados, (b) padrões monotônicos; (c) padrões arbitrários.

Fonte: SCHAFER e GRAHAM (2002)

Em um conjunto de amostras com padrão monotônico a quantidade de valores ausentes é sempre crescente de uma amostra para a outra. Um exemplo bastante claro deste tipo de padrão encontra-se em RUBIN (1987) que descreve uma pesquisa longitudinal realizada em três etapas com os mesmos entrevistados. Na primeira estapa todos os dados estão presentes. Na segunda etapa, realizada alguns anos depois, como nem todas as pessoas foram encontradas, há dados ausentes. A terceira etapa, realizada após alguns anos, concentrou-se apenas nos entrevistados da segunda etapa. Aqui novamente alguns entrevistados não foram encontrados apresentando um número maior de dados ausentes.

Nesta tese se utiliza dados provenientes de uma única amostra e com padrão de ausência arbitrária.

NEWGARD, HAUKOOS, LEWIS (2006), consideram ainda mais um padrão de ausência, que poderia ser chamado de exclusivo, quando duas variáveis nunca apresentam valores simultaneamente, ou seja, nunca são observadas juntas conforme demonstrado na figura 3.2.

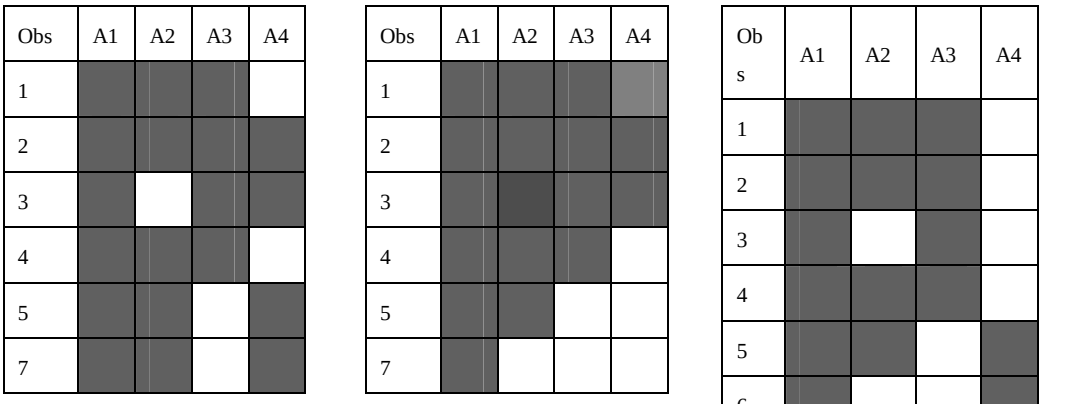


Figura 3.2 Padrões de ausência de dados
Adaptado de NewGard, Haukoos e Lewis (2006)

3.5. Soluções para o tratamento de dados ausentes

Uma vez que a ausência de dados é um fato nas bases existentes no mundo real e causa grande transtorno se não for devidamente tratada, diversas soluções vêm sendo propostas dependendo da área de aplicação e da razão para o tratamento da ausência.

Há duas grandes propostas para tratar os dados ausentes: eles podem ser ignorados e removidos ou podem preenchidos com novos valores.

A primeira solução só pode ser adotada se não acarretar em prejuízo para a análise dos dados. Esta solução pode ser escolhida quando há poucos valores ausentes e muitos casos nas bases de dados. Esta não é a situação usual, pois, em geral, as bases contêm uma quantidade relativamente grande de valores ausentes.

Para substituir os dados ausentes por valores válidos, conhecido como imputação de dados, diferentes técnicas, oriundas principalmente da estatística e da área de inteligência artificial, em particular utilizando os conceitos de aprendizado de máquina, vêm sendo propostas. Os métodos mais antigos de imputação são, em geral, simples e computacionalmente baratos, enquanto os métodos mais novos utilizam procedimentos relativamente complexos levando a um esforço computacional maior e melhores resultados. As novas técnicas desenvolvidas, mais sofisticadas, buscam aumentar a acurácia, e usam como medida de desempenho e/ou qualidade, a comparação com técnicas bem sucedidas já existentes. Cabe ressaltar, no entanto, que o mais importante neste processo é escolher o modelo de imputação compatível com a análise a ser feita e que preserve as relações entre os atributos das tuplas da tabela.

3.5.1 Técnicas e taxonomias para tratamento de dados ausentes

Devido a diversidade das técnicas para o tratamento dos valores ausentes várias taxonomias são propostas na literatura. Cabe ressaltar que não há consenso entre elas exceto na primeira grande divisão: descartar os registros/colunas com valores ausentes ou preenchê-los com novos valores (imputação). Neste segundo grupo, onde todos os métodos têm como meta complementar os valores, as taxonomias variam muito e o número de subdivisões depende do foco e dos conceitos utilizados para a classificação. Por exemplo, em taxonomias mais generalistas, os métodos podem ser organizados considerando apenas os paradigmas que os norteiam e estarem divididos em duas grandes classes: *tradicionais* - utilizando conceitos da estatística (incorporando métodos simples como a média, regressão simples e imputação local (*hot-deck)*, ou mais complexos como a imputação baseada em regressão múltipla ou de máxima verossimilhança) e *baseados em aprendizado de máquina* – embasados em conceitos da área de Inteligência Computacional. Agora, em outras taxonomias, que agrupam estas técnicas considerando o número de iterações para estimar o valor para o atributo

ausente, tem-se: *imputação simples* - onde um único valor plausível é gerado para cada variável com valor ausente no conjunto de dados e, a seguir, a análise é conduzida como se todos os dados fossem originais e a *imputação múltipla* - introduzida por Rubin (RUBIN,1978), onde cada valor é imputado *m* vezes (em geral, $3 \leq m \leq 5$) pelo mesmo algoritmo de imputação incluindo algum modelo para introduzir aleatoriedade. Deste modo são geradas *m* bases "completas" que são combinadas para gerar o valor final.

Portanto, é objetivo desta seção, apresentar algumas das soluções existentes para o tratamento dos valores ausentes, algumas das taxonomias propostas na literatura e finalmente, descrever uma divisão orientada a aspectos, que se considera mais condizente com a realidade.

3.5.2 Taxonomia de SOARES para tratamento de dados ausentes

SOARES (2007) fez uma boa revisão e análise das soluções e taxonomias existentes na literatura para tratar a ausência de dados. Propôs, também, uma nova taxonomia, fortemente influenciada pela classificação de MAGNANI (2004), descrita abaixo e detalhada nas próximas seções:

- Métodos Convencionais: o problema é solucionado pelo descarte das observações/atributos com valores ausentes. A remoção pode ser realizada de três formas distintas:

  Remoção completa de casos

  Remoção em pares

  Remoção de colunas com valores ausentes

- Imputação: os métodos desta classe solucionam o problema estimando os valores ausentes com base nos valores atuais existentes na base. Há três grandes paradigmas:

  Imputação Global Baseada no Atributo Ausente

  Imputação Global Baseada nos Atributos não Ausentes

  Imputação Local

- Modelagem de dados: incorpora todos os métodos que buscam criar um modelo para expressar de forma genérica as características dos dados e com base nestes modelos estimarem o valor ausente. Subdividem-se em:

  Métodos de Verossimilhança

  Modelos Bayesianos

- Gerenciamento Direto dos Dados Ausentes: inclui todos os métodos que são robustos em relação aos valores ausentes, dispensando a imputação.

- Métodos Híbridos**:** reúne os métodos que combinam soluções e/ou sequenciam tarefas para estimar o valor a ser preenchido e são agrupados em duas grandes classes:

  Imputação Múltipla

  Imputação Composta

3.5.2.1 Métodos Convencionais

3.5.1.1.1Remoção Completa de Casos (*listwise deletion ou Complete Case Analysis*)

Caracterizam-se por excluir da amostra todos os registros (ou observações) com valores ausentes em qualquer um de seus atributos (variáveis), limitando a análise apenas aos registros com todos os valores originalmente preenchidos. É a forma mais simples de tratar os valores ausentes, mas apresenta as seguintes desvantagens:

1) A remoção pode tornar a amostra muito pequena, perdendo a precisão;
2) A remoção pode descartar grande parte dos dados, tornando-os tendenciosos;
3) Para tarefas de classificação ou sumarização, a remoção pode também causar a poda de algumas regras, e com isso alterar o suporte e a confiança das regras restantes;
4) Só pode ser aplicada quando o mecanismo de ausência dos dados é completamente aleatório (MCAR).

3.5.1.1.2 Remoção em pares

A remoção em pares (*pairwise deletion*) é uma variante da remoção completa de casos. No entanto, busca preservar a maior quantidade de registros possíveis. A remoção

é instanciada durante cada análise, ou seja, um registro incompleto é considerado válido para a análise de uma variável se possui valor para ela. Este método é frequentemente descrito como sendo baseado na matriz de covariância (ou correlação), onde a variância e covariância são calculadas usando todos os casos completos para um dado atributo ou par de atributos. Em suma, utiliza todos os dados disponíveis na base de dados para a análise de uma variável. Como desvantagens, podem ser citadas:

1) A análise comparativa dentro de um estudo pode ser problemática, pois diferentes subconjuntos de casos são usados para cada uma;
2) Nem sempre a matriz de covariância pode ser definida, isto é, certos elementos podem assumir valores impossíveis dados outros elementos (PEUGH, ENDERS, 2004);
3) Só pode ser aplicada quando o mecanismo de ausência dos dados é completamente aleatório (MCAR).

3.5.1.1.3 Análise Completa de Casos Ponderada

A análise completa de casos ponderada é também uma variante da remoção completa de casos usada geralmente em *surveys* onde casos sem respostas não possuem dados disponíveis. Os casos completos são ponderados para ajustar as perguntas não respondidas e corrigir tendências possíveis.

3.5.1.1.4 Remoção de Colunas Ausentes

Neste método, caso um atributo esteja ausente em algum registro da base de dados, este é removido de todos os registros. A perda de informação resultante deste processo é, em geral, significativa, inviabilizando sua aplicação. É fácil também constatar que esta técnica pode modificar a relação existente entre as colunas na base original e tornar os dados bastante tendenciosos.

3.5.2.2 Imputação

A imputação é um procedimento que visa substituir os valores ausentes por valores estimados. Seu objetivo não é criar valores artificiais, mas deduzi-los a partir das informações existentes na base de dados permitindo que a base resultante, agora completa, possa ser utilizada por qualquer ferramenta de análise de dados e obter inferências estatísticas válidas (HU, SALVUCCI e COHEN, 1998). Existem diversas

maneiras de estimar estes valores. As abordagens mais simples utilizam técnicas estatísticas como a média, para valores contínuos ou a moda para valores categóricos. Embora computacionalmente baratos estes métodos, como já salientado, podem modificar as relações originais dos dados e levar a análises que não reflitam a realidade. Métodos mais sofisticados vêm sendo utilizados e em geral extraem os novos valores baseando-se nas relações existentes entre os atributos. Seu uso tem se mostrado bastante adequado para grandes bases de dados (CARTWRIGHT *et al*, 2003).

Um conjunto de dados com valores imputados não pode ser interpretado como um conjunto de dados originalmente preenchido, já que a imputação oculta a incerteza inerente ao processo. FORD (1983) sugere que os dados imputados devem estar preferencialmente marcados, pois isto permite que eles sejam novamente gerados segundo a vontade do analista de dados que o usa. Devemos considerar também a marcação dos dados ausentes, pois alguns usuários não gostam de usar dados imputados, e outros pelo menos desejam saber se eles existem, e quantos são.

AUSTIN e ESCOBAR (2005) também se referem à adição de um sinal indicador (*flag*) ao lado de cada atributo de uma tupla que apresente valores ausentes. Todavia, esta abordagem enviesa os coeficientes de regressão linear múltipla (AUSTIN e ESCOBAR, 2005 apud JONES, 1996) e seu uso é desencorajado (AUSTIN e ESCOBAR, 2005 apud VACH, 1994).

3.5.2.2.1 Imputação Global Baseada no Atributo com Valores Ausentes

A imputação global baseada no atributo com valores ausentes utiliza todos os valores existentes nas demais tuplas para preencher os que são desconhecidos. Eles podem ser de dois tipos:

1) *Determinísticos*: substituem os valores ausentes pelos valores do centro da distribuição do atributo.
2) *Estocásticos*: introdução de uma perturbação na média visando diminuir os efeitos tendenciosos da média.

A imputação global baseada no atributo com valores ausentes do tipo determinístico é uma das técnicas mais populares. Os atributos contínuos ausentes são substituídos pelo valor médio do atributo na base e os atributos categóricos pela moda, que representa o valor mais freqüente na base para o referido atributo.

No entanto, embora seja uma estratégia simples e barata de preencher os valores ausentes, apresenta duas grandes desvantagens: (a) caso a base não esteja balanceada ou o espectro dos atributos seja muito grande, com ocorrência de valores extremos, a média não representa a realidade da base e os dados tornam-se enviesados pelos valores presentes nos originais (b) a diversidade é diminuída, reduzindo, em geral drasticamente, o desvio-padrão da amostra.

Para que a média não distorça tanto os dados alterando as relações existentes nos atributos, prejudicando a análise, pode-se optar em selecionar apenas uma amostra adequada dos elementos da base ou introduzir uma perturbação no valor calculado. Esta perturbação pode adicionar ou remover um valor Δm da média, tentando reduzir os efeitos descritos anteriormente. Neste caso, a imputação global baseada no atributo com valores ausentes é do tipo estocástico.

TEKNOMO (2007a) faz uso de outras formas de calcular a média para tentar minimizar os problemas acima, entre elas estão:

- a <u>média geométrica</u> aplicável apenas quando todos os dados da amostra são positivos e calculada por:

$$\bar{x} = \sqrt[n]{x_1 * x_2 * \ldots * x_n} = \left(\prod_{i=1}^{n} x_i \right)^{1/n}$$

- a <u>média aritmética ponderada</u> considera pesos w_i para cada componente x_i da amostra em questão introduzindo o conceito de importância relativa dos atributos e calculada por :

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i * x_i}{\sum_{i=1}^{n} w_i}$$

- a <u>média harmônica H</u> representa a capacidade média individual da ação de n elementos que estão agindo harmonicamente. Ou seja, H representa a capacidade de um elemento que é capaz de substituir cada um dos n agentes quando atuando em conjunto:

$$H = \frac{1}{\frac{1}{n}\left(\frac{1}{x_1} + \frac{1}{x_2} + \ldots + \frac{1}{x_n} \right)} = \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{x_i} \right)^{-1}$$

Magnani e Montesi (2004) sugerem o uso da **média balanceada,** uma média de tendência central que é menos sensível à variação de valores extremos. Nesta técnica é introduzida uma tolerância *t*, fornecida pelo usuário, e um fator de corte de 100(1-*t*) %. Por exemplo, se uma tolerância $t = 0{,}03$ for utilizada, valores menores e maiores do que 1,5% são descartados, e a seguir a média aritmética é calculada.

A aplicação da média como forma de preencher os valores ausentes em tarefas que extraem o conhecimento em bases de dados não é uma boa opção, pois elas norteaim-se na diversificação existente na base, e os valores preenchidos expressam um resumo dos dados existentes. São métodos conservadores em relação ao próprio atributo, que diminuem a dispersão dos atributos, podendo alterar as relações entre eles.

3.5.2.2.2 Imputação Global Baseada nos Demais Atributos

A imputação global baseada nos demais atributos estima os novos valores fazendo uso da relação que possa existir entre os atributos da amostra. As técnicas de regressão podem pertencer a esta classe.

Soares (2007) salienta o problema da escolha da técnica de regressão na imputação global baseada nos outros atributos pois (a) em bases com vários atributos ausentes, pode haver valores ausentes também nos atributos de entrada do algoritmo de regressão e (b) a regressão é baseada no fato de que o modelo escolhido é o melhor para os dados, mas nem sempre este modelo existe ou é o mais adequado para representar o comportamento dos dados.

3.5.2.2.3 Imputação Local (Procedimentos hot-deck)

Na imputação local, o valor ausente é estimado a partir de um sub-conjunto completo da base de dados. Os elementos deste sub-conjunto são selecionados utilizando algum critério de similaridade e são considerados os possíveis “doadores” para obtenção do valor ausente. O cálculo do valor ausente, usando os doadores, pode variar e não é determinante na técnica. No entanto, a construção dos grupos, isto é, o subconjunto de doadores, é determinante para o sucesso desta técnica. Como os objetos de um grupo precisam ser similares, a escolha do critério e da métrica para determinar a similaridade tem grande influência na geração de resultados corretos. Uma restrição da técnica é assumir que os registros podem ser agrupados, o que nem sempre é verdade. Porém, na Mineração de Dados esta é uma premissa bastante usual.

A técnica *hot-deck* (FORD, 1983) realiza imputação local e é uma das mais difundidas na imputação. A diferença entre a técnica *cold-deck* e a *hot-deck* está na fonte dos possíveis doadores. Enquanto a técnica *hot*-deck escolhe seus doadores do conjunto de dados correntes, a técnica *cold-deck* utiliza dados de outra fonte, que não os dados correntes.

Entre as razões que impulsionam a utilização da técnica *hot-deck* pode-se citar (MAGNANI, 2004):

1) É possível obter-se uma redução de desvio sem a imposição de um modelo rígido. Ao reduzir o tamanho do grupo, há uma tendência dos dados tornarem-se homogêneos;
2) Produção de um conjunto de dados limpos, sem valores ausentes;
3) Preservação da distribuição da população representada pela amostra.
4) Para alguns valores ausentes, nenhuma informação sobre imputação pode ser encontrada. Isto permite que outras técnicas de imputação possam ser usadas em conjunto;
5) Pode-se usar uma técnica diferente para cada grupo gerado.
6) Não precisa de um modelo robusto para prever valores ausentes
7) Não assume nenhuma distribuição em particular.

O método dos k-Vizinhos Mais Próximos, bastante popular, utiliza a técnica *hot-deck,* e por ser utilizado nesta tese, é descrito a seguir.

O algoritmo dos k-Vizinhos Mais Próximos

O algoritmo dos *k*-vizinhos mais próximos (*k-Nearest Neighbors- k-NN*) (MITCHELL, 1997, AHA, KIBLER, ALBERT, 1991, DASARATHY, 1990) é uma técnica de aprendizado que busca entre os objetos do conjunto de treinamento quais são os *k* objetos mais próximos de um objeto desconhecido. Utiliza em geral uma medida de distância como métrica de similaridade que define a proximidade dos elementos do grupo.(vide apêndice I) É frequentemente utilizado na tarefa de classificação e de imputação.

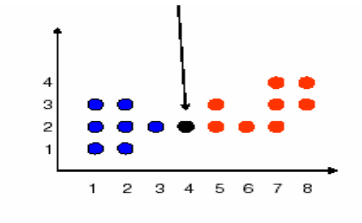
- *O algoritmo dos k-vizinhos mais próximos como classificador:* dada uma tupla i, cuja classe é desconhecida, busca-se no conjunto de n tuplas, as k tuplas mais

similares à tupla i em relação aos atributos conhecidos, e, deste subconjunto selecionado, identifica-se a classe de maior ocorrência (a moda) para atribuir como a classe da tupla i. A similaridade normalmente é medida através do cálculo da distância Euclidiana entre duas tuplas ti e tj:

$$d(t_i, t_j) = \sqrt{\sum_{l=1}^{k} (t_{il} - t_{jl})^2} \ ,$$

onde *k* é o número de atributos das tuplas. Usualmente são escolhidos valores ímpares para o *k* evitando, assim, o empate. Os efeitos de dados com ruídos podem ser eliminados, escolhendo um valor grande para o *k.* Um dos problemas desta técnica é a determinação do valor de k, pois a classificação pode mudar de acordo com o valor escolhido. O exemplo abaixo ilustra a interferência do *k.*

Dado o seguinte conjunto de pontos:



a que classe pertence o ponto preto? À classe azul ou à vermelha?

Para o k=1, nada se pode afirmar.

Para o k=3, tem-se o (5,2), (5,3) e (3,2), portanto a classe será vermelha

Para o k=5, tem-se o (5,2), (5,3), (6,2), (3,2) e (2,2), portanto a classe será vermelha

Para o k=7, tem-se o (5,2), (5,3), (6,2), (3,2) e (2,2), (1,2), (3,2) portanto a classe será azul.

Além da determinação do valor de *k*, caso os objetos de classes diferentes apresentem valores de atributos muito próximos o algoritmo k-NN pode, também, falhar na tarefa de classificação. Todavia, para classes onde haja uma boa distinção entre os campos das tuplas, o algoritmo tem um bom desempenho.

Em uma variante deste algoritmo, escolhe-se um valor t (pré-definido ou definido pelo usuário) tal que (0 < t <= k) e classifica-se a nova tupla apenas quando após

encontrar os k vizinhos mais próximos da tupla a ser classificada, há pelo menos t tuplas em uma das classes (C_i). Caso contrário o exemplo não é classificado.

- *A utilização do algoritmo dos k-vizinhos mais próximos no processo de imputação*: Como a imputação de um valor sempre pode ser interpretada como um problema de classificação onde o atributo com valor ausente é considerado o atributo meta, o algoritmo k-NN é adequado como um regressor de dados contínuos. A adaptação necessária é na escolha do valor da nova tupla, em vez de utilizar a moda, utiliza-se a média. O algoritmo k-NN para imputação de uma tupla t_i: é definido como:

Entrada: Um conjunto de dados com o atributo t_{*k} preenchido, uma tupla t_i com um atributo t_{ik} ausente e o número k de vizinhos da tupla t_i.

Saída: Tupla t_i com o atributo t_{ik} prenchido.

Algoritmo:

Calcule a distância da tupla t_i para todas as demais tuplas t_j, $j \neq i$.

Ordene crescentemente as tuplas pelas distâncias, e selecione as k primeiras.

Atribua à tupla t_i a média dos atributos t_{jk} das k tuplas selecionadas no passo anterior.

Nesta tarefa, também, a escolha do k é vital para o desempenho correto do método. Por exemplo, considere os seguintes exemplos de treinamento:

$x_1 = \{1,2,7,8\}$

$x_2 = \{1,3,9, 40\}$

$x_3 = \{1,2,10,6\}$

e deseja-se imputar o valor do último atributo da tupla x_4:

$x_4 = \{ 1,2,7,?\}$

Inicialmente, calcula-se as distâncias Euclidianas entre as tuplas:

$dist(x_1 , x_4) = 0$, $dist(x_2 , x_4) = 2$ e $dist(x_3 , x_4) = 3$

Agora, para imputar o valor do quarto atributo de x_4, usando, por exemplo, a média dos valores deste atributo em tuplas similares, os k vizinhos mais próximos são selecionados . Considerando:

- k= 1: x_1 está mais próximo de x_4 e o valor estimado será 8, o que parece ser condizente com a realidade.
- k= 2: x_1 e x_2 estão mais próximos de x_4 e o valor estimado será (40+8)/2 = 24, o que parece não refletir a realidade.

Vantagens e desvantagens do algoritmo k-*NN (SOARES, 2007)*

Algumas vantagens da técnica k-NN são:

- É robusto, inclusive para dados com ruídos;
- Funciona bem, mesmo com um grande volume de dados;
- Pode ser usado para regredir tanto atributos categóricos quanto numéricos;
- Não cria modelos explícitos para cada atributo com dados ausentes;;

Já as seguintes desvantagens podem ser enumeradas:

- O custo computacional é bastante elevado, já que devemos computar a distância de cada tupla para todas as outras;
- A determinação de qual o melhor valor de k pode ser um problema bem complexo;
- Como a essência do algoritmo reside na determinação da distância, o quanto a forma de calculá-la afeta o desempenho do algoritmo
- Quais os atributos devem ser utilizados no cálculo da distância entre duas tuplas

JÖNSSON e WOHLIN (2004) mencionam que a vantagem do algoritmo dos K vizinhos mais próximos sobre a imputação por média reside no fato de que a substituição de valores ausentes utilizando o algoritmo *k*-NN é influenciada apenas pelo subconjunto de registros do conjunto de dados que são mais similares à tupla que necessita ter seu valor imputado, e não por todos os registros da tabela. Os autores citam estudos onde a imputação com *k*-NN se comporta tão bem ou melhor do que outros métodos, tanto no contexto de projetos de Engenharia de Software (JÖNSSON, WOHLIN, 2004 SONG, SHEPPERD, CARTWRIGHT, 2005, STRIKE, EL EMAM,

MADHAVJI, 2001) ou convencionais (JÖNSSON, WOHLIN, 2004 CHEN, SHAO, 2000, TROYANSKAYA *et al*, 2001).

3.5.2.3 Modelagem de Dados

Esta classe de soluções para tratamento de valores ausentes engloba técnicas estatísticas e probabilísticas de obtenção de um modelo que consiga representar de forma genérica as características dos dados. As técnicas mais utilizadas desta categoria são os algoritmos de verossimilhança e os métodos bayesianos, que não são o alvo de estudo desta tese.

3.5.2.3.1 Métodos de Verossimilhança

Métodos de verossimilhança são técnicas que procuram estimar os parâmetros de uma função de distribuição estatística, para encontrar um modelo que represente o conjunto de dados, e, com isso ter condições de regredir qualquer valor ausente existente. Entre seus principais representantes estão o EM – *Expectation-Maximization* (DEMPSTER, LAIRD e RUBIN ,1977) e m*étodo de verossimilhança com informações completas* (MYRTVEIT, STENSRUD e OLSSON 2001)

3.5.2.3.2 Imputação de valores ausentes com métodos Bayesianos

A aplicação de redes Bayesianas no contexto da imputação é bastante em relação ao seu uso nas demais áreas, mesmo assim há diversas variantes.

Uma rede bayesiana é um grafo orientado acíclico onde os vértices, ou nós, representam as variáveis do problema, e os arcos, ou arestas, indicam uma relação causal entre as variáveis conectadas. A intensidade de cada relacionamento é dada por uma tabela de probabilidades condicionais.

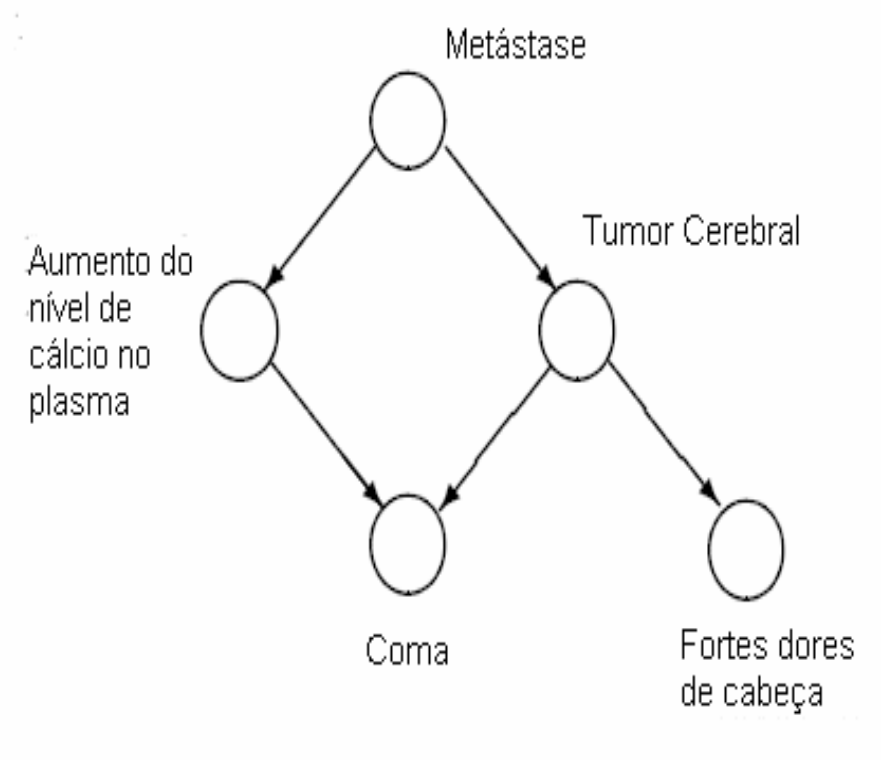


Figura 3.3: Conhecimento médico representado com uma rede bayesiana. Fonte: Adaptado de HRUSCHKA, HRUSCHKA JR. e EBECKEN (*2002b*); *e PARSONS* (*1996*).

Um exemplo de rede bayesiana é o problema de metástase de câncer (SOARES, 2007 apud HRUSCHKA *et al*, 2002b apud PEARL, 1988, PARSONS, 1996), mostrado na figura 3.3. Os nós representam as variáveis (câncer, tumor cerebral, aumento total de cálcio no plasma, coma e dores de cabeça severas) e as ligações entre os nós mostram a influência causal entre as variáveis.

A relação de causalidade só existe entre variáveis conectadas e a intensidade desta influência é dada pela probabilidade condicional $P\left(X_i \mid \Pi_{X_i}\right)$, onde Xi é a i-ésima variável, e Π_{X_i} o conjunto de nós-pai de Xi.

O algoritmo proposto por HRUSCHKA Jr (2003) para imputação em sua tese de doutorado, usando redes bayesianas, baseia-se em busca heurística em base de dados com atributos discretos e no algoritmo Global Bayesian Condicioning (GBC) proposto por PEARL (1988) para inferir valores ausentes adequadamente. Pode ser sumarizado por (MAGALHÃES 2007):

Entrada: Base de Dados com valores ausentes

Saída: Base de Dados imputada

Algoritmo:

1. Identificar o subconjunto X'_1, ... , X'_p. de atributos (variáveis) com valores ausentes
2. Criar uma nova base de dados, removendo todos os registros com valores ausentes em qualquer atributo
3. Para cada atributo alvo:
    a. Definir uma base de dados de aprendizado, para criar uma estrutura de classificação para a variável alvo;
    b. Procure por um atributo *X* que, submetido a um teste específico, gere as saídas O_1, ... , O_n.

Baseado nos valores O_i, o conjunto de testes é particionado em *n* partições disjuntas (T_1, T_2, ... , T_n), onde cada partição T_i contém todas as instâncias de *T* que satisfazem à saída O_i.

Se existirem tuplas com valores ausentes em *X*, utilizar o subconjunto com todos os valores conhecidos de *X* para calcular a taxa de ganho de informação.

Designar pesos nas atribuições de elementos às partições T_i:

a) se um elemento é associado diretamente, após a aplicação do teste, à saída O_i, isto indica que o peso da alocação dele em T_i é 1, e nas demais partições 0.

b) se um elementos não apresenta o valor de X, ele é alocado em todas as partições, com pesos w_j, $1 \leq j \leq n$.

c) Cálculo dos pesos:

$$cálculo\,dos\,pesos = \frac{\sum_{k=1}^{m} \left(w_k \mid E(k)\, satisfaz\, O_i \right)}{\sum_{l=1}^{m} \left(w_l \mid E(l)\, apresenta\, valor\, em\, X \right)}$$

onde *m* é o número de elementos.

O algoritmo K2I é sensível à ordem das variáveis na rede Bayesiana e, por isso, o autor propõe uma variante de seu método inicial, denominado K2I – Qui-quadrado, onde o teste de qui-quadrado é aplicado para ordenar as variáveis.

Outro trabalho que utiliza redes Bayesianas para imputação é o de Di Zio, Scanu e Vicardl (2004), cuja idéia básica pode ser sumarizada por (MAGALHÃES,2007):

Entrada: Base de Dados com valores ausentes

Saída: Base de Dados imputada

Algoritmo:

1. Ordenar decrescentemente as variáveis X_1, X_2 ... , X_p por confiabilidade formando uma partição de v subconjuntos mutuamente exclusivos.
2. Ajustar uma rede Bayesiana de acordo com esta ordenação
3. Para cada ítem ausente de $a = 1$, observar em que partição está a variável:
    a. Se a variável pertecer ao primeiro conjunto, gerar o dado a ser imputado aletaoriamente de acordo com a distribuição marginal da mesma
    b. Nos demais casos, gerar o dado de acordo com a estrutura da rede ajustada em 2.
4. Repetir 3. para $a = 2, ..., n$

Magalhães, 2007, em sua proposta de doutoramento, também utiliza redes Bayesianas para imputação tratando tanto variáveis discretas como de contínuas. No entanto, não avalia a ordem de imputação dentro de variáveis do mesmo tipo. O algoritmo proposto é resumido abaixo:

Entrada: Rede Bayesiana ajustada

Base de Dados com valores ausentes

Saída: Base de Dados imputada

Algoritmo:

5. Identifique os subconjuntos P_0, P_1 ... , P_v na rede Bayesiana de entrada.
6. Defina uma ordem de imputação em cada subconjunto onde as variáveis discretas são imputadas antes das contínuas. Em um mesmo subconjunto, obedecendo o fato das discretas serem imputadas antes das contínuas, qualquer critério de ordenação pode ser utilizado.
7. Para cada subconjunto j = 1,2, ..., v, faça:
    a. Se a variável pertecer ao primeiro conjunto, (ou conjunto das variáveis sem pais) gerar o dado a ser imputado aletaoriamente de acordo com a distribuição marginal da mesma, se esta for discreta ou se for contínua de acordo coma seguinte regra:

$$(X_j|pa_D(X_j), pa_C(X_j)) = X_j \sim N(\mu_j = \beta_{0,X_j}, \sigma^2_{X_j}).$$

    b. Nos demais casos, gerar o dado de acordo com a estrutura da rede ajustada em 2., se discreta ou se for contínua de acordo com as seguintes regras:
        - caso o X_j não tem nenhuma varriável do tipo discreto como pai:

$$\begin{aligned}(X_j|pa_D(X_j), pa_C(X_j)) &= (X_j|pa_C(X_j)) \\ &\sim N(\mu_j = \beta_{0,X_j} + \beta_{i,X_j} x_{i,pa_C(X_j)}, \sigma^2_{X_j});\end{aligned}$$

        - caso o X_j não tem nenhuma varriável do tipo contínuo como pai:

$$\begin{aligned}(X_j|pa_D(X_j), pa_C(X_j)) &= (X_j|pa_D(X_j)) \\ &\sim N(\mu_j = \beta_{0,X_j|pa_D(X_j)}, \sigma^2_{X_j|pa_D(X_j)})\end{aligned}$$

3.5.2.4 Gerenciamento Direto de Dados Ausentes

Há métodos que são robustos em relação aos valores ausentes, dispensando a imputação. O algoritmo de classificação C4.5 (QUINLAN, 1993) é um dos mais populares representante, restringe-se a dados com valores discretos, é robusto em relação à ausência de valores e utiliza para a segmentação o conceito de taxa de ganho de informação da Teoria da Informação. As regras de classificação resultantes do algoritmo sumarizado abaixo e ilustrado na figura 3.4 são deduzidas da árvore de decisão gerada pelo particionamento recursivo dos dados de entrada.

Entrada: Conjunto de treinamento *T*

Saída: Árvore de decisão em um classificador C

Algoritmo:

Procure por um atributo *X* que, submetido a um teste específico, gere as saídas O_1, ... , O_n.

Baseado nos valores O_i, o conjunto de testes é particionado em *n* partições disjuntas (T_1, T_2, ... , T_n), onde cada partição T_i contém todas as instâncias de *T* que satisfazem à saída O_i.

Se existirem tuplas com valores ausentes em *X*, utilizar o subconjunto com todos os valores conhecidos de *X* para calcular a taxa de ganho de informação.

Designar pesos nas atribuições de elementos às partições T_i:

(2) se um elemento é associado diretamente, após a aplicação do teste, à saída O_i, isto indica que o peso da alocação dele em T_i é 1, e nas demais partições 0.

(3) se um elementos não apresenta o valor de X, ele é alocado em todas as partições, com pesos w_j, $1 \leq j \leq n$.

(4) Cálculo dos pesos:

$$cálculo\,dos\,pesos = \frac{\sum_{k=1}^{m}\left(w_k \mid E(k)\,satisfaz\,O_i\right)}{\sum_{l=1}^{m}\left(w_l \mid E(l)\,apresenta\,valor\,em\,X\right)}$$

onde *m* é o número de elementos.

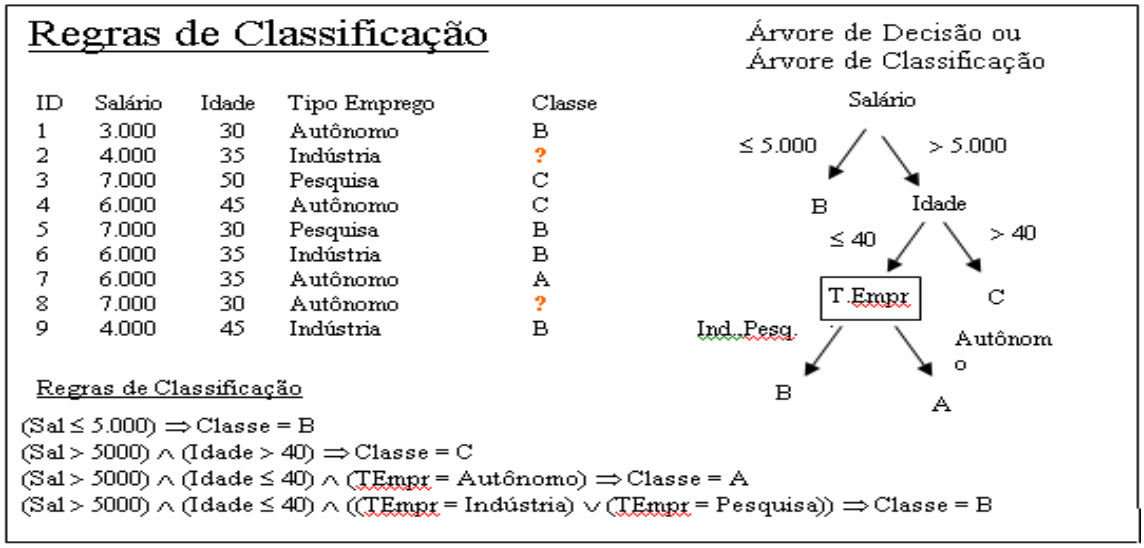


Figura 3.4: Regras de Classificação gerada pelo algoritmo C4.5 para a Determinação da Classe Social de uma Pessoa. Fonte: PLASTINO (2002)

3.5.2.5 Métodos Híbridos

Os métodos de imputação híbrida combinam dois ou mais métodos, com o objetivo de melhorar a qualidade do processo de imputação como um todo. Nesta categoria, estão as técnicas de imputação múltipla, proposta por RUBIN (1988), e de imputação composta proposta por SOARES(2007).

3.5.2.5.1 Imputação Múltipla

Na imputação única, ou imputação simples, apenas um único valor plausível é gerado para cada variável incompleta no conjunto de dados. Embora mais simples, produz valores com pequena variância e pode levar a resultados tendenciosos (CARTWRIGHT *et al*, 2003). Há vários tipos de imputação simples tais como média, moda, regressão simples, regressão estocástica, entre outros. TWALA, CARTWRIGHT e SHEPPERD (2005) salientam que a imputação simples esconde a incerteza do dado e leva a intervalos de testes e confiança inválidos, já que os valores estimados são derivados dos dados existentes.

Rubin define imputação múltipla como: *um processo onde diversos bancos de dados completos são criados pela imputação de valores diferentes para refletir incerteza sobre o valor correto a imputar. No próximo passo, as bases são tratadas pelos procedimentos padrões de análise de bases completas. Por fim, as análises de cada base são combinadas produzindo o resultado final (RUBIN 1986).*

Detalhando a definição acima, o método produz *n* sugestões para cada atributo de um registro que contenha valores ausentes. Cada um destes *n* valores é preenche o campo com valor desconhecido, criando *n* novos conjuntos de dados, como se houvessem ocorrido *n* imputações simples. Desta forma, uma célula ausente da base gera *n* novos conjuntos de dados, que são analisados usando métodos convencionais (figura 3.4). Estas análises são combinadas em um segundo momento, usando um conjunto de regras que levam em consideração a incerteza (em geral a variância) no processo (CARTWRIGHT *et al*, 2003) gerando o resultado consolidado daquele conjunto de dados. WAYMAN (2003) afirma que imputação múltipla reflete a incerteza associada aos valores estimados, produzindo estimativas de parâmetros não tendenciosos. Em suma, pode-se considerar a imputação múltipla como uma extensão da imputação simples, onde cada valor ausente é substituído por um conjunto (n>1) de valores simulados para que a incerteza inerente ao processo de imputação seja reduzida(RUBIN, 1988)

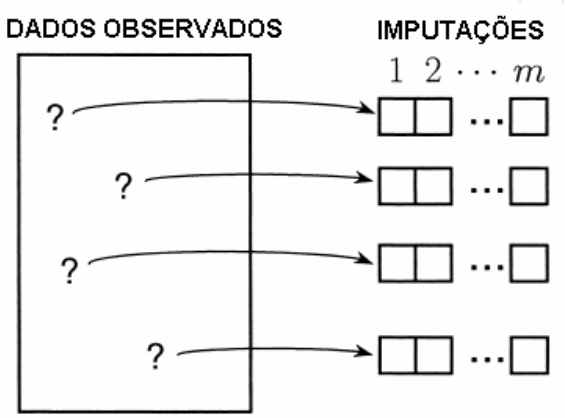


Figura 3.4: Esquema de Imputação Múltipla.
Fonte: Adaptado de SCHAFER e GRAHAM (2002).

Há diversas análises na literatura para o valor de *n*. ALLISON (2000) cita que o valor de *n* varia entre três e cinco, de forma heurística. Já CARTWRIGHT, SHEPPERD e SONG (2003) mencionam que o valor de *n* normalmente varia entre três e dez.

Cartwright e seu grupo apontam como desvantagem e restrição no uso do processo de imputação múltipla o fato do mecanismo de ausência dos dados necessitar ser aleatório (MAR).

(SOARES, 2007) optou por enquadrar a imputação múltipla na categoria de *métodos híbridos*, e não como um tipo de imputação, pois as múltiplas sugestões podem ser feitas por qualquer método de imputação, seja ele simples, baseado em modelos de dados, convencional, ou até mesmo por outro método híbrido.

3.5.2.5.2 Imputação Composta

A imputação composta, proposta por SOARES (2007), representa uma classe de técnicas de imputação que combinam uma ou mais tarefas usadas em KDD, principalmente na etapa de Mineração de Dados, para gerarem um novo valor a ser imputado.

A seqüência de tarefas de KDD com intuito de imputar um valor é denominada de uma estratégia e um plano de imputação é a associação de um algorimto a cada uma destas tarefas. Por exemplo, uma estratégia poderia ser: 1º) selecionar atributos principais; 2º) agrupar casos e 3º) imputar valores. Um plano possível para esta estratégia poderia ser: 1º) selecionar atributos principais com o algoritmo PCA; 2º) agrupar casos com o K-Means e 3º) imputar valores pela média.

Soares (2007) optou por enquadrar a imputação múltipla na categoria de *métodos híbridos*, e não como um tipo de imputação, pois as múltiplas sugestões podem ser feitas por qualquer método de imputação, seja ele simples, baseado em modelos de dados, convencional, ou até mesmo por outro método híbrido.

3.5.3 Imputação Sequencial

Em sua taxonomia, SOARES (2007) manteve o foco em imputação univariada, onde os dados ausentes ocorrem em apenas uma coluna da base de dados. A complementação de uma base de dados que apresenta diferentes colunas com valores ausentes é denominada multivariada. A imputação seqüencial utiliza os métodos de imputação univariada para reduzir a complexidade de problemas multivariados preenchendo sequencialmente cada um de seus atributos incompletos. Portanto, problemas que são multivariados são tratados como N problemas univariados, onde N é o número de atributos que contém pelo menos um valor ausente.

Por exemplo, considere uma base de dados com valores ausentes apresentados como uma matriz Y, com as seguintes colunas: Y_1, Y_2, … Y_k. A imputação seqüencial

inicia processando o atributo Y_1 com base nos demais atributos Y_2, … Y_k. Opcionalmente os valores recentemente imputados para Y_1 podem ser considerados de forma a imputar Y_2. Porém, reutilizando ou não o valor estimado para Y_1, Y_2 será estimado com base em Y_1 + Y_3, … Y_k , e assim sucessivamente até que todas as variáveis com valores ausentes tenham sido complementadas. Este processo é similar, porém não corresponde ao procedimento de imputação iterativa (GELMAN, 2006).

Dois exemplos simples encontram-se apresentados nas figuras 3.5 e 3.6, que, respectivamente, ignoram e reusam valores imputados nas iterações de cada atributo.

Na figura 3.5, cada coluna com valores ausentes é independentemente preenchida em sua própria iteração como um problema univariado, mas os valores sugeridos somente são efetivamente preenchidos após todos os atributos terem sido processados.

Por outro lado, como mostra a figura 3.6, na imputação com reuso de valores todas as sugestões de cada atributo são efetivamente preenchidas antes do processamento do atributo subseqüente.

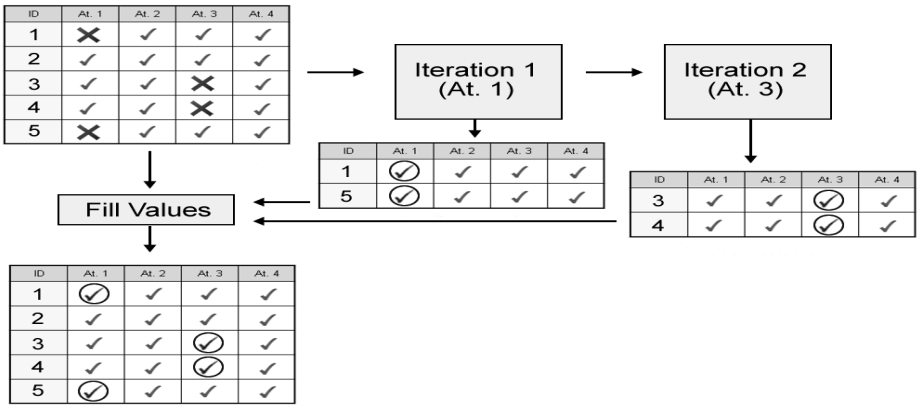


Figura 3.5 – Imputação Seqüencial sem Reuso

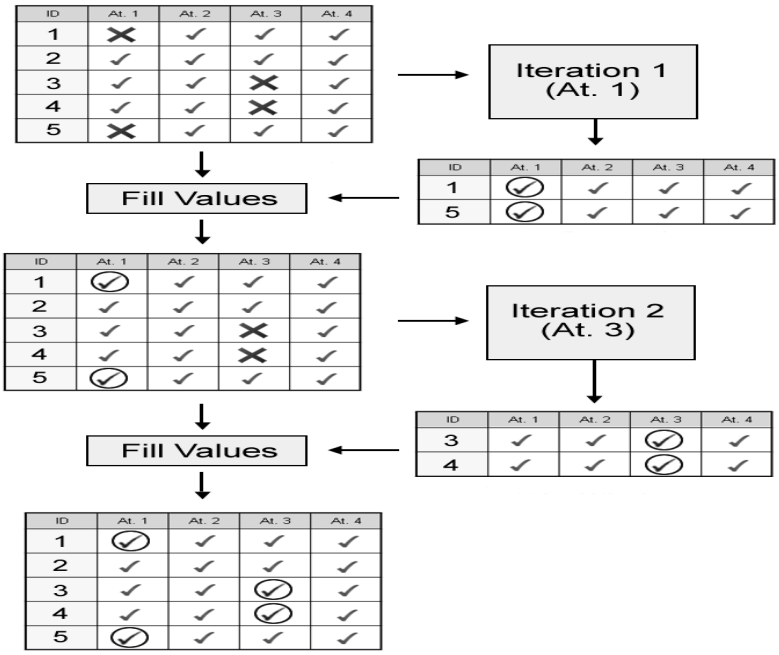


Figura 3.6 – Imputação Seqüencial com Reuso

Gleason e Staelin (1974) são pesquisadores pioneiros na reutilização dos valores imputados. Em sua proposta os valores estimados são aproveitados para recalcular medidas estatísticas, tais como desvio-padrão e covariância, auxiliares na previsão de próximos atributos.

Pesquisadores divergem quanto a utilizar os valores previamente imputados para a estimativa dos próximos valores. De acordo com Oudshoorn et al(1999), um dos grandes problemas de não reutilizar os valores previamente imputados consiste em solucionar as colunas de forma independente, situação que pode introduzir inconsistências como número de cigarros para um não fumante. O autor relata experimentos com imputação seqüencial onde casos de “homens grávidos” foram gerados. A razão afirmada pelos autores para tais acontecimentos reside no fato dos métodos sem realimentação de valores desconsiderarem a correlação existente entre os atributos.

Convém ainda ressaltar que a Imputação Seqüencial pode ser considerada como uma generalização de um método de Imputação Simples, onde este último é aplicado separadamente para cada atributo com valores ausentes da base de dados.

3.5.4 Taxonomia dos métodos de imputação de acordo com Glossário europeu

De acordo com o glossário de termos da area, "U. N. S. Comission and E. C. for Europe. Glossary of terms on statistical data editing" (GENEVA 2000) os algoritmos de imputação podem ser classificados em seis categorias:

- Imputação determinística: na qual há uma resposta precisa e correta para cada valor ausente, como por exemplo, uma totalização de atributos de uma coluna da base;
- Imputação baseada em modelos**:** Esta é uma extensa categoria que inclui desde médias e medianas até sofisticados modelos estatísticos e equações de regressão;
- Imputação "*Deck-Based*" - O valor ausente para um caso é derivado de possíveis casos doadores (podendo estes ser ou não completos). Quando os doadores são casos da mesma base de dados é denominado "*hot-deck*" e quando são de outras bases do mesmo domínio é conhecido como "*cold-deck*". A técnica de busca "Vizinhos mais Próximos" é geralmente aplicada para encontrar os melhores doadores;
- Imputação mista - implementações híbridas que combinam diversas técnicas de imputação em uma determinada ordem;
- *Expert Systems* - sistemas que simulam o comportamento de um especialista humano e possuem um mecanismo de inferência capaz de responder questões a partir de bases de conhecimentos Uma base de conhecimento pode ser, por exemplo, uma coleção de afirmativas do tipo Se/então;
- Redes Neuronais – imputação baseada em redes neuronais como Back-propagation ou SOM.

3.5.5 Classificação dos métodos pelos apectos utilizados

FARHANGAR, KURGAN e PEDRYCZ (2007), ao propor um *framework* para imputação (descrito na seção 3.6 sobre trabalhos relacionados), realizaram uma análise bastante interessante de diferentes métodos para tratamento de dados ausentes, categorizando-os em função de diferentes critérios. As duas categorias mais genéricas propostas por eles, onde há consenso entre os pesquisadores, são: 1) remoção dos valores ausentes e 2) imputação dos dados ausentes. Na remoção de valores ausentes, os

registros ou atributos com valores ausentes são descartados. A remoção de atributos só pode ser realizada quando estes não são necessários para a análise dos dados. No entanto, ambos resultam em perda de informação dos dados e só podem ser usados com o padrão de ausência MCAR, quando há muito poucos dados ausentes ou quando é seguro que a análise dos dados restantes não ficará tendenciosa (LAKSHMINARAYAN et al 1999). Um outro método que pertence a esta categoria propõe substituir os valores ausentes em cada atributo por uma nova categoria. Embora este método seja de fácil implementação seu uso resulta na ocorrência de problemas em análises subseqüentes (TRESP 1997). A segunda grande classe, imputação de valores ausentes, possui muitas técnicas. Um critério de agrupamento proposto pelos autores é o número de vezes que o valor a ser imputado é calculado o que leva a classificar os métodos como métodos de imputação simples e métodos de imputação múltipla. Os autores também propõem como critério de agrupamento a orientação dos métodos, levando a agrupá-los em três categorias: 1) dirigidos pelos dados, 2) baseados em modelos, 3) baseados em aprendizado de máquina. Sendo assim, tanto métodos de imputação simples quanto de imputação múltipla podem pertencer a qualquer uma destas três categorias. Os métodos dirigidos pelos dados utilizam apenas os dados completos para computar os valores. Estão nesta categoria, média, média condicional, *hot-deck*, *cold-deck* e imputação por substituição (aplicável somente em *surveys*). Métodos baseados em modelos utilizam algum modelo de dado para gerar os valores a imputar. Assumem que os dados são calculados por algum modelo governado por parâmetros desconhecidos. Nesta categoria encaixam-se, entre outros, os métodos de imputação baseados em regressão, em verossimilhança, em análise de discriminante linear. Nos métodos baseados em regressão, os valores ausentes de um registro são imputados pela criação de um modelo de regressão sobre os valores dos atributos preenchidos deste registro. Estes métodos exigem múltiplas equações de regressão, uma para cada conjunto diferente de atributos completos levando a um alto custo computacional. Também diferentes modelos de regressão devem ser criados para diferentes tipos de dados: em atributos contínuos aplicam-se modelos lineares ou polinomiais e para atributos discretos, modelos log-linear (LAKSHMINARAYAN et al 1999). Métodos baseados em verossimilhança são aplicáveis só em atributos discretos, pois assumem que os dados são descritos por um modelo parametrizado, onde os parâmetros são estimados pela máxima verossimilhança ou procedimentos de máximos *a posteriori* usados em variantes do algoritmo EM.

Métodos baseados em aprendizado de máquina são mais recentes. Nesta categoria estão, entre outros, os modelos que utilizam redes neuronais, regras de associação, e métodos de imputação probabilística que usam estimativas de densidades de probabilidade e a abordagem Bayesiana. Algoritmos de aprendizagem supervisionada também são usados quando, neste caso, a imputação é executada serialmente e o atributo a preencher é utilizado como alvo. Entre eles podem ser citadas as árvores de decisão, regras de decisão e probabilísticas. Novos métodos têm surgido visando melhorar a acurácia do processo de imputação e tendem a ser híbridos e compostos por uma seqüência de tarefas, o que aumenta o custo computacional. Com o rápido crescimento das bases de dados estes métodos tornam-se inaplicáveis em situações reais. Portanto, novos métodos devem ser também escaláveis (FARHANGAR 2007)

Observando-se os diferentes métodos de imputação existentes na literatura e em concordância com Farhangar (2007), acredita-se que os mesmos pertençam a diferentes categorias dependendo do aspecto que está sendo considerado. Isto significa que um mesmo método pode ser classificado em diferentes categorias. Esta múltipla classificação, que é dependente do aspecto, dificulta a definição de uma taxonomia em sua forma tradicional, como uma estrutura hierárquica do tipo árvore. Portanto, no trabalho realizado nesta tese, salienta-se os seguintes aspectos para classificar os métodos de imputação:

- **Imputação Univariada x Imputação Multivariada:** na imputação univariada, apenas um atributo da base possui valores ausentes, enquanto na multivariada, diversos atributos podem apresentar valores ausentes na mesma tupla ou em tuplas diferentes.
- **Imputação Simples x Imputação Múltipla**: considera-se a quantidade de simulações dos valores estimados. Na imputação simples, o valor estimado é simulado apenas uma vez, enquanto na imputação múltipla o valor estimado é resultante da combinação de um conjunto de $m>1$ valores simulados e armazenados em m bases de dados completas.
- **Dirigidos pelos dados x Baseados em modelos x Baseados em aprendizado de máquina**: Os métodos dirigidos pelos dados utilizam apenas os dados completos para computar os valores. Métodos baseados em modelos utilizam algum modelo de dado (em geral, estatístico) governado por parâmetros

desconhecidos para gerar os valores a imputar. Já os métodos baseados em aprendizado de máquina utilizam técnicas capazes de inferir automaticamente, a partir dos dados disponíveis, os valores ausentes.

- **Seqüencial x Simultânea**: Neste aspecto considera-se o paralelismo para a geração do valor ausente. Na imputação seqüencial, os atributos ausentes em uma tupla são estimados um a um, enquanto na imputação simultânea, todos os atributos ausentes da tupla são gerados ao mesmo tempo.
- **Casos Completos x Casos Incompletos**: o foco está nas tuplas que serão aproveitadas para a imputação. Nos casos completos, todas as tuplas que possuem algum valor ausente são descartadas na geração do novo valor, enquanto nos casos incompletos qualquer tupla pode ser aproveitada.
- **Com reutilização x Sem reutilização**: considera-se o uso dos valores previamente imputados. As técnicas com reutilização aproveitam valores previamente imputados para a geração de valores conseguintes.
- **Métodos Híbridos**: utilizam combinação de tarefas de KDD para realizar a imputação

3.6.Trabalhos relacionados

A imputação é muito necessária e bastante aplicada em pesquisas em geral ou que envolvam estudos longitudinais. Estas tarefas costumam produzir dados categóricos, ordinais, conhecidos como *likert data* representando o nível de concordância/discordância de um entrevistado com uma determinada pergunta. Por exemplo, questões do tipo “Qual sua opinião sobre o desempenho de um político x?” podem apresentar como respostas uma das seguintes alternativas: “Péssima/ Ruim / Satisfatória / Muito Boa”. Os trabalhos de JÖNSSON e WOHLIN (2004) e JÖNSSON e WOHLIN (2006), onde a idéia de casos incompletos utilizada nesta tese foi introduzida, trabalham com dados deste tipo no contexto Engenharia de Software. Aplicam os métodos dos K vizinhos mais próximos para imputar os valores. Em sua forma clássica, o algoritmo utiliza todas as variáveis no cálculo da distância entre dois objetos, o que restringe a escolha dos doadores aos registros com todas as variáveis

preenchidas. Eles chamam esta abordagem de casos completos (CC – *complete cases*). Todavia, definem um modo de considerar como possíveis doadores tuplas parcialmente completas. Para que uma tupla incompleta possa ser considerada, ela deve conter valores nos mesmos atributos que o objeto sendo imputado, além de possuir valor no atributo que está sendo imputado, é óbvio. Esta estratégia é chamada de casos incompletos (IC – *incomplete cases*).

Para ilustrar esta idéia, considere o exemplo abaixo:

id	a_1	a_2	a_3	a_4
1	3	8	6	4
2	*	5	3	*
3	1	*	*	2
4	*	3	*	3
5	*	7	1	6

Para regredir o atributo a_3 do registro de identificador 4, a abordagem *CC* retornaria a tupla 1. Já a abordagem *IC* retornaria também a tupla 5, pois ela apresenta valores completos nos mesmos atributos da tupla 4 (a_2 e a_4), além do atributo a ser imputado (a_3). A tupla 2 não foi selecionada pela abordagem *IC* pois não apresenta valor no atributo a_4 e a tupla 3 porque o atributo a_2 está ausente.

Os autores consideram que os dados possuem mecanismo de ausência completamente aleatório (MCAR), e não reutilizamo dos valores previamente imputados.

O trabalho conclui que a imputação de dados de pesquisas com o algoritmo *k*-NN é viável e produz bons resultados qualitativos, dependendo mais do número de casos completos do que o de valores ausentes. Quanto ao valor de *k*, problema inerente ao método, os autores constatam que nenhum experimento teve bons resultados com o valor de $k = 1$. Também de forma heurística, eles sugerem que os melhores resultados foram obtidos com o valor de *k* aproximadamente igual à raiz quadrada dos casos válidos, arredondada para o inteiro ímpar mais próximo.

CARTWRIGHT, SHEPPERD e SONG (2003) avaliam duas técnicas de imputação de dados também no contexto de projetos de Engenharia de Software: a média amostral e o algoritmo do *k*-vizinhos mais próximos (*k*-NN) e os comparam com a remoção completa de casos (*listwise deletion*). O trabalho utiliza duas bases de

projetos de softwares, uma proveniente de um banco de investimentos, com 17 tuplas, e outra de uma multinacional, com 24 tuplas. Ambas as bases possuem nos experimentos 15% e 18% de dados ausentes. Os autores alegam que estas condições são desafiadoras para técnicas de imputação. Os dados das bases são na maioria numéricos porém como há dados categóricos, os autores propoem uma modificação no algoritnmo k_NN para tratá-los adequadamente. Os parâmetros do *k*-NN foram ajustados para $k = 2$ e distância Euclidiana. Segundo os autores esta métrica produz os melhores resultados. O trabalho chega à conclusão de que o algoritmo *k*-NN apresentou uma melhor desempenho do que a utilização da imputação com a média, e ambas as opções são melhores do que deixar a base sem tratamento. Porém, os autores mencionam que não há um resultado conclusivo, e que mais testes precisam ser feitos.

Os mesmos autores em 2005 (SONG, SHEPPERD e CARTWRIGHT, 2005) extendendo o trabalho anterior, experimentam comparar a imputação com média na classe (CMI – *Class Mean Imputation*) e o algoritmo dos *k*-vizinhos mais próximos, assumindo em momentos distintos que os dados são completamente aleatórios (MCAR) e aleatórios (MAR), e chegam à conclusão de que a imputação por média na classe, para os problemas tratados, é mais adequada por ser mais precisa. Todavia, os autores mencionam que as duas técnicas têm aplicações práticas para conjuntos de dados pequenos de Engenharia de Software.. Outro destaque é a da não importância estatística do mecanismo de ausência de dados, que, por esta razão, podem ser sempre assumidos como sendo aleatórios (MAR).

Regras de associação também são encontradas como solução para o tratamento de valores ausentes. Como exemplo, o trabalho de CRÉMILLEUX, RAGEL e BOSSON (1999) e RAGEL e CRÉMILLEUX (1999), que apresentam o algoritmo MVC – *Missing Values Completion*. Esta técnica exige a intervenção do usuário no processo de complementação de dados, pois após criar o conjunto de regras de associação, estas precisam ser submetidas à apreciação. Sendo assim, o usuário decide qual regra ou quais regras utilizar, baseado em alguns parâmetros, onde os principais são: a confiança e o suporte de cada uma das regras. O método trabalha apenas com valores discretos, mas não descreve a forma como foram discretizados os atributos numéricos.

BATISTA e MONARD (2003a) comparam com a média ou moda, C4.5 (QUINLAN, 1993) e CN2 (BOLL, ST. CLAIR, 1995) o desempenho do algoritmo dos *k*

vizinhos mais próximos (*k*-NN) como método de imputação. O C4.5 gera uma árvore de decisão sobre um atributo-classe. Já o CN2 gera regras de associação, assumindo o valor da moda nos atributos ausentes antes de medir a entropia. Estes dois algoritmos são robustos em relação à ausência de dados exceto para o atributo-classe. O intuito do trabalho é analisar se o tratamento dos dados em um conjunto que possui valores ausentes melhora o processo de classificação dos dados. Portanto, um sub-conjunto de dados tem os valores ausentes preenchidos com o algoritmo *k*-NN, e o outro subconjunto com a média ou moda. Cada um destes conjuntos é então enviado aos algoritmos C4.5 e CN2 para classificação, e seu desempenho comparado com estes dois algoritmos no conjunto de testes.

A base, inicialmente completa, tem valores sujos artificialmente, com percentuais de ausência variando entre 10% e 60%, com saltos de 10%. Os autores também sujaram um, dois e três atributos nas bases, gerando diferentes conjuntos de dados, resultado da combinação de percentual e número de atributos de valores ausentes. Os dados ausentes são MCAR. O número de vizinhos utilizados no *k*-NN foram 1, 3, 5, 10, 20, 30, 50 e 100.

Os melhores resultados foram alcançados com a aplicação do *k*-NN com 10 vizinhos. O algoritmo C4.5, à medida que a ausência aumenta, descarta gradualmente os atributos incompletos. O C4.5 também descarta valores que foram imputados com média ou moda, já que eles diminuem o seu poder discriminatório (entropia). Os modelos de classificação simplificaram-se a medida que atributos com valores ausentes aumentam. Assim, concluem que a imputação prévia de dados ausentes pode evitar que o modelo gerado de um conjunto de dados torne-se extremamente simplificado. Por outro lado os autores, em função de seus resultados na *Breast Cancer*, questionam se realmente os atributos devam ser tratados com algum método de imputação.

HRUSCHKA, HRUSCHKA JR. e EBECKEN (2003a) propõem agrupar os objetos por meio de um algoritmo genético por eles proposto e denominado CGA – *Clustering Genetic Algorithm*., antes da imputação dos dados ausentes com a utilização do vizinho mais próximo (com $k = 1$). Experimentaram seu trabalho na base *Wisconsin Breast Cancer*, do repositório da Universidade da Califórnia, Irvine (MERZ 1998). A cada vez, apenas um atributo apresenta valores ausentes e todos são numéricos. A base utilizada possui um atributo classificador, que indica se o paciente está ou não com a

doença. Este atributo foi desconsiderado pelo algoritmo CGA na formação do grupo. Para fins de comparação, os autores também realizam a imputação com a utilização da média aritmética simples. A qualidade do dado imputado também é medida pela reclassificação das tuplas com valores preenchidos e a taxa média de acertos na base imputada é comparada com a taxa média de classificação dos dados originais. Os resultados mostram que o algoritmo do vizinho mais próximo foi melhor do que a média em todos os casos. A taxa média de classificação das tuplas imputadas também foi bastante satisfatória com o algoritmo k-NN, obtendo taxas de acerto variando entre 94.44% e 95.46%.

O trabalho de MAGNANI e MONTESI (2004) propõe um método qualitativo de imputação local que leva em consideração todos os registros de uma tabela, procurando aproveitar toda a informação contida nos registros do conjunto de dados para diminuir as chances do conjunto de dados se tornar tendencioso. O método considera que um conjunto de dados representa objetos. Estes, por sua vez, pertencem a conceitos. Conceitos são estritos (possuem uma função característica), e são organizados em classes hierárquicas ou nebulosas (funções membro). Um conceito possui uma única descrição baseada na relevância dos seus atributos. Alguns erros no conjunto de dados não mudam a estrutura do conceito de uma forma significativa.

O algoritmo de agrupamento utiliza uma função de similaridade descrita como:

$$f(x_{ik}, x_{jk}) = \frac{\sum_{k=1}^{m} w_k * sim(x_{ik}, x_{jk})}{\sum_{k=1}^{m} w_k},$$

onde:
$$sim(x_{ik}, x_{jk}) = \begin{cases} 0, se(x_{ik} \neq x_{jk}) \ (para\ atributos\ categóricos) \\ q_k, se(x_{ik} = ?) \vee (x_{jk} = ?) \\ 1, se(x_{ik} = x_{jk}) \\ 1 - \dfrac{\left| x_{ik} - x_{jk} \right|}{\left| \max_{\in [1,n]}(x_{pk}) - \min_{\in [1,n]}(x_{pk}) \right|}, caso\ contrário \end{cases}$$

O parâmetro *q* permite que o algoritmo seja mais ou menos restritivo com a importância dos valores ausentes dos registros.

O artigo utiliza para efetuar a imputação uma média chamada central, calculada para valores numéricos ausentes, em tuplas dentro de um grupo, que é menos sensível a valores completos do atributo muito distintos dos demais. Esta média é calculada

segundo a fórmula 100*(1-t)%, onde *t* é a tolerância fornecida como parâmetro. Este parâmetro *t* também é utilizado para tratar valores ausentes.

O trabalho de SCHEFFER (2002) mostra como a média e o desvio-padrão são afetados por diferentes métodos de imputação, considerando a existência de diferentes mecanismos de ausência de dados. Os autores sugerem que melhores opções do que as convencionais estão disponíveis em pacotes estatísticos possibilitando a obtenção de melhores resultados. Os resultados obtidos sugerem que: a utilização da média é a pior opção sempre; a remoção de casos incompletos é ruim, por destruir a variância inerente ao conjunto de dados e os métodos de imputação simples podem funcionar com dados com mecanismo de ausência aleatória (MAR), mas somente com até 10% de valores ausentes na base. Os autores também tecem as seguintes recomendações: não se deve usar remoção completa de casos se o seu mecanismo de ausência não for completamente aleatório (MCAR); tentar evitar o acontecimento de dados ausentes; se a regressão simples tiver de ser usada, use a imputação por regressão ou o algoritmo EM; sempre que possível use imputação múltipla e quando usar a imputação múltipla, use um modelo compatível ao modelo de análise onde for possível.

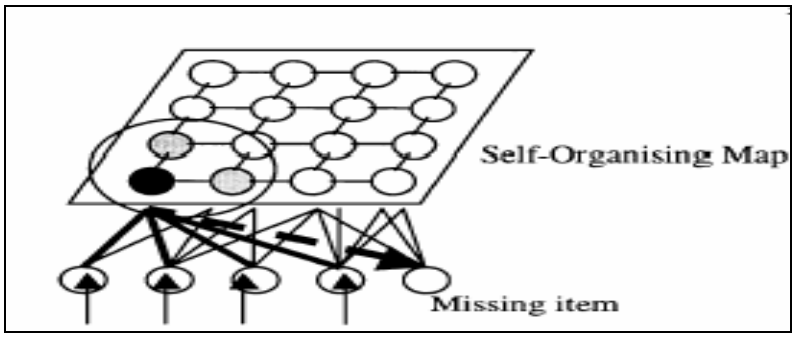


Figura 3.6: Modelo SOM aplicado na imputação
Fonte: FESSANT e MIDENET (2002)

FESSANT e MIDENET (2002) aplicaram redes neurais SOM para imputação de dados numa pesquisa com o objetivo de descrever os hábitos de locomoção de donas de casa que moram na França e como elas usam os meios de transporte público e particular. Compararam os resultados com os obtidos com imputação por média, “hot-deck” e um modelo MLP de rede neuronal *(feed-foward multi-layered perceptron*). O processo de imputação seguiu os seguintes passos, ilustrados na figura 3.6:

1) Apresentação da tupla incompleta na camada de entrada

2) Seleção do nó-imagem que minimiza a distância entre a tupla incompleta e protótipos considerando apenas os atributos disponíveis. Os atributos com valores ausentes foram ignorados.

3) Seleção do grupo de ativação composto pelos vizinhos do nó imagem.

4) Determinação do valor do atributo ausente baseando-se nos pesos dos nós do grupo de ativação nos atributos ausentes.

A rede foi treinada com os registros originais completos para ser consistente com os demais métodos. Como a base é originalmente desbalanceada (75% das viagens utilizam carro como meio de transporte) foi necessário um ajuste no número de registros. Cada tupla foi apresentada 100 vezes e afirmam que o SOM não é muito sensível a este parâmetro. No entanto, mostrou-se extremamente sensível ao número de nós e o esquema de codificação das tuplas. Sendo assim, as variáveis contínuas foram normalizadas, cada uma representado por um nó no mapa de entrada. Para as variáveis categóricas foi associado um nó de entrada para cada valor possível. Os resultados entre os métodos foram similares. A MLP (por atributo) apresentou melhores resultados para variáveis contínuas, mas cabe lembrar que não foram criadas redes independentes para cada atributo. Por ser um modelo integrado necessitou de menos tuplas para treinamento sendo que estas, inclusive, poderiam ser incompletas.

Wei Wei e Ying Tang (WEI eTANG, 2001) propuseram uma arquitetura genérica para sistemas de preenchimento de valores ausentes para mineração de dados que utilizam redes neuronais. A arquitetura é dividida em três unidades:

- *Unidade de pré-processamento:* como os dados utilizados para a mineração podem ser provenientes de diferentes fontes e estarem armazenados em diferentes formatos, a primeira tarefa a realizar é representá-los em um formato único, padrão. No sistema por eles proposto as estruturas originais nas quais os dados estão representados são desfeitas e os mesmos são armazenados linearmente, separados por vírgulas (sem nenhuma estrutura), os dados ausentes ou corrompidos são marcados, as dimensões dos conjuntos de dados são determinadas e cada registro é identificado por um ID.

- *Analisador da base de dados:* Este módulo prepara a base de dados para ser usada pela unidade que imputa os valores. Portanto, é responsável em determinar: (a) o tipo do atributo (categórico ou contínuo); (b) a importância do atributo, permitindo

que atributos não importantes sejam descartados pelo próximo módulo, eliminando ruídos e diminuindo a dimensão da base; (c) o valor padrão para inicializar os atributos ausentes; (d) prioridade de preenchimento dos registros.

Para atingir seus objetivos, este módulo primeiro avalia as características da base de dados em dois sentidos: horizontal, ou seja, por registro e vertical, ou seja, por atributo. O percentual de dados ausentes em cada atributo é contabilizado para determinar a prioridade do preenchimento. O registro com maior percentual de atributos ausentes é considerado o de mais alta prioridade.

A importância do atributo é refletida em um peso associado a ele e é proporcional ao quão próximo seu padrão de distribuição está da distribuição uniforme. Quanto mais próximas forem as distribuições, menor é a importância do atributo e menor será o valor do peso associado. O atributo é descartado quando sua distribuição é exatamente igual à distribuição uniforme.

- *Unidade de preparação dos dados:* A rede SOM é usada para preencher os dados ausentes através de um processo de imputação em dois níveis. Inicialmente, todos os atributos ausentes que não foram descartados são preenchidos com o valor padrão inicial determinado pelo módulo anterior (no experimento foi usada a média para os atributos contínuos e para os atributos categóricos, a moda.). Em segundo lugar, seguindo a prioridade de preenchimento, a SOM separa os registros ponderados em vários grupos. Os grupos gerados pela SOM com densidade superior a um limiar pré-determinado, são sub-divididos. Quando este processo iterativo de divisão finaliza, o valor de cada atributo originalmente ausente e pré-preenchido com o valor padrão é substituído pelo valor do atributo correspondente no subconjunto.

  O algoritmo de preenchimento proposto para a imputação é detalhado a seguir:

  1) Preencher os valores ausentes com os valores padrão;
  2) Aplicar os algoritmos K-S e Qui-Quadrado para determinar o fator de peso de cada atributo (pela comparação da similaridade/distância dos padrões de distribuição dos atributos categóricos e contínuos respectivamente);
  3) Modificar os valores originais da base de dados, multiplicando cada atributo por seu fator de peso;
  4) Classificar os atributos em ordem descendente de suas prioridades;

5) Inicializar a SOM;
6) Para cada atributo
    i. DS = toda a base de dados
    ii. Treinar a SOM com uma passagem única em DS
    iii. Determinar a densidade de cada subgrupo gerado pela SOM
    iv. Se a densidade de qualquer subgrupo é superior ao limiar, DS= subgrupo e retorna ao passo b)
    v. Em cada subgrupo, para cada vetor com valor ausente neste atributo:
7) Encontrar o mais próximo vetor com valor válido para o atributo(usando como critério a distância Euclidiana)
8) Preencher o atributo ausente do seguinte modo: novo valor = β * valor padrão deste atributo no sub-grupo + (1- β) valor do atributo no vetor, onde β é um fator de ajuste

Os autores testaram a arquitetura proposta em uma base real de cartões de crédito, com quase 500.000 registros e 110 atributos cada um, mas não apresentam os resultados obtidos. Afirmam que o método aumenta a acurácia no preenchimento dos dados e diminui a complexidade em termos de tempo e espaço em grandes bases de dados. Para diminuir a dimensionalidade da base e antes de sua utilização pela SOM, é utilizado o peso associado aos atributos e o conceito de distância.

Zhu, Zhang, Zhang, Zhang, (ZHU, ZHANG, ZHANG, ZHANG, 2006) apresentam duas estratégias para determinar a ordem de imputação dos atributos visando minimizar o custo da imputação e maximizar a acurácia em problemas de classificação. Neste trabalho, o foco está na ordenação dos valores e qualquer método de imputação pode ser aplicado. No primeiro método, conhecido como Estratégia Iterativa Incremental, o último valor imputado é adicionado ao conjunto de treinamento e utilizado na imputação dos demais valores. Este processo é repetido até que não haja mais modificação na acurácia. No segundo método, Estratégia Iterativa, cada valor ausente é imputado utilizando todas as informações disponíveis na base de dados, inclusive os registros com valores ausentes. Para isto, os atributos com valores ausentes são inicializados com a média/moda, respectivamente para atributos contínuos/categóricos.

Após a inicialização, o processo é repetido até que não haja mais melhora na acurácia. Ambos os métodos baseiam-se em duas medidas:

- *Critério Econômico (EC)*: usado para achar o atributo mais econômico e determinar a seqüência em que os atributos serão imputados. É uma relação entre custo e benefício e definido como: EC=custo/MI, onde custo é a soma de todos os custos, ou seja, o custo de imputação e de classificação inválida. É visto como uma unidade financeira, por exemplo, um dólar. MI é a informação mútua utilizada por Quinlan (QUINLAN, 1986) para a seleção de atributos em árvores de decisão. Os atributos com baixo MI em relação a uma determinada classe tem menos chances de serem decisivos na processo de classificação e por isso seu preenchimento terá pouco impacto na acurácia da classificação e vice-versa.
- *Informação Eficiente (EI):* é o percentual de todas as instâncias que podem ser usadas para construir o modelo de imputação da base. Para cada valor ausente, um modelo de imputação é construído usando o máximo possível das informações observadas de modo a obter um desempenho ótimo. Quanto mais informações observadas há para um valor ausente, mais confiante são considerados os resultados da imputação. Deste modo o EI para o valor ausente no j-ésimo atributo da i-ésima instância é definido como:

$$\mathrm{EI}_{\mathrm{i,\,j}} = \frac{\text{Informação Útil para os valores ausentes}}{\text{Todas informação na base de dados}}$$

e, como pode ser observado, quanto maior o valor de EI melhor o desempenho do sistema.

O critério para ordenação dos valores a imputar utiliza a média harmônica destas duas medidas (EC e EI) e é definido como: $Rank(i,j) = \frac{(\alpha+1)\mathrm{EI}_{\mathrm{i,\,j}} * \mathrm{EC}_{\mathrm{j}}}{\mathrm{EI}_{\mathrm{i,\,j}} + \alpha \mathrm{EC}_{\mathrm{j}}}$

o coeficiente $\alpha \in (0, +\infty)$ permite modificar a importância dos termos no critério de ordenação. Após computar este critério para cada valor ausente, os mesmos são ordenados em ordem decrescente.

Considerando as seguintes definições:

- MVS: conjunto de valores ausentes
- OS: conjunto de instâncias observadas

- CIV: valor ausente em imputação (valor corrente), ou seja, o com maior *Rank(i,j)*
- CS: informação completa do valor ausente atualmente imputado (CIV)

As estratégias podem ser descritas do seguinte modo:

1) Estratégia de Imputação Incremental

Faça

Enquanto tem elementos em MVS

Construir um classificador para o CIV considerando o CS

Imputar o valor baseado neste classificador

MVS=MVS – CIV

OS= OS + CIV

Calcular a acurácia na classificação (CA)

Atualizar o número de iterações

enquanto a acurácia atual é maior que a acurácia da iteração anterior.

2) Estratégia de Imputação Iterativa

Para a primeira imputação

Para cada valor ausente

Imputar com a média para os atributos contínuos

Imputar com a moda para os atributos categóricos

Para as demais imputações

Método equivalente à Estratégia de Imputação Incremental

Os autores aplicaram as estratégias propostas em 4 bases da UCI: “Abalone”, “Ecoli”, “Pima” e “Vowel”, para 10%, 20% e 30% de ausência e os compararam com o método de ordenação, desenvolvido por Cláudio (2003), obtendo um melhor desempenho tanto no número de iterações necessárias como na medida de erro utilizada. Compararam, também os três métodos que determinam uma ordem inicial com um método que desconsidera a ordem sendo que os primeiros foram bem mais eficientes.

Wong e Graham Wood (2007), no contexto da BioInformática, propuseram uma estratégia para ser aplicada tanto na tarefa de agrupamento como de imputação para expressões gênicas. A idéia usada é decompor os “*profiles*” em componentes ortogonais e o agrupamento ocorrer em níveis, onde, em cada nível (ou estágio) um componente

mais específico é o determinante no critério de agrupamento. Geometricamente, o espaço S-dimensional dos dados é separado em *n* subespaços ortogonais. Nesta proposta, o agrupamento é realizado em estágios usando uma hierarquia de medidas de distâncias. Estas medidas estão relacionadas ao domínio do experimento e refletem atributos diferentes dos dados. A hierarquia inicia com o atributo mais dominante e continua com os demais atributos até os atributos mais finos, imitando a forma humana de classificação. Por exemplo, no envio de uma carta primeiro é identificado o país, no país, identifica-se o estado, no estado identifica-se a cidade e assim sucessivamente. Em suma para agrupar em n estágios, primeiro agrupa-se usando um atributo D_1 (mais abrangente). A seguir para cada grupo do estágio *i-1*, re-agrupa usando D_i para i variando de 2 até n. Como já observado, em cada estágio é aplicada uma métrica distinta para determinar a similaridade, mas em todos os estágios, é usado um método hierárquico aglomerativo de agrupamento. A medida de distância em cada estágio é resultante da separação da distância Euclidiana em componentes ortogonais. O critério para a união de dois grupos segue o método proposto e conhecido como *Ward's linkage method*. Este método une dois grupos que minimizam o aumento da soma do erro quadrático total (ESS). O ESS de um grupo é a soma dos quadrados dos desvios padrões. Ao aplicar o método para expressões gênicas, no primeiro estágio, os grupos estão com genes em níveis similares. No segundo estágio, os grupos estão com genes similares em nível e forma e assim sucessivamente.

Para imputação, a métrica utilizada para agrupar os registros é uma modificação da distância que comanda o agrupamento para aquele estágio. Isto significa calcular a raiz quadrada da distância Euclidiana para as amostras comuns aos dois genes. Os valores imputados são provenientes das informações dos genes pertencentes ao mesmo grupo em todos os estágios.

A abordagem multi-estágios foi testada removendo de 1% até 20% dos dados, num total de 1000 testes. Duas bases foram utilizadas: "*Yeast cell cycle data*" da UCI e uma base artificial. O desempenho da imputação foi medida pela raiz do erro quadrático médio (RMSE) e a solução proposta foi comparada com outros cinco métodos: média, K-nn, least squares imputation (LLSimpute) (KIM et al.,2005), Bayesian principal component (BPCA) (OBA et al., 2003) e Collateral Missing Value imputation (CMVE) (SEHGAL et al., 2005).

Para a base artificial, o valor de k para os métodos K-NN, LLSimpute e CMV foi estipulado em 10, pois cada grupo contém 10 genes e por ter sido reportado na literatura como um valor para os quais os métodos têm bom desempenho (SEHGAL et al 2005). Para a base *Yeast cell cycle data*, o k foi ajustado para 8 e o número de grupos para imputação em dois estágios em 30 (ficando em torno de 8 genes em cada grupo). Com base no RMSE , o método de 2 estágios só não teve melhor desempenho que o BPCA, embora o tempo de execução tenha sido muito inferior (1 iteração do BPCA gasta em torno de 40s e método de 2 estágios 1s) e a abordagem Bayesiana ser extremamente dependente da escolha da distribuição a priori. Alguns dos resultados apresentados no trabalho estão apresentados na tabela 3.1.

RMSE /	Base artificial			Base *Yeast cell cycle*		
métodos de imputação	Média	Menor	Máximo	Média	Menor	Máximo
Dois Estágios	0.1225	0.0915	0.1866	0.2743	0.2182	0.3374
Média	0.7053	0.6485	0.7530	0.3837	0.3230	0.4522
KNN	0.1325	0.097	0.1325	0.2860	0.2332	0.3500
LLSimpute	0.1849	0.1202	0.3005	0.4600	0.2787	8.2121
CMVE	-	0.7216	-	-	1.8897	-
BPCA	0.096	0.0738	0.1333	0.2493	0.2032	0.3058

Tabela 3.1 – Raiz do erro quadrático médio para 1000 iterações com 10% de valores ausentes para os métodos de imputação dois estágios, KNN , LLSimpute, CMVE (k=10) e BPCA

O foco dos testes da imputação em três estágios foi compará-la com a imputação em dois estágios. Para percentuais baixos de ausência (1 a 5%) obteve um desempenho melhor, mas a partir daí, devido às características da base, o comportamento do método assemelha-se à imputação por média e o desempenho cai bastante.

A divisão do espaço inicial em espaços ortogonais proposta por esta abordagem torna-se mais vantajosa quando o atributo dominante é consideravelmente maior que os demais. No entanto, é extremamente sensível à formação dos agrupamentos.

ZHU, ZHANG, FU (2004) apresentaram uma solução para imputação multivariada com variáveis discretas, cujo algoritmo foi denominado ROUSTIDA (*Rough Set Theory based Incomplete Data Analysis*), utilizando a teoria dos “Rough

Sets". O algoritmo proposto usa apenas informações que podem ser extraídas dos dados existentes na base de dados. A teoria matemática "*Rough Sets"* foi inventada por Pawlak, em 1982 e é um método matemático conceitual que lida com dados incertos ou ambíguos. Nesta teoria, um sistema de informação é uma tupla $I = <U , A>$. U é um conjunto finito, não vazio de n objetos, chamado domínio, ou seja $U = \{u_1,u_2, \ldots,u_n\}$. A é um conjunto finito, não vazio de m atributos, ou seja, $A = \{a_1,a_2, \ldots,a_m\}$. Para todo $a_i \in A$, $a_i : U \rightarrow Va_i$, onde Va_i é valor no domínio do atributo a_i. O domínio Va_i pode conter valores ausentes, representado por *. Neste caso, o sistema de informação é denominado sistema de informação incompleto. Quando A pode ser dividido em dois conjuntos sem interseção: o conjunto C de atributos condicionais e o conjunto D de atributos de decisão e C *e* D obedecem as seguintes condições: $A = C \cup D$ *e* $C \cap D = \varnothing$, este sistema de informação é chamado de sistema de decisão. Em geral, D contém apenas um atributo. Para um sistema de informação $I = <U,A>$, $B \subseteq A$, a matriz de discernibilidade $M(B)$ é uma matriz quadrada nxn, onde $M(B) = \{M(i,j)\}_{n*n,}$ $1 \leq i \leq$ n=|U|. A definição de uma unidade da matriz quadrada é a seguinte:

$$M(i,j)= \{a \in B: a(u_i) \neq a(u_j), u_i , u_j \in U \; i,j = 1,2, \ldots,n\}$$

ou seja, uma unidade da matriz quadrada é um conjunto de atributos, que especifica a diferença entre duas categorias no conjunto de atributos.

Uma categoria de objetos relevantes frequentemente tem algo em comum e os dados em um sistema de informação refletem as similaridades dos atributos mesmo havendo alguns dados ausentes. O objetivo básico é tornar os atributos dos objetos com valores ausentes tão consistentes quantos os de outros objetos similares no sistema de informação. A matriz de discernibilidade reflete as diferenças entre os atributos dos objetos e foi adaptada para um sistema de informação incompleto.

Considerando I o sistema de informação, M_ε a matriz de discernibilidade, MAS_i, o conjunto de atributos ausentes do objeto u_i , NS_i o conjunto de objetos sem diferenças em relação a u_i e MOS o conjunto de objetos com valores ausentes, o algoritmo iterativo ROUSTIDA pode ser descrito como a seqüência dos seguintes passos:

1) Calcula a matriz M_ε^0, MAS_i^0 para $i= 1,2, \ldots,n$, MOS^0 , $r=0$

2) *Passo 2*

Para todo i $\in$ MOS^r calcula NS_i^r

a. Gerar I^{r+1}

b. *Se* $I^{r+1} \neq I^r$

Calcula a matriz M_ε^{r+1}, MAS_i^{r+1} e MOS^{r+1} , $r= r+1$

Volta para o passo 2

3) Se ainda houver dados ausentes no sistema de informação I, utilizar outra abordagem para completá-los.

O cálculo de I^{r+1} segue os seguintes passos:

Para todo $i \notin MOS^r$

$a_k(u_i^{r+1}) = a_k(u_i^r)$ $k=1,2, \ldots,m;$

Para todo i $\in$ MOS^r

Para todo $k \in MAS_i^r$

Se $|NS_i^r| == 1$ *então*

seja j $\in$ NS_i^r

Se $a_k(u_j^r) = = *$ *então* $a_k(u_i^{r+1})=*$

senão $a_k(u_i^{r+1})= a_k(u_j^r)$

senão

Se $\exists$ um j_0 e $j_1 \in NS_i^r$ e $a_k(u_{j0}^r) \neq *$ e $a_k(u_{j1}^r) \neq *$ e $a_k(u_{j0}^r) \neq a_k(u_{j1}^r)$

então $a_k(u_i^{r+1})=*$

senão Se $\exists$ um $j_0 \in NS_i^r$ e $a_k(u_{j0}^r) \neq *$

então $a_k(u_i^{r+1})=a_k(u_{j0}^r)$

senão $a_k(u_i^{r+1})= *$

O exemplo abaixo ilustra a aplicação do algoritmo ROUSTIDA.

A tabela 3.2 (a) contém os dados de um sistema de informação incompleto, com 6 dados ausentes. O sistema transiente I^1 está representado em (b) e o sistema I^2 complementado pelo algoritmo ROUSTIDA é mostrado em (c). Os objetos u_1 e u_3 , u_2 e u_5 tem os mesmos valores em diferentes atributos e pertencem a mesma classe equivalente.

U	*a1*	*a2*	*a3*	*a4*
u1	4	*	1	2
u2	3	1	*	*
u3	*	1	1	*
u4	2	1	4	3
u5	*	1	3	4

(a)

U	*a1*	*a2*	*a3*	*a4*
u1	4	1	1	2
u2	3	1	*	*
u3	*	1	1	*
u4	2	1	4	3
u5	3	1	3	4

(b)

	a1	*a2*	*a3*	*a4*
u1	4	1	1	2
u2	3	1	3	4
u3	4	1	1	2
u4	2	1	4	3
u5	3	1	3	4

(c)

Tabela 3.2 – Base de exemplo com valores ausentes

O ROUSTIDA foi testado sobre duas bases: *Hayes data set* e *Íris data set* (sendo esta discretizada) com 2 a 5% de valores ausentes e caso o algoritmo não imputasse todos os valores foi utilizado o algoritmo CC para completá-los. Este algoritmo e os demais com os quais foi comparado, MMF e CMMF, CC pertencem ao software Rosetta, desenvolvido em conjunto pelas universidades de Varsóvia e Noruega e apresentou bom desempenho para ser adotado como um método de pré-processamento em mineração de dados. Foi aplicado na mineração de dados de um sistema gerencial de compras em shopping para encontrar regras de associação e, de acordo com os autores, mas não demonstrado, obteve bom desempenho.

KIM, KIM e YI (2004) desenvolveram uma modificação do método KNN também no contexto de Bioinformática, denominado SKNN. O método imputa os valores ausentes, aqui genes, utilizando, também valores previamente imputados. O valor ausente é preenchido pela média ponderada da coluna correspondente dos vizinhos mais próximos no conjunto completo. Após todos os valores de um gene terem sido imputados, o gene, agora preenchido com os valores estimados, é movido para o conjunto completo e usado para a imputação dos demais genes do conjunto incompleto. Neste processo, todos os valores ausentes em um gene são imputados simultaneamente a partir dos genes selecionados. Propõem, também, o algoritmo EM-SKNN onde o SKNN é aplicado iterativamente para melhorar a acurácia. O método foi testado em três bases, uma com dados mistos, outra com série temporais e outra com séries não temporais. Foi comparado aos métodos KNN, máxima verossimilhança (MLE) e imputação múltipla (MI). Utilizaram como métricas de comparação a taxa de erro RMS (erro quadrático médio), a preservação da correlação e o tempo computacional.

Os autores constatam que o valor do k é dependente do tipo de dados e da taxa de ausência. Como as menores taxas de erros para séries temporais e dados mistos foram obtidas para o k igual a 10, independente da taxa de ausência e para séries não temporais

não houve modificações significativas para o k entre 10 e 20. Dez foi o valor escolhido para o k nos testes publicados. A partir de taxas de 30% de ausência, a acurácia dos métodos propostos é bastante superior ao KNN, certamente porque utiliza os genes imputados para estimar os próximos.

A eficiência do MLE, considerando a taxa de erro, é muito menor que a do SKNN para todas as bases testadas. Em séries temporais com baixas taxas de ausência, o MI é melhor que o SKNN, mas um pouco pior para taxas mais altas. Para séries não temporais, o MI é menos eficiente que o SKNN. No entanto, para bases mistas, o melhor método foi o MI. Deste modo, os autores concluem que a eficiência do MI é similar a do SKNN para *microarrays*, embora seja mais dependente dos tipos de dados.

Foram também analisadas as correlações entre os dados nas bases originais e imputadas, concluindo que quanto maior for a correlação na base original, mais preservada mantém-se esta relação na base imputada pelo SKNN. Em comparação com os demais métodos, o SKNN foi quem melhor preservou a estrutura original tanto para séries temporais, quanto para dados não-temporais e mistos. Em contraste com o desempenho em relação ao RMSE, o MI comporta-se muito pior que o SKNN no que diz respeito à preservação da correlação original.

Em relação ao tempo computacional, o SKNN diminui o tempo de execução em relação ao k-NN, pois imputa todos os valores simultaneamente dado o subconjunto de vizinhos mais próximos. A aplicação do Expectation Maximization (EM) no SKNN (EM-SKNN) melhora a acurácia do método em detrimento do tempo computacional que aumenta proporcional ao número de iterações.

SEHGAL, GONDAL e DOOLEY, (2005) também no contexto da BioInformática, mais especificamente para o tratamento de *microarrays* incluindo seqüências de dados, propuseram um algoritmo conhecido como CMVE (“*Collateral missinig value estimation”* baseado em múltiplas matrizes de covariância para predizer os valores ausentes. As matrizes são computadas e otimizadas usando a regressão dos mínimos quadrados (LS) e métodos de programação linear, ou seja, combina matrizes múltiplas para um dado ausente em particular e otimiza seus parâmetros usando programação linear e a regressão LS.

O algoritmo proposto, a partir da matriz de expressões gênicas *Y(m,n)*, onde *m* representa o número de genes e *n* representa o número de amostras, segue os seguintes passos:

1. Localizar o valor ausente Y_{ij} no gene *I* e amostra *J*
2. Computar a covariância absoluta *CoV* da expressão gênica armazenada no vetor *v* do gene *I* de acordo com a seguinte equação:

$$CoV = \frac{1}{(n-1)} \sum_{i=1}^{n} (v_i - \bar{v})(\omega_i - \bar{\omega})$$

onde ω é o vetor utilizado na predição e *v* é o vetor com a expressão do gene.

Primeiro computa-se a covariância diagonal absoluta para o gene v, com cada gene sendo considerado iterativamente como ω, exceto o I

1. Ordenar os genes (linhas) de acordo com *CoV;*
2. Selecionar as K mais efetivas linhas Rk.

Após a ordenação dos genes, os primeiros K covariantes genes ($R_{k)}$ são selecionados, ou seja, aqueles cujos vetores de expressão são mais similares ao gene *I* de *Y* para todas as amostras exceto para amostra *J:*

1. Usar os valores de R_k para estimar o parâmetro Φ_1 de acordo com o método de regressão LS desenvolvido por Harvey e Arthur (2004):

$$\Phi_1 = \alpha + \beta X + \xi$$

2. Calcular Φ_2 e Φ_3 de acordo com o algoritmo NNLS (*non-negative least square*)
3. Calcular o valor ausente Y_{ij} usando como estimador $\chi = \rho.\Phi_1 + \Delta.\Phi_2 + \Lambda.\Phi_3$, $\rho = \Delta = \Lambda = 0.33$ para que os estimadores sejam equi-ponderados.
4. Procurar o próximo valor ausente Y_{ij} e repetir os passo 2 a 7 enquanto houver valores ausentes a estimar em *Y* .

Em suma, o valor final imputado é uma média ponderada dos três estimadores calculados para o valor ausente. Por sua vez, os estimadores são calculados utilizando os K genes com maior covariância em relação ao gene incompleto e a função objetivo

utilizada no algoritmo, baseada em técnicas de programação linear, minimiza o erro da predição (ξ).

Os autores testaram o algoritmo em três séries de uma base de dados sobre câncer no ovário com aproximadamente 6500 microarrays de genes e uma base com dados possuindo aproximadamente 6200 expressões gênicas sobre *"yeast sporulation"*. Compararam com os algoritmos *BPCA*, KNN e LSImpute com o k=10 para os dois últimos. Testaram e compararam os métodos com até 20% de valores ausentes, removendo-os aleatoriamente. O erro da estimativa foi calculado pela raiz normalizada do erro quadrático mínimo, proposta em (OUYANG et al.,2004) e definida por :

$$\Theta = \frac{\mathrm{RMS}(M - M_{\mathrm{est}})}{\mathrm{RMS}(M)}$$

onde *M* é a matriz de dados originais *e* M_{est} é a matriz estimada para cada um dos métodos. Nos testes realizados, o CMVE mostrou menor erro para quase a totalidade dos experimentos, perdendo apenas uma vez para o BPCA.

VERBOVEN, BRANDEN, GOOS (2007), também na área de Bioinformática, criaram o método chamado SEQImpute que complementa os valores ausentes sequencialmente (razão do nome) considerando a relação *acurácia x tempo* e introduz a idéia de minimizar a distância estatística em vez da distância Euclidiana, normalmente utilizada.

Considerando que:

- X é o conjunto de expressões gênicas com valores ausentes, representados por uma matriz de dimensão g × s, onde g>>s. As linhas representam os níveis de expressões e as colunas, as amostras/experimentos. X é composta necessariamente por um subconjunto de linhas com dados completos, isto é, sem valores ausentes, e, portanto, pode ser vista como: X = [Xc´ Xm´] onde Xc´ é uma matriz de dimensão c × s com os c casos completos e Xm´ é uma matriz de dimensão (g-c) × s com os casos incompletos.
- $\boldsymbol{x}_i$' = [$\boldsymbol{x}_m$', $\boldsymbol{x}_o$'] é uma linha da matriz , $\boldsymbol{x}_m$' refere-se à parte ausente do gene *i* e $\boldsymbol{x}_o$' à parte observada.

O método SEQImpute baseia-se nos conceitos de covariância e de determinante. A covariância mede o relacionamento linear entre duas variáveis da matriz. A diagonal

principal, portanto, mede a variância de cada variável e representa o quão espalhados estão os dados em relação à média. O determinante da matriz de covariância mede o grau de concentração dos dados, ou seja, quanto mais concentrados são os dados, menor é o determinante. Levando em consideração esta propriedades do determinante, o SEQimpute busca minimizar o determinante da covariância da matriz de dados X*=[Xc´, x*], onde x*=[x_m^{*}´,x_o^{*}´] é o gene incompleto. Portanto, minimizando o determinante da cov(X*) em relação a x_m^{*} pode-se obter um bom valor estimado para x_m^{*}. O algoritmo proposto segue os seguintes passos gerais:

1. Separar a matriz **X** em duas: $\hat{X}_c$ com os casos completos e X_m com os incompletos
2. Ordenar os genes incompletos $x_l^{*} = \left\lfloor x_m^{*}, x_o^{*} \right\rfloor$ em ordem crescente de quantidade de valores ausentes
3. Para cada l = 1,…, g-c (número de genes incompletos)
    i. X* = [$\hat{X}_c, x_l^{*}$] onde x_l^{*} é o l-ésimo gene incompleto x*=[x_m^{*}, x_o^{*}],
    ii. Calcular a covariância de X* :cov(X*)
    iii. Minimizar o determinante da cov(X*) em relação x_m^{*}
    iv. Incluir o valor estimado x_m^{*} em X_c
4. X_c final contém todos os dados completos (originais mais imputados)

É um método seqüencial, similar ao SKNN, com a vantagem da inexistência de parâmetros e necessidade de valores iniciais embora necessite de um subconjunto de dados completos iniciais. Usa imputação simples, mas pode ser adaptado para imputação múltipla.

Os autores compararam o método SEQImpute com os métodos SVDImpute,. BPCAImpute e o. KNNImpute em três bases, eliminando os dados originalmente ausentes nas bases *Lymphoma* resultando em 2317 genes de 65 pacientes com linfoma, *Mixed data set* resultando em 4380 genes de 24 amostras de fermentos vivos, *Golub* resultando em 7129 genes de 72 pacientes com leucemia. Nestas bases resultantes foram separados 5% dos dados para permanecer completos. Nos genes restantes, testaram os métodos com 1,3,5,10,15,20 e 30% de ausência. Deste modo obtiveram 21 casos testes e para cada caso simularam em torno de 100 vezes observando os seguintes critérios: tempo médio de computação, o erro relativo, o coeficiente de correlação entre as

variáveis originais e as estimadas e o valor absoluto do bias. O método ótimo foi considerado aquele onde o erro, o *bias* e o tempo computacional foram o mais próximo possível de 0 e onde todos os coeficientes de correlação são próximos de 1. Como conclusão dos testes, o SEQImpute é um dos mais rápidos, com o menor erro e a mais alta correlação. Como desvantagem, pode-se destacar o fato do método necessitar que uma parte da base esteja inicialmente completa e ser sensível a *outliers*, embora os demais métodos também o sejam.

O algoritmo BPCA proposto Oba et Al (2003) no contexto da BioInformática, estima os valores Y^{miss} de uma matriz Y usando os genes Y^{obs} que não possuem valores ausentes. O método atualiza os valores ausentes iterativamente. Alterna entre a atualização da distribuição a posteriori dos parâmetros do PCA e a distribuição a posteriori dos valores ausentes. Nenhum parâmetro precisa ser determinado porque o próprio algoritmo determina a dimensão do PCA. Apresenta bons resultados, mas é caro computacionalmente (VERBOVEN et al,2007).

O algoritmo conhecido como SRMI, ou *Sequential Regression Multivariate Imputation* (LEPKOWSKI, RAGHUNATHAN, SOLENBERGER, HOEWYK, 2001), constrói modelos preditivos para cada atributo de forma seqüencial. Inicia com o atributo com menos valores ausentes até o com mais valores ausentes. A cada iteração os valores imputados para um atributo participam da construção do modelo preditivo do próximo atributo. Utiliza o conceito de “restrições e limites” onde as informações são submetidas a um analista de dados para que este avalie e determine valores limítrofes e à regras auxiliares para a imputação, como por exemplo “ se idade<18 anos não tem habilitação de motorista”. Este conceito torna o método semi-automático e sujeito a conclusões nem sempre corretas dependendo do conhecimento do analista sobre o domínio dos dados que estão sendo imputados.

Um trabalho similar ao SRMI, denominado MICE , *Multivariate Imputation by Chained equations*, (OUDSHOOM, BUUREN, RIJCKEVORSEL,1999) constrói modelos preditivos como uma cadeia de equações que une a descoberta do valor de um atributo com a descoberta de valores de outros atributos com valores ausentes. Como qualquer método de imputação baseado em modelos, a construção de tais equações pode ser otimizada pela interferência de um analista, que aqui, deve ser capaz de especificar a distribuição condicional de um atributo em relação a outrso atributos preditivos

Troyanskaya et al (2001) aplicaram o método k-NN também no contexto da Bioinformática. Os microarrays são armazenados em uma matriz onde as linhas representam os genes e as colunas os experimentos. Neste método, os k genes mais similares, ou seja, os k vizinhos mais próximos pela distância Euclidiana, considerando todos os elementos da matriz completos na posição do gene a ser imputado, podem ser escolhidos para estimar o valor. O valor estimado é calculado como a média ponderada da coluna correspondente dos k-genes mais próximos. O peso do i-ésimo gene é calculado por $\sum_{i=1}^{k} \frac{1}{D_i}$, onde k é a quantidade de genes selecionados e Di é a distância entre o i-ésimo gene e o gene a ser imputado.

Tseng (2003) para aumentar a capacidade do SOM na mineração de dados, uma vez que este não lida com valores ausentes, aplicou a lógica nebulosa no tratamento dos dados incompletos. Neste tipo de rede, em vez de usar medidas estatísticas, utilizam-se conjuntos nebulosos como medida para o agrupamento. Como é necessário que os dados sejam discretos, deve-se pré-processar os valores contínuos dividindo-os em intervalos. A quantidade de intervalos influencia o tempo computacional, mas não necessariamente a descoberta do conhecimento. O método propõe transformar observações com valores ausentes em observações nebulosas para então treinar uma rede SOM gerando um mapa nebuloso. Após as observações com dados ausentes terem sido convertidas para observações nebulosas, cada uma delas torna-se completa, mas com sua “função de pertinência nebulosa”. Todos os dados (nebulosos ou não) são então utilizados para treinar a rede. A principal diferença desta rede para a tradicional é que nesta rede a incerteza de uma observação caso ela seja derivada de uma observação nebulosa fica registrada e o valor de pertinência nebuloso para cada nó de saída ativado no mapa nebuloso é acumulado.

FARHANGFAR, KJURGAN e PEDRYCZ (2007) propõem um *framework* para métodos de imputação almejando melhorar a acurácia da imputação em relação à utilização de um método simples de imputação, não modificando sua complexidade computacional e aplicável a diferentes métodos de imputação, incluindo técnicas estatísticas e baseadas em imputação múltipla (MI).

Para atingir estes critérios alguns dos valores são imputados diversas vezes, mas de forma distinta das técnicas que utilizam MI tradicional. As principais diferenças propostas são:

- Imputar somente um subconjunto de valores ausentes diversas vezes. A imputação é executada de forma iterativa. A cada iteração, os valores imputados que forem considerados de alta qualidade são aceitos e os que não o forem são imputados novamente (multi-imputados), sendo o número total de imputações demonstrado no artigo, inferior a 2*k*, onde k o número de valores ausentes. Nos métodos tradicionais MI, o número total de imputações não é menor que 3*k*, podendo ser superior a 10*k* (SHAFER,1999).
- Utilizar os valores aceitos como de alta qualidade para imputar os valores ausentes restantes. A reutilização dos valores pré-imputados também é uma abordagem utilizada nesta tese.

Nos testes realizados pelos autores, a utilização dos valores previamente imputados aumentou a acurácia, o mesmo acontecendo nos testes desta tese..

A arquitetura proposta consiste de 3 grandes módulos:

- Módulo 1 - Pré-imputação pela média - inicialmente os valores ausentes são pré-imputados, ou seja, temporariamente preenchidos com um valor calculado por um método rápido que calcula a média linear. A seguir, cada valor pré-imputado é imputado usando o algoritmo desejado e o valor calculado é filtrado usando um intervalo de confiança (Módulo 2).
- Módulo 2- Intervalos de Confiança - os intervalos de confiança são utilizados para selecionar os valores imputados mais prováveis, rejeitando possíveis valores fora dos limites (*outliers*). Uma vez que todos os valores foram imputados e filtrados, é atribuído a cada um, um quantificador que o qualifica, que pode ser expresso como uma probabilidade ou uma distância. Com este contexto finalizado, o Módulo 3 é executado.
- Módulo 3 – *Boosting* - Baseando-se nos qualificadores, o terceiro módulo aceita os valores imputados considerados de alta qualidade, enquanto os demais valores são rejeitados e o processo se repete com a nova base de dados parcialmente imputada. O processo termina quando todos os valores forem

computados ou após um máximo de 10 iterações, quando todos os dados são aceitos. O limite de 10 iterações foi determinado por experimentação sendo considerados o tempo computacional e a acurácia. *Boosting* (SCHAPIRE *et al*, 1998) é, originalmente, um método de aprendizado de máquina que valoriza a acurácia em problemas de classificação. Neste método, a partir de um conjunto dados, novos conjuntos são gerados sequencialmente, modificando pesos associados aos registros para que a próxima geração melhore a classificação de registros erroneamente classificados. Cada novo classificador é gerado a partir de um conjunto de dados onde as amostras classificadas erroneamente pelo classificador anterior têm maior peso e chance de seleção. Ele muda a distribuição do conjunto de treinamento em função do desempenho dos classificadores criados previamente e usa tal desempenho para definir um peso para o classificador no processo de votação. As classificações geradas pelos diferentes modelos são combinadas usando um tipo de votação (SCHAPIRE, 1999).

O *framework* proposto utiliza uma técnica similar ao *Boosting*. Um valor imputado é aceito ou rejeitado caso o peso a ele associado esteja acima ou abaixo de um limiar determinado. O peso deve refletir a qualidade da imputação. Tanto o peso como o limiar provém dos dados e são dependentes do método de imputação utilizado.

O *framework* foi testado com 8 métodos de imputação em 16 bases de dados, sobre dados discretos (numéricos e categóricos), provenientes dos repositórios University of Califórnia at Irvine ML (UCI-ML) e Knowledge Discovery in Databases (UCI KDD) e em uma base de dados artificial. Os valores ausentes foram introduzidos artificial e uniformemente nos atributos, exceto nas classes, em seis níveis percentuais: 5%, 10%, 20%, 30%, 40% e 50%, para estudar, também, o impacto da quantidade de dados ausentes na qualidade da imputação. O *framework* foi analisado sob 3 focos distintos: quanto ao desempenho dos módulos observando o efeito de cada um dos módulos na melhora da acurácia da imputação, quanto à contribuição do *framework* no desempenho dos métodos, comparando a qualidade da imputação entre os métodos sem o uso do *framework* e com o uso do *framework* e quanto à complexidade, cujo objetivo é mostrar que a complexidade computacional da aplicação do *framework* é linear e, portanto, não piora a complexidade inerente ao método.

A acurácia é definida pela divisão do total de imputações corretas pelo total de valores ausentes.

Concluíram que aplicação de cada um dos módulos do *framework* e ele como um todo, sempre resulta em alguma melhora na acurácia da imputação, sendo que o nível desta melhora é dependente do próprio método, ou seja, é maior para métodos com pior desempenho e menor para métodos de melhor desempenho. Esta mesma conclusão foi alcançada quanto ao desempenho do método de imputação com ou sem o uso do *framework.* A complexidade do método, demonstrada no artigo, também foi confirmada experimentalmente.

TSENG, WANG e LEE (2003) propõem uma técnica denominada *Regression Clustering* (RC) onde combinam imputação e agrupamento para tratar o problema de valores ausentes numéricos. Em primeiro lugar, os dados ausentes do conjunto de dados são inicializados com valores derivados de todos os registros da tabela. Esta imputação global é necessária para que os registros possam ser utilizados pelo método de agrupamento, no entanto, podem tornar os grupos tendenciosos. A seguir, com a base "completa", um algoritmo de agrupamento partitivo, denominado CAST, cria grupos disjuntos. Para cada grupo, então, o algoritmo de imputação é aplicado gerando os valores ausentes em função das informações apenas dos registros do grupo. Em linhas gerais, o funcionamento é o seguinte:

1) Dividir o conjunto de registros D em dois subconjuntos: D_C (dados completos) e D_M (dados ausentes). Completar D_M com um algoritmo de imputação baseado em D_C, obtendo o conjunto D'.

2) Agrupar os elementos de D' em k grupos $C_1, C_2, \ldots, C_K$ $(\sum |C_i| = |D'|)$

3) Para cada grupo C_i, a imputação é aplicada a todos os registros R_j, tais que $R_j \in D_M \cap C_i$. A base usada para a imputação é o conjunto $\{R_C \mid R_C \in D_M \cap C_i\}$

O método RC foi comparado com os algoritmos EM, média, regressão (que os autores não especificam qual), e com o número de grupos do algoritmo *K*-Means variando entre 3 e 48, com saltos de 3 unidades.

O erro relativo absoluto (RAD *Relative Absolute Deviation*) foi a medida de erro usada para mensurar os valores gerados pelos algoritmos, definido por:

$$RAD = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|X_O^i - X_R^i\right|}{X_O^i}$$

onde X_O^i é o valor original do atributo *X* da tupla *i*, X_R^i é o valor imputado do atributo *X* nesta tupla *i*, e *n* é o total de tuplas com valores ausentes no atributo *X*.

Os experimentos foram executados sobre duas classes de bases de dados: uma base gerada aleatoriamente e outra com 5000 registros e 10 atributos, divididos em 4 grupos, com até 20% de valores ausentes.. Nesta base, o método proposto pelos autores sobressaiu-se sobre os demais sendo que o algoritmo K-Means obteve o pior desempenho, principalmente para as taxas de ausência mais elevadas. O método proposto também apresentou um desempenho pior em níveis mais altos de valores ausentes, embora os autores justifiquem a queda de desempenho não pelo percentual de sujeira mas pela distribuição dos dados. Na base gerada aleatoriamente, em geral todos os métodos apresentaram um desempenho inferior à base agrupada. Os métodos de agrupamento (K-Means e o RegressionClustering) obtiveram melhor desempenho.

RALLO, FERR e GIRALT (2004) desenvolveram uma estratégia de imputação múltipla utilizando conjuntos de mapas auto-organizáveis (SOM) aplicando-a em duas bases industriais na área química.

A idéia principal baseia-se no conceito de agregação de modelos. A agregação tenta melhorar a qualidade dos valores imputados gerando múltiplas versões do sistema de imputação $\varphi(x_i)$, combinando-os de algum modo, em geral pela média definida por:

$$\varphi_{aggregated}(X) = \frac{1}{N}\sum_{i=1}^{N}\varphi_i(x)$$

onde $\varphi_{aggregated}(x)$ é o sistema de imputação agregado e *N* é a cardinalidade do conjunto.

A diversidade dos modelos que devem ser posteriormente agregados foi introduzida de duas formas. A primeira pela troca do tamanho dos mapas em cada modelo de imputação simples. Deste modo, um conjunto pode conter modelos subajustados (com maior capacidade de generalização) e modelos superajustados (com grande acurácia e adaptado a certas regiões dos dados de treinamento). A segunda forma

de diversidade é pela manipulação dos conjuntos de treinamento usando técnicas de "B*agging*" (BREIMAN, 1996). O método *Bagging* gera diferentes classificadores a partir de diferentes amostras geradas pela técnica *boostrap*. Esta técnica consiste em dividir um conjunto de dados com *N* elementos em um conjunto de treinamento, com *N* seleções uniformes com reposição, e um conjunto de teste, com os objetos não incluídos no conjunto de treinamento. O algoritmo muda estocasticamente os conjuntos de treinamento e usa pesos iguais para os classificadores no processo de votação.

O procedimento inicia considerando o conjunto de treinamento TR formado por N padrões, cada um com uma probabilidade 1/N associada. Um novo conjunto de treinamento TR_{bag} é criado por amostragem usando estas probabilidades. Deste modo, alguns casos em TR podem nunca aparecer em TR_{bag} enquanto outros podem aparecer múltiplas vezes. O novo conjunto de treinamento é usado para treinar o modelo de imputação. Este processo é repetido diversas vezes e os resultados de cada modelo individual são combinados. O modelo de imputação simples usando SOM baseia-se na estimação dos valores ausentes usando os protótipos nos grupos correspondentes na SOM. Neste trabalho foram utilizadas duas abordagens para selecionar os protótipos e combiná-los: a) substituição direta pelo componente correspondente no protótipo do *bmu* (*"best match unit") e* b) substituindo pelo valor médio obtido utilizando o *bmu* e sua vizinhança de raio 3. O tamanho da vizinhança foi escolhido por experimentação. Além disso, executou-se *n-fold cross validation* com n igual a 10 no treinamento do modelo para obter maior acurácia na representação dos mapas.

Os autores testaram o método proposto em dados provenientes de dois processos industriais e o comparam com a média e uma técnica de imputação simples.

A primeira base contém dados provenientes da polimerização de etileno, com 148 variáveis (pressões, vazão, temperaturas de aquecimento / resfriamento, etc.) extremamente relacionadas ao processo produtivo e amostradas em intervalos de 10 minutos. A base utilizada para o treinamento da rede SOM continha 5548 padrões de entrada completos e incompletos. Para os casos analisados, o efeito de falhas aleatórias no sensor de temperatura, falhas no medidor de fluxo, ou falha em qualquer um dos 148 sensores, as principais conclusões dos autores foram: (i) independente do método usado para construir o sistema de imputação simples, o erro médio absoluto da agregação dos mapas é sempre menor que os dos modelos individualmente. (ii) o uso de conjunto de

mapas de tamanhos diferentes na imputação múltipla leva a melhores resultados que os que utilizam técnicas baseadas em *bagging*. (iii) resultados obtidos usando apenas os componentes do protótipo *bmu* em imputação baseada em SOM obtiveram melhores resultados em relação aos que utilizam a média dos vizinhos. (iv) todo sistema de imputação mantém um comportamento estável com o aumento de dados ausentes, independente da técnica de imputação utilizada. Os sistemas de imputação baseados em SOM são muito robustos em relação à quantidade de dados ausentes. Pelos testes realizados com percentual de ausência variando de 10% a 70%, o desempenho do sistema em termos de erro médio absoluto não se degradou de modo significativo. (v) todos os métodos de imputação por SOM tendem a superestimar a média do valor imputado, enquanto aqueles baseados somente pela substituição com a média são mais precisos, mas não reproduzem bem a variância dos dados. (vi) dependendo das propriedades estatísticas dos dados modelados, sistemas de imputação baseados em protótipos têm desempenho baixo.

O segundo caso estudado utiliza uma base de dados da "UCI Machine Learning Repository" (MERZ 1998) que contém registros da operação de uma estação de tratamento de água - "Waste Water Treatment Plant" (WWTP). Esta base contém 521 registros, com uma grande quantidade de dados ausentes. Cada registro possui 38 atributos sendo que 29 correspondem a medidas tomadas na estação e 9 são medidas de desempenho. Pela alta taxa de valores ausentes, não é possível treinar um sistema sensor convencional e considerando os resultados do experimento anterior, os métodos de imputação utilizaram apenas o *bmu*. Após a reconstrução da base usando cada um dos métodos de imputação em análise uma rede neuronal *backpropagation* para cada um dos três sensores alvos foi treinada. Neste contexto, as redes treinadas com mapas baseados em *bagging* tiveram melhor desempenho.

CAPÍTULO 4

IMPUTAÇÃO EM CASCATA

4.1.Introdução

A substituição de valores ausentes em um único atributo em bases de dados é um problema de difícil solução. A medida que as bases crescem a tendência destas apresentarem uma maior quantidade de valores incompletos também cresce. Nos problemas reais qualquer atributo da base pode estar ausente e, numa mesma tupla diversos atributos podem estar incompletos. Em frente a este caos, é natural perguntar-se: “por onde começar?” Na imputação seqüencial, um paradigma de solução para esta questão, a ordem de imputação, quais tuplas devem ser consideradas e se os valores previamente estimados devem ou não ser utilizados para imputações futuras são questões relevantes (GELMAN e HILL, 2006) e pouco estudadas, tornando-se um dos alvos de estudo desta tese. Conforme afirmação de JUNNINEN et al (2004), métodos de imputação não podem ser considerados um tipo de alquimia estatística onde a informação é gerada a partir do nada. Dados ausentes são sempre perdidos, na sua totalidade e para sempre, mas um esquema adequado de imputação pode ajudar a remediar a situação.

No capítulo 3 foram mostradas algumas soluções existentes para o processo de imputação de valores ausentes em bases de dados. No entanto, é nítido que os métodos se valem de características do domínio de aplicação para melhorar o desempenho. Em sua maioria, a solução proposta é comparada com outros métodos existentes de acordo com alguma métrica. Como esperado, não é possível identificar um método de imputação universal, nem mesmo uma orientação de que algoritmos são mais indicados para uma dada configuração de um conjunto de dados que apresente valores ausentes. A maioria das pesquisas aponta o mecanismo de ausência aleatório (MAR) como o mais apropriado para a complementação, e indicam o mecanismo de ausência não aleatório (NMAR) como os de tratamento mais difícil. Com intuito de preservar a variância inerente à base de dados, há uma tendência em aplicar imputação múltipla proposta por RUBIN (1988), onde métodos de imputação simples são utilizados para gerar várias

imputações, que posteriormente são combinados e analisados. Porém, como esta combinação e análise são realizadas não é claramente definida. As abordagens híbridas vêm sendo mais utilizadas e tendem a combinar as melhores características dos métodos de imputação, pagando o respectivo preço computacional.

Diversos problemas adicionais podem ser observados quando a ausência não se restringe a apenas um atributo da base. Muitos trabalhos da área de Bioinformática abordam a questão da ausência em diversos atributos, porém poucos explicam como tratam este problema, ou seja, por tupla ou por atributo. Claramente o desempenho dos métodos de imputação não depende exclusivamente da quantidade de valores ausentes, mas também do padrão, do mecanismo e da distribuição desta ausência. O aumento da complexidade destes padrões de ausência dificulta mais ainda o processo de imputação bem como a ordem de processamento de uma série de atributos.

Em frente a estes questionamentos inerentes à complementação, propõe-se uma abordagem para lidar com a complementação de múltiplos atributos, denominado Imputação em Cascata.

A Imputação em Cascata é uma proposta para imputação multivariada que trabalha com tabelas compostas por atributos numéricos e em cujas tuplas pode haver vários atributos com valores ausentes simultaneamente. Levando em consideração o bom desempenho da tarefa de agrupamento precedendo a imputação, experimentada por SOARES (2007) no contexto de imputação simples de um atributo, a abordagem aqui proposta é híbrida, onde a tarefa de agrupamento precede o processo de imputação. Os casos incompletos são distribuídos em grupos considerando como critério de alocação o conceito de morfologia da ausência neles existentes. A morfologia de ausência, conceito aqui proposto, considera a distribuição espacial da ausência, a forma como os valores presentes e ausentes nos atributos estão distribuídos nos casos da base. Portanto, analisa a relação posicional dos atributos preenchidos e não preenchidos para agrupar os casos. Os grupos previamente complementados são reutilizados para a imputação dos grupos posteriores em um efeito cascata. Com esta abordagem do tipo divisão e conquista, acredita-se simplificar o processo de imputação e melhorar a qualidade do dado imputado A figura 4.1 mostra a idéia da abordagem proposta:

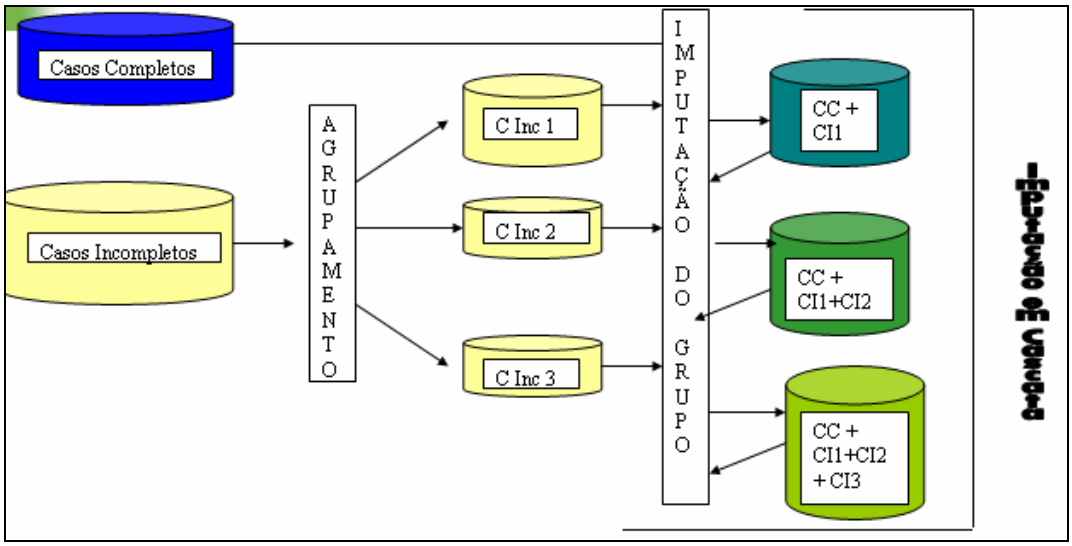


Figura 4.1 - A Imputação em Cascata

O método de agrupamento ou imputação escolhido não é pré-determinado. O método de imputação de cada grupo pode ser distinto, ou seja, grupos com padrões menos complexos podem utilizar métodos de imputação mais simples enquanto grupos com padrões mais complexos podem ser imputados por métodos mais caros e sofisticados. Uma vez que um grupo foi completado, ele é absorvido pela base para auxiliar na complementação dos demais grupos. Todavia, por questões operacionais, entre elas podemos citar a explosão combinatória de métodos e parâmetros, o tempo computacional e critérios para comparação de resultados, este estudo limitou-se em um método de agrupamento e de imputação. Dado o viés em Inteligência Computacional, para a tarefa de agrupamento optou-se por uma rede SOM (s*elf-organizing map)* e para a restauração dos casos, a imputação seqüencial com realimentação de valores. A regressão de cada célula ausente é resultado da média dos valores dos atributos dos casos selecionados pelo algoritmo dos K-vizinhos mais próximos (kNN).

Em função destas escolhas outras variáveis podem ser analisadas e não apenas o desempenho do método. Portanto, também é objeto de estudo avaliar topologias da rede, número de vizinhos, critérios de distância, critérios de ordenação entre grupos e intragrupo e o impacto da imputação na relação original entre os atributos da base. Os experimentos realizados são frutos da combinação das diversas alternativas possíveis dos inúmeros parâmetros ajustáveis dos métodos escolhidos. Para viabilizá-los, foi construída uma plataforma, *workflow-like*, em Java, que gera todas as combinações possíveis destas alternativas e as dispara automaticamente.

Assim, na seção 4.2 descreve-se a abordagem proposta. Na seção 4.3, é descrita a plataforma implementada que utiliza a filosofia de workflow. Este plataforma abriga todas as tarefas necessárias ao processo descrito, bem como sua seqüência de execução e variação de parâmetros. O processo de validação dos resultados é feito de duas formas: com a aferição do erro das imputações e do acréscimo da covariância. Esta última avaliação permite verificar se os valores imputados mantêm as relações originais intrínsecas às colunas da tabela.

4.2.Descrição da abordagem proposta

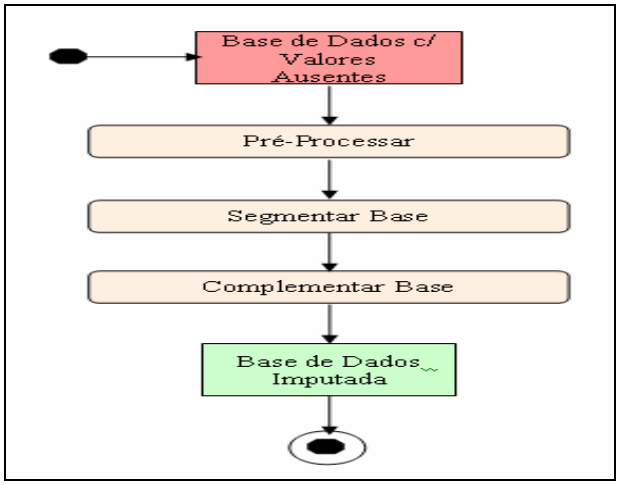


Figura 4.2 - Arquitetura Geral da Imputação em Cascata

A arquitetura geral da imputação em cascata é visualizada na figura 4.2. Ela consiste de três grandes módulos funcionais. 1) pré-processamento; 2) segmentação da base; 3) complementação dos segmentos. A primeira tarefa, o pré-processamento, normaliza os valores da base e extrai da base original os casos com valores ausentes, produzindo duas bases: a base de dados com casos incompletos e a base de dados com casos completos. Portanto, o conjunto de casos originalmente completos são inicialmente separados e compõe a base original inicial para o terceiro módulo, a imputação dos segmentos.

O segundo módulo, denominado segmentação da base incompleta, tem como objetivo dividir a base de entrada, que contém casos incompletos em m>= 1 sub-bases ou grupos de acordo com a morfologia da ausência. Para que o processo de divisão não seja influenciado por nenhum valor existente na base, os casos são binarizados,

representando a ausência de valor em um atributo pelo número zero e a presença de valor pelo número um.

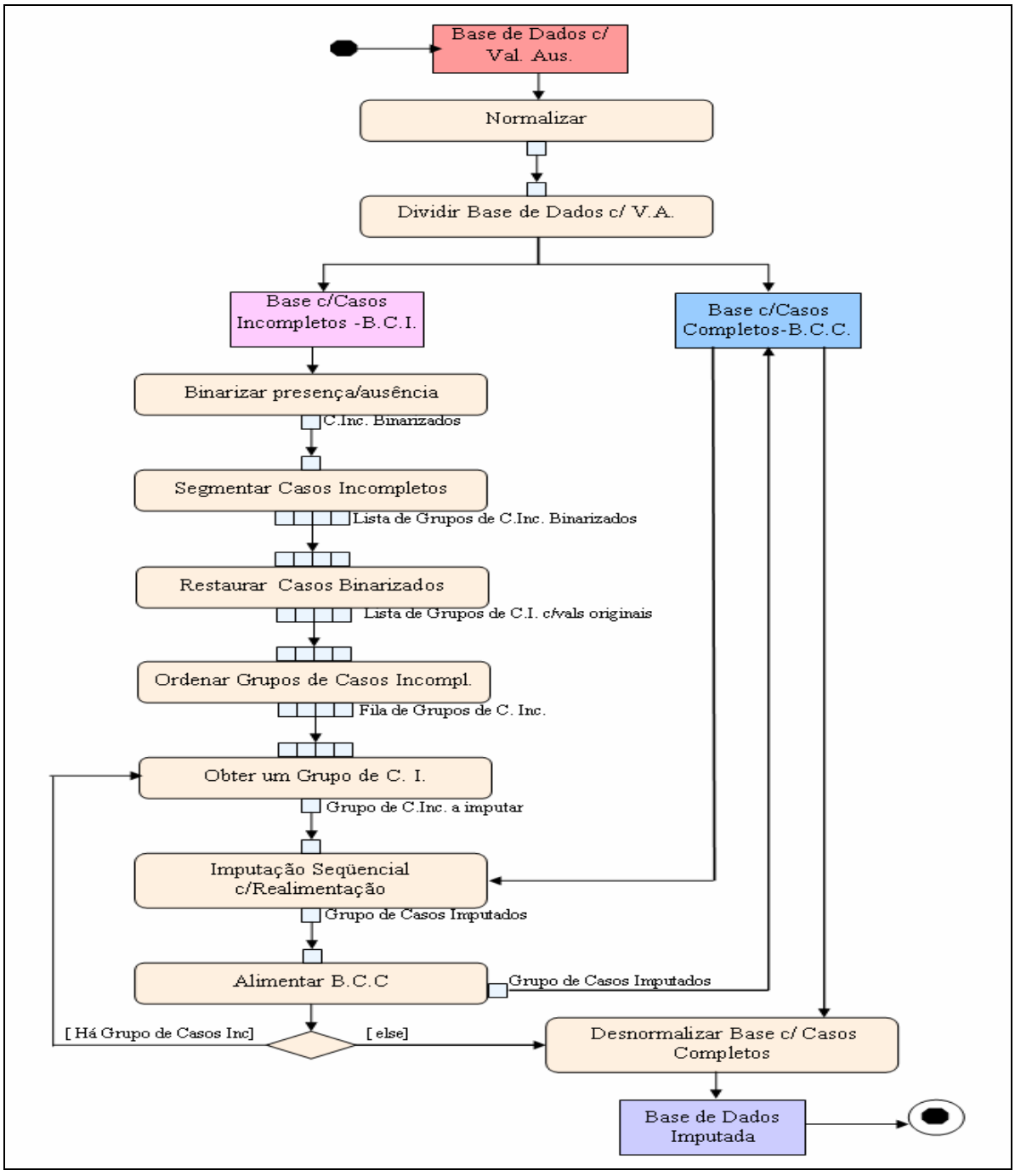


Figura 4.3 – Visão mais detalhada da Imputação em Cascata

A terceira tarefa, complementação dos segmentos, é responsável pelo processo de complementação dos valores ausentes de cada sub-base, gerando, ao final, a base restaurada. Para tanto, são selecionadas iterativamente e de acordo com o critério de

ordenação estabelecido. A cada ciclo a sub-base restaurada é incorporada à base inicial, tornando-a uma base parcialmente imputada, em um efeito cascata. Na imputação de cada segmento, o método básico de imputação pode ser distinto e levar em consideração tanto a complexidade morfológica da ausência presente no segmento atual como a ordem de imputação dos atributos, a reutilização dos valores previamente imputados ou qualquer outra característica relevante presente nos dados.

A figura 4.3 acima detalha a Imputação em Cascata.

4.2.1 Pré-Processamento

O pré-processamento pode ser sumarizado conforme a figura 4.4 abaixo:

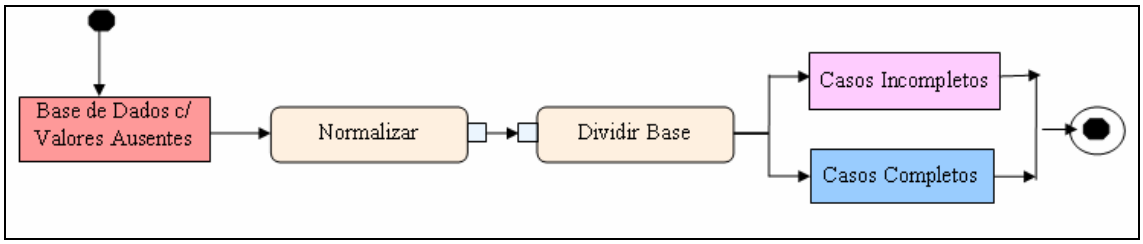


Figura 4.4 - Etapa de pré-processamento

Mesmo focando apenas em atributos numéricos contínuos, o domínio dos valores de cada atributo é extremamente diversificado. Deste modo, alguns atributos podem sobrepujar de tal forma os demais que invalide o processo como um todo. Portanto, na etapa de pré-processamento para ajustar as escalas dos valores dos atributos, estes devem ser inicialmente normalizados. O método de normalização escolhido é irrelevante e pode ser determinado pelo analista.

Para exemplificar, considere a seguinte tabela de dados:

ID	Atr1	Atr2	Atr3	Atr4
1	12	89	2333	?
2	?	23.5	?	?
3	11	?	2	?
4	10	876	12345	2342
5	10	980	97689	678906
6	?	?	?	7899
7	10	?	3	2342
8	?	43.2	?	2342

Com a normalização, segundo o método Z-score, os valores desta tabela ficam como abaixo:

ID	Atr1	Atr2	Atr3	Atr4
1	1.565248	-0.65028	-0.47555	?
2	?	-0.78622	?	?
3	0.447214	?	-0.53058	?
4	-0.67082	0.982997	-0.23916	-0.4518
5	-0.67082	1.198831	1.77585	1.788798
6	?	?	?	-0.4334
7	-0.67082	?	-0.53056	-0.4518
8	?	-0.74533	?	-0.4518

Após a normalização, a base de dados original é separada em duas bases:

1) base com dados completos: nesta base ficam armazenadas apenas as tuplas que estão com todos os seus atributos preenchidos;

2) base com dados incompletos: nesta base ficam armazenadas todas as tuplas que tenham valor ausente em qualquer um de seus atributos.

Estas bases são as saídas deste módulo.

Para a tabela acima, após a divisão, duas novas tabelas são geradas:

Base Completa

ID	Atr1	Atr2	Atr3	Atr4
4	-0.67082	0.982997	-0.23916	-0.4518
5	-0.67082	1.198831	1.77585	1.788798

Base Incompleta

ID	Atr1	Atr2	Atr3	Atr4
1	1.565248	-0.65028	-0.47555	?
2	?	-0.78622	?	?
3	0.447214	?	-0.53058	?
6	?	?	?	-0.4334
7	-0.67082	?	-0.53056	-0.4518
8	?	-0.74533	?	-0.4518

4.2.1.1 Considerações sobre a normalização na aplicação implementada

Como a Imputação em Cascata não depende de como normalização é realizada, apenas exige que os dados estejam normalizados, alguns métodos existentes na literatura (vide apêndice I) foram anexados na plataforma desenvolvida e podem ser escolhidos pelo analista do processo. Testes iniciais foram realizados utilizando dois

métodos de normalização: score-Z e Min-Max apresentando comportamento semelhante. Deste modo, o método Z-score foi escolhido nos experimentos devido à sua popularidade.

4.2.2 Segmentação da base incompleta

Aqui, a base incompleta é particionada em *m* sub-bases utilizando como critério de divisão a morfologia do padrão de ausência, conforme figura 4.5 abaixo:

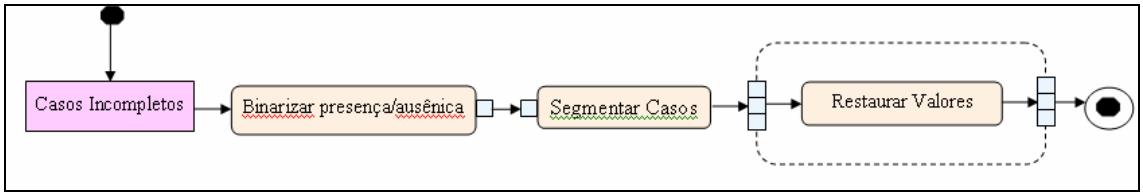


Figura 4.5 Segmentação da base Incompleta

Como em qualquer tarefa de agrupamento, deseja-se que os grupos sejam bem formados, preferencialmente eqüitativos, e que possa haver um modo de caracterizá-los, ou seja, se uma tarefa de sumarização sucedesse o agrupamento obteria-se uma descrição dos mesmos. Deste modo, o critério utilizado para a formação dos grupos é primordial e preferencialmente independente dos valores presentes na base de dados.

Seguindo este pensamento e observando a tabela exemplo, verifica-se que a tupla 1 e 7 possuem apenas um atributo ausente, as tuplas 3 e 8 possuem dois atributos ausentes enquanto as tuplas 2 e 6 possuem três atributos incompletos. Desta observação, em uma primeira análise, a quantidade de atributos ausentes poderia ser escolhida como critério de divisão. Porém, num questionamento mais amplo surge naturalmente a hipótese de considerar também quem são estes atributos ausentes e como eles estão distribuídos, pois podem existir diversas tuplas com a mesma conformação de ausência (o que justificaria a existência de um grupo) e o mesmo atributo pode estar ausente em conjunto com outros (mas não necessariamente os mesmos). Ao considerar tais questões surgem novos critérios, como por exemplo, a divisão por atributo ausente ou combinações de atributos ausentes. Unindo a estas questões o fato do agrupamento dos registros ter como objetivo a complementação dos mesmos e que para imputar os valores ausentes deseja-se buscar a resposta nas outras tuplas existentes na base (por exemplo utilizando o paradigma *hot-deck* para a escolha de possíveis doadores dos

valores estimados) torna-se bastante plausível usar como critério de divisão a morfologia da ausência. A morfologia da ausência preocupa-se com a relação espacial da presença e ausência de valores nos atributos, ou seja, em como a falta de valores está distribuída nos casos, a forma desta distribuição. Portanto, para a divisão, os registros iniciais são vistos apenas como apresentando (ou não) valor no atributo, conforme ilustrado no item(b) da figura 4.6 para a tabela do exemplo anterior.

Busca-se aqui, uma separação menos elaborada, pode-se até dizer que são desejados “montes” e não grupos no sentido estrito.

Em resumo para a segmentação não interessa os valores com os quais os atributos completos estão instanciados, apenas a conformação espacial dos mesmos, a morfologia. Portanto, para manter a integridade desta etapa, seja qual for o método de agrupamento escolhido, os casos incompletos passam por um processo de binarização, onde a presença de valor em um atributo é representado pelo número um e a ausência de valor é representada pelo número zero (figura 4.6.item(c)). Deste modo, garante-se que o algoritmo de agrupamento escolhido não seja influenciado pelos valores dos atributos, apenas pela forma. Após a segmentação, cada uma das sub-bases é restaurada, ou seja, os valores normalizados dos atributos completos são restabelecidos, sendo então encaminhados para o próximo módulo.

Base Incompleta

Atr1	Atr2	Atr3	Atr4
1.565248	-0.65028	-0.47555	?
?	-0.78622	?	?
0.447214	?	-0.53058	?
?	?	?	-0.4334
-0.67082	?	-0.53056	-0.4518
?	-0.74533	?	-0.4518

(a)

Morfologia da Ausência

Atr1	Atr2	Atr3	Atr4
√	√	√	X
X	√	X	X
√	X	√	X
X	X	X	√
√	X	√	√
X	√	X	√

(b)

Base Binarizada

Atr1	Atr2	Atr3	Atr4
1	1	1	0
0	1	0	0
1	0	1	0
0	0	0	1
1	0	1	1
0	1	0	1

(c)

Figura 4.6 – Representação da morfologia da Ausência e Binarização

Como já salientado, para a Imputação em Cascata (e na ferramenta desenvolvida) o método de agrupamento não é relevante desde que as relações inerentes nos padrões de entrada estejam refletidas nos grupos e que os grupos formados preservem a

formação topológica de ausência. Ele pode ser escolhido pelo analista do processo. Porém, para uma maior automatização desta estapa é interessante que a quantidade de grupos possa ser determinada automaticamente.

4.2.2.1 Considerações sobre a segmentação da base na aplicação implementada

Os métodos de agrupamento mais usuais (entre eles o *k-means)* em geral esperam que o número de grupos seja pré-determinado. Embora 2^k-1 seja o número máximo de grupos (visto ser a maior quantidade de combinações possíveis) é desconhecido a priori se tal particionamento refletirá as relações inerentes nos padrões de entrada.

As redes SOM são muito aplicadas para agrupamento apresentando bom desempenho em diversas áreas do conhecimento. Exploram bem os padrões de entrada e os projetam em um mapa de menor dimensão (no caso desta tese, bidimensional), que pode ser efetivamente utilizado para visualizar e explorar as propriedades dos dados (VESANTO e ALHONIEMI, 2000). Duas grandes vantagens desta rede como técnica de agrupamento, (tornando-a uma ótima candidata à segmentação da base em sub-bases) são não precisar da pré-determinação do número de grupos e após seu treinamento ser possível dividi-la de forma automática. Portanto, o uso da rede SOM para a fase de segmentação (vide apêndice IV) é uma boa escolha. Para a segmentação da rede treinada há diversos algoritmos possíveis. O algoritmo Costa-Netto (vide apêndice IV) foi escolhido neste estudo pelo seu bom desempenho e simplicidade.

O passo de agrupamento implementado pode então ser resumido por:

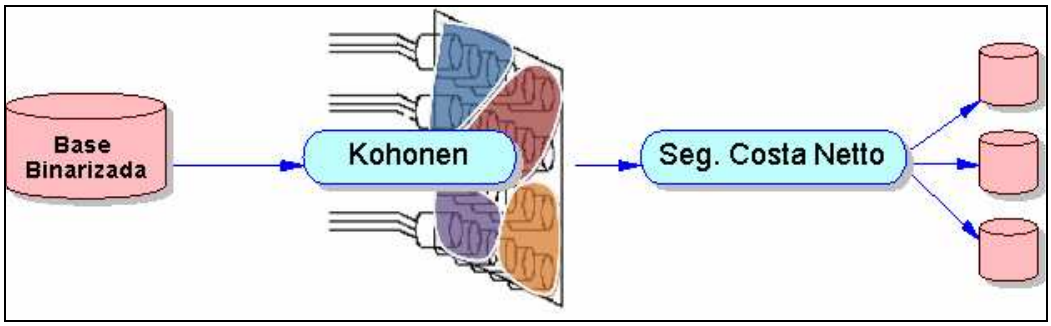


Figura 4.7 Etapa de Agrupamento implementada

A rede SOM entre outros parâmetros, é sensível ao número de nós do mapa (topologia). O excesso de nós implica na perda da capacidade de compactação, enquanto a falta de nós não permite a distribuição dos padrões. Logo, a topologia da rede deve ser

suficientemente grande para que a rede consiga espalhar os padrões de entrada no mapa, mas sem os "decorar" (*overfitting*). Este ajuste é um procedimento ad-hoc e geralmente resultante de tentativas e erros. Como a Imputação em Cascata é uma proposta para o problema de complementação de dados ausentes no processo de KDD, não é muito apropriado ter-se uma sub-tarefa dependente de ajustes por tentativa e erro. Conforme já comentado, não almeja-se o agrupamento ótimo, mas suficiente para identificação de macro-grupos. Deste modo, com o objetivo de manter-se independente do domínio, propõe-se e avalia-se nos experimentos cinco topologias, a saber:

1. Número de atributos**:** nesta topologia o mapa é composto por 2^k nós, onde k representa o número de atributos da base. Deste modo há um protótipo no mapa para cada combinação possível de ausência.

2. Número de tuplas: nesta topologia o mapa é composto por tantos nós quantos os registros da base original (tuplas completas e incompletas).

3. Número de células ausentes**:** esta topologia considera a quantidade de células ausentes da base, independente do fato de ser o mesmo atributo, ou estarem na mesma tupla ou em tuplas distintas. O conceito que norteia esta escolha pode ser ilustrado pela diagonal principal de uma matriz. Ou seja, há um nó no mapa para cada célula ausente e no pior caso, este ocorre uma única vez e em um único registro.

4. Número de tuplas com valores ausentes**:** esta topologia considera a quantidade de tuplas com valores ausentes independente da quantidade de atributos ausentes que cada tupla possui. Embora não muito desejado, cada tupla pode tornar-se um grupo. A lógica desta topologia, é que se todos os registros possuírem algum valor ausente e com morfologia distinta, há no máximo *número de tuplas* grupos distintos.

5. Média de atributos**:** nesta topologia o número de nós determinado pela divisão da quantidade de atributos ausentes pela quantidade de atributos da base considerando uma distribuição eqüitativa dos atributos ausentes nas tuplas. Busca-se aqui, uma relação entre as duas topologias anteriores.

4.2.3 Complementação dos segmentos

Esta etapa incorpora o processo de complementação dos valores ausentes, propriamente dito. Este processo é dividido em dois grandes ciclos. Um ciclo externo para gerenciar a fila de segmentos e o segundo, interno, que controla a imputação dos atributos dentro do segmento.

4.2.3.1 Gerenciador de segmentos

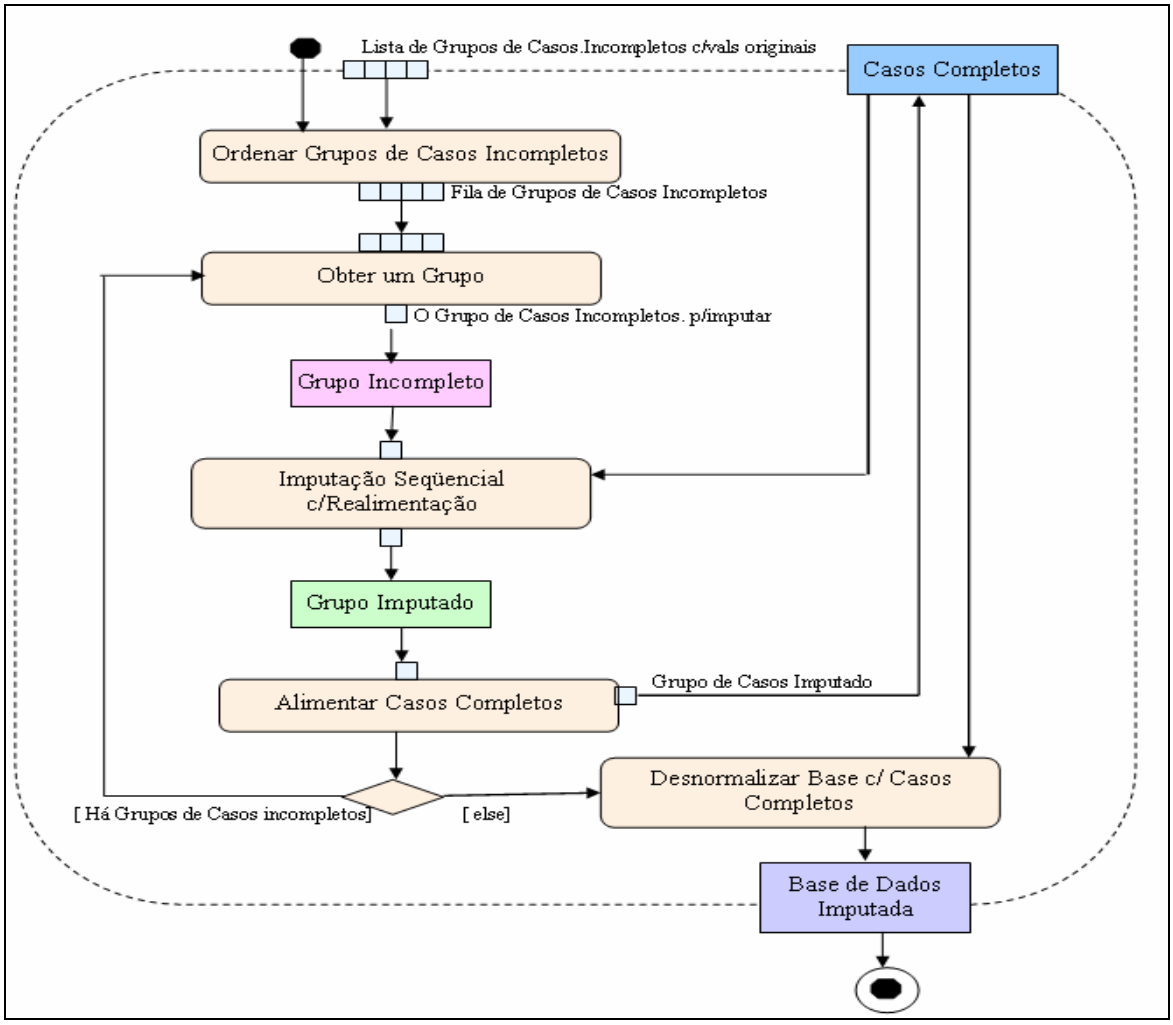


Figura 4.8 - Gerenciador de Segmentos

Inicialmente, há *m* sub-bases para imputar. São organizadas em uma fila de acordo com um critério de ordenação escolhido. A primeira sub-base incompleta será complementada com os dados da sub-base completa inicial gerada pelo módulo de pré-processamento pelo módulo responsável pela imputação. Após todos os seus valores ausentes terem sido preenchidos, ela é unida à sub-base completa. Deste modo, após a primeira iteração, a sub-base completa é composta pelos registros originalmente

completos e os registros restaurados da primeira sub-base. A seguir, a segunda sub-base incompleta é restaurada usando como fonte de dados da imputação a sub-base completa recém gerada. Após a complementação de todos os seus registros é também incorporada à sub-base completa. Este ciclo continua até não haver mais sub-bases incompletas. Portanto, a cada ciclo, a sub-base completa possui uma maior quantidade de registros, pois é composta pelos registros originalmente completos e pelos registros imputados das sub-bases restauradas nos ciclos anteriores.

No final do processo, a sub_base completa contém todos os registros originalmente existentes na base, mas normalizados. Portanto, a última ação é desnormalizá-la. O funcionamento do gerenciador de segmentos pode ser descrito pelo algoritmo abaixo:

Entrada: Conjunto de Sub-Bases Incompletas

Sub-Base Completa (Conjunto de treinamento gerado no módulo de pré-processamento)

Saída: Base de dados Imputada

Algoritmo:

Ordena as sub-bases do conjunto de Sub-Bases Incompletas, de acordo com um critério determinado, gerando uma Fila de Sub-Bases Incompletas.

Enquanto há elementos na Fila de Sub-Bases Incompletas

Sub-base a Imputar = Retira da Fila de Sub-Base incompletas

Sub-base_restaurada = Gerenciador da Imputação (Sub-base a Imputar $\cup$ Sub-base Completa)

Sub-base Completa = União da Sub-Base restaurada neste ciclo com Sub-base Completa

Fim-enquanto

Base de dados Imputada = Desnormalização da Sub-base Completa

---

4.2.3.2 Considerações sobre a ordenação das sub-bases na aplicação implementada

Como visto, a primeira sub-base incompleta é preenchida a partir da sub-base completa inicial gerada pelo módulo de pré-processamento, ou seja, com menor

quantidade de registros, mas todos com informações originais. As demais sub-bases utilizam os dados originais e os dados previamente imputados. Portanto a determinação da ordem de imputação é muito relevante. Por um lado, os valores estimados devem ser o mais próximo do original possível, para que ruídos não sejam introduzidos na base futura, por outro lado, dependendo do número de registros na base completa, a quantidade de possíveis "doadores" pode ser insuficiente para esta tarefa.

Neste estudo, foram considerados os seguintes aspectos dos grupos para determinar as ordens:

1) Quantidade de tuplas com valores ausentes**:** neste critério observa-se apenas a quantidade de tuplas existentes no segmento, independente de quantos atributos ausentes há em cada tupla. Neste critério, pode-se observar o impacto do crescimento da base em número de casos, mesmo que a qualidade da imputação não seja a melhor.

2) Quantidade de células com valores ausentes**:** neste critério observa-se apenas a quantidade total de células ausentes no segmento, independente do fato de serem do mesmo atributo, ou estarem na mesma tupla ou em tuplas distintas. Neste critério, pode-se observar o impacto do valor imputado, em detrimento da quantidade de casos.

3) Células por tupla**:** neste critério os segmentos são enfileirados de acordo com a média de valores ausentes por tupla, ou seja: número de células com valores ausentes no segmento/ número de tuplas do segmento. Busca diferenciar morfologias de ausência mais complexas de outras mais simples. Por exemplo, em uma base com 8 atributos e apenas 5 registros completos, qual segmento deve ser imputado primeiro, um segmento com 40 tuplas mas todas com um único (podendo ser o mesmo) atributo ou um segmento com 5 tuplas mas cada uma com 6 atributos ausentes? Neste critério tenta-se ponderar o crescimento da base com a qualidade da imputação.

4) Aleatória**:** sem critério específico de ordenação. Utiliza-se a ordem que os segmentos foram criados pelo algoritmo de identificação de grupos do Costa-Netto (vide apêndice III).

Neste trabalho somente aspectos quantitativos simples foram avaliados. Outros aspectos qualitativos ou quantitativos, mais elaborados e computacionalmente mais

caros, poderiam também ter sido citados, entre eles os que evidenciassem a quantidade de atributos considerados como principais. Neste caso, teria sido necessária uma análise prévia para determinar os componentes principais por algum método como, por exemplo, PCA ou algoritmos genéticos. Aspectos como a entropia ou o ganho de informação que o segmento irá proporcionar ao ser incorporado à base bem como o grau de coesão ou compactação do segmento, também podem servir como critério de ordenação. Acredita-se que considerar a compactação dos grupos e a separabilidade dos mesmos possa ser um bom critério para ordenação dos grupos e escolha automática do método de imputação intra-grupo. No entanto, embora algumas análises tenham sido feitas neste sentido, a complexidade do tema, por si só, já justifica uma tese implicando em postergá-lo como trabalho futuro.

4.2.3.2. Gerenciador da Imputação

Nesta etapa o objetivo é preencher as lacunas causadas pelos atributos ausentes. A imputação de n atributos distintos é tratada sequencialmente, isto é, como n imputações simples de um atributo. Os *n* atributos incompletos existentes são identificados e enfileirados de acordo com um critério de ordenação escolhido. Esta tarefa pode ser realizada por registro ou por sub-base. Se for escolhida a imputação por registro, todos os *n* atributos incompletos da tupla em processamento são restaurados por alguma técnica de imputação univariada para, então iniciar o processamento de uma nova tupla. Caso a imputação seja por sub-base, utilizada neste experimento, todas tuplas que estão incompletas em relação ao atual atributo alvo são restauradas antes do próximo atributo ser selecionado. Este ciclo continua até não haver mais tuplas com atributos incompletos. Cabe ressaltar que o analista do processo pode escolher se deseja ou não utilizar os valores dos atributos já preenchidos para estimar os próximos atributos bem como escolher a técnica de imputação. O processo de imputação por atributo é ilustrado pela figura 4.9.a e a figura 4.9.b mostra a imputação do atributo selecionado pelo gerenciador

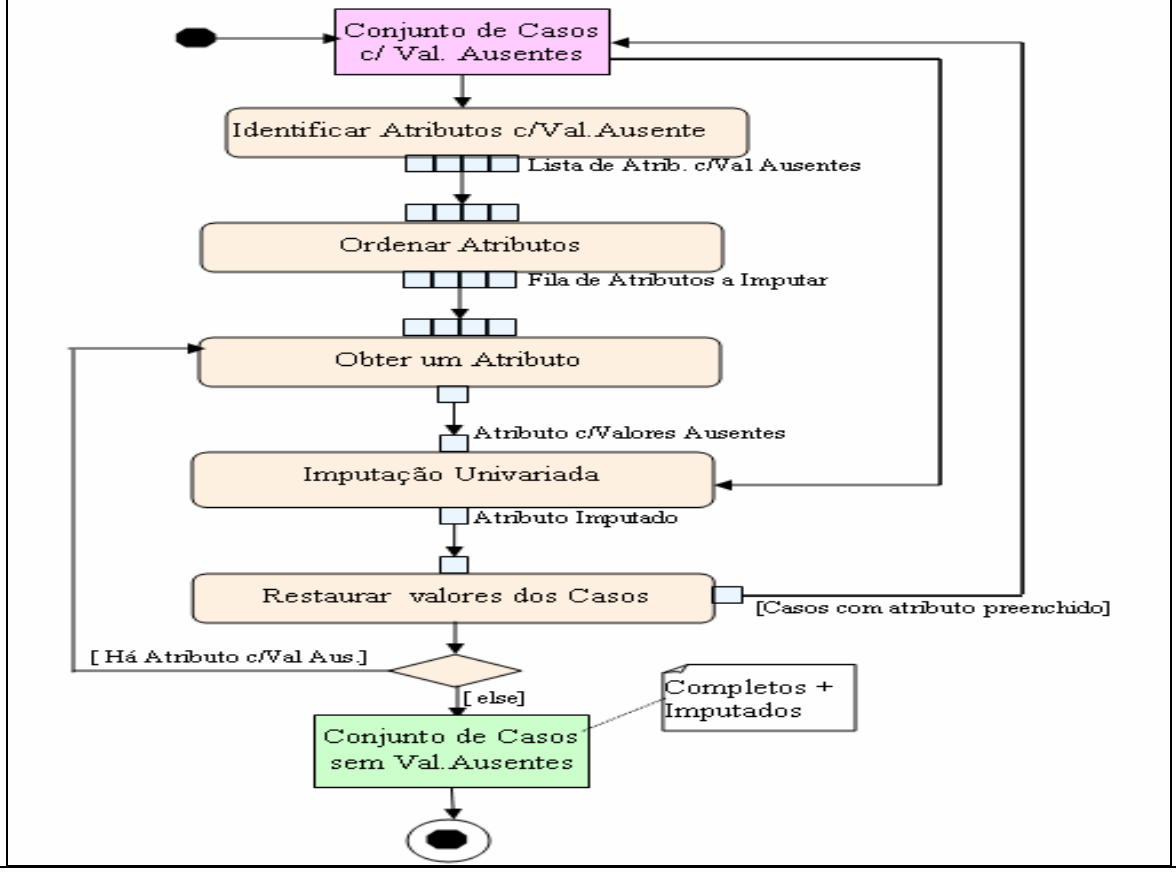


Figura 4.9a – Seleção dos Atributos a Imputar

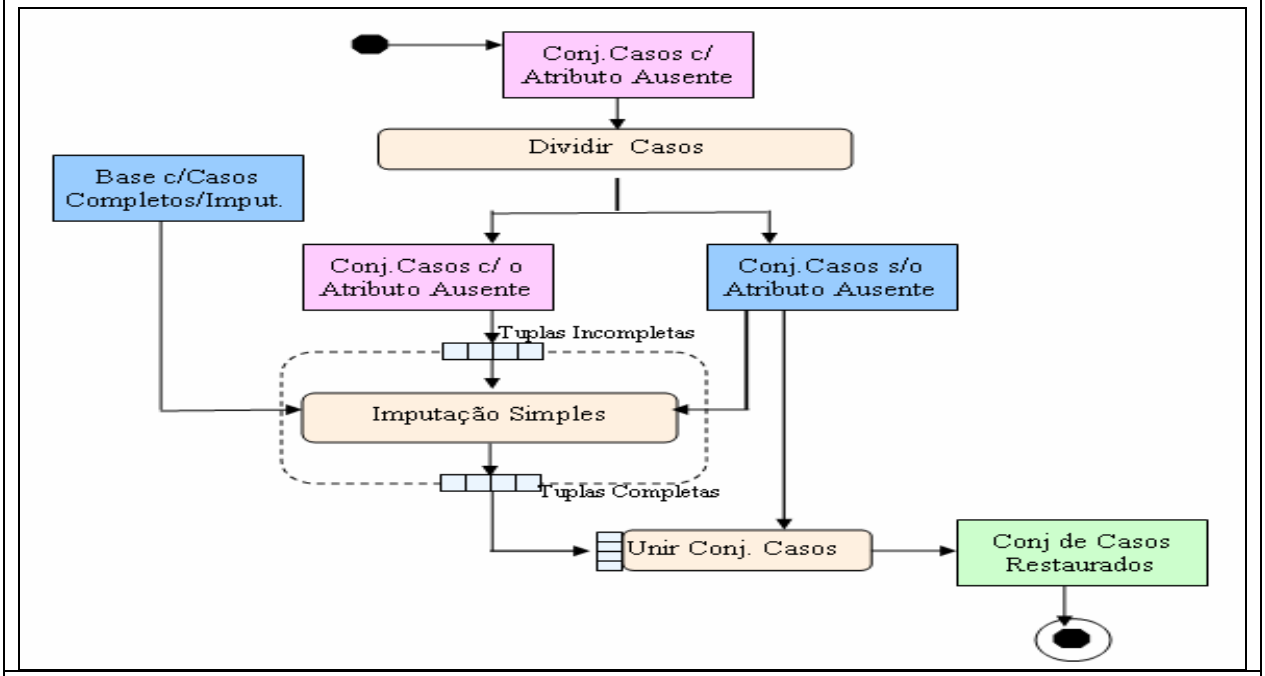


Figura 4.9b – Imputação do Atributo Selecionado

Figura 4.9 - Gerenciador de Imputação

O gerenciador de imputação é regido por dois ciclos: um ciclo externo para gerenciar a ordem de imputação dos atributos e outro ciclo interno para imputar cada

uma das células com valores ausentes para o atributo em questão. O algoritmo abaixo descreve o processo mostrado nas figuras:

Entrada: Sub-Base Incompleta

Saída: Sub-Base de dados Restaurada

Algoritmo:

Ordenar os atributos incompletos da Sub-Base Incompleta de acordo com o critério determinado, gerando uma Fila de Atributos a Imputar.

Enquanto há elementos na Fila de Atributos

Atributo a Imputar = Retira da Fila de Atributos a Imputar

Divide a Sub_Base Incompleta em dois Conjuntos:

Conjunto de tuplas com Atributo a Imputar ausente (CAAusente) e

Conjunto de tuplas com Atributo a Imputar preenchido (CACompleto)

Cria um conjunto vazio de tuplas restauradas (CTRestauradas)

Enquanto há tuplas no CAAusente

Tupla restaurada = Imputação Simples(Tupla a Restaurar, CACompleto)

Inclui Tupla restaurada no CTRestauradas

Fim-enquanto

Sub-Base Incompleta = CTuplasRestauradas ∪CACompleto (OPCIONAL)

Fim-enquanto

Sub-Base de dados Restaurada = Sub-base Incompleta

4.2.3.3 Considerações sobre a ordenação dos atributos incompletos na aplicação implementada

Neste processo também a ordem de imputação dos atributos é relevante. O primeiro atributo a ser imputado utiliza a sub-base inicial. Os demais atributos podem usar esta mesma base ou a sub-base parcialmente imputada. A mesma sub-base é usada caso a reutilização dos valores na imputação não seja desejado. Caso contrário, os valores estimados para o atributo alvo são considerados para a imputação do próximo atributo. A reutilização dos valores é interessante porque possíveis interdependências entre os atributos podem ser exploradas nos próximos ciclos, evitando inferências de valores como “homens grávidos” apresentado por van Buuren (VAN BUUREN et al. 2006).

Nesta tese, os seguintes critérios foram utilizados para determinar a ordem em que os atributos serão preenchidos:

1) Correlação**:** neste critério considera-se a soma do valor absoluto da correlação do atributo em relação aos demais atributos.

2) Quantidade de células com valores ausentes**:** neste critério observa-se a quantidade de vezes (células) que o atributo não está preenchido.

3) Aleatória: sem critério específico de ordenação**.** Utiliza-se a ordem que os atributos estão na base**.**

4.2.3.4 Considerações sobre a técnica de imputação dos atributos incompletos na aplicação implementada

Ao dividir o problema da imputação multivariada em n problemas univariados, onde n é o número de atributos com valores ausentes é possível a aplicação de qualquer técnica de imputação (simples ou múltipla) como o algoritmo *k*NN vizinhos, redes neuronais (por exemplo SOM ou back propagation), redes Bayesianas, média entre outros (GELMAN e RAGHUNATHAN 2001)(LITTLE e RUBIN 2003).

Neste trabalho foi escolhido o algoritmo *k*-NN vizinhos devido a sua vasta utilização na literatura e comprovada eficiência. Todos os parâmetros ajustáveis do método podem ser determinados pelo analista de dados no ambiente de desenvolvimento. Nesta tese esta escolha foi determinada em função de experimentos preliminares e por relatos em trabalhos relacionados:

- A escolha do *k:* o número de vizinhos considerados similares, é sempre um fator de preocupação. Experimentos foram conduzidos para k=1, por tradição. Os demais valores escolhidos para o k, 3, 5 e 10 são justificados pela literatura.
- Medidas de distância: duas medidas de distância, para avaliar o critério de similaridade, foram avaliadas: Distância Euclidiana, consagrada universalmente e a Distância de Mahalanobis(vide apêndice I). Ao selecionar a distância de Mahalanobis pretende-se considerar correlações entre as características dos objetos, questão considerada de suma importância no processo de complementação.
- Possíveis vizinhos: Usualmente o conjunto de treino, ou o conjunto de casos candidatos a doadores, só comporta as tuplas completas.No entanto, em bases com várias colunas ausentes é possível incorporar como casos candidatos, também os que têm atributos ausentes que não interfiram na imputação da

coluna alvo, conforme definição proposta por JÖNSSON e WOHLIN (2004) (vide capítulo 3) e chamados de "Casos Incompletos".

- Cálculo do valor estimado**:** aqui optou-se pela média dos valores dos atributos preenchidos dos vizinhos mais próximos.

Cabe ressaltar duas questões importantes no aproveitamento de valores imputados no processo de complementação: “Os valores imputados devem ter a mesma importância que os valores originais na predição de um novo valor ausente?” e “Casos com valores imputados podem ser considerados tão similares quanto casos originalmente completos?”. Estes questionamentos levam naturalmente a uma ponderação iferenciada dos valores originais e imputados (inclusive podendo usar conceitos da lógica fuzzy) e a modificações no cálculo das distâncias para considerar tais ponderações. Alguns ensaios nesta linha foram realizados, mas não obtiveram o êxito esperado e, devido a extensão do tema, optou-se em aprimorar a forma de ponderação e o ajuste das métricas de distância como trabalhos futuros.

4.2.3.5 Visão geral do processo de imputação em cascata

Após o detalhamento de cada uma das etapas do processo de Imputação em Cascata, a figura 4.8 apresenta sua visão completa.

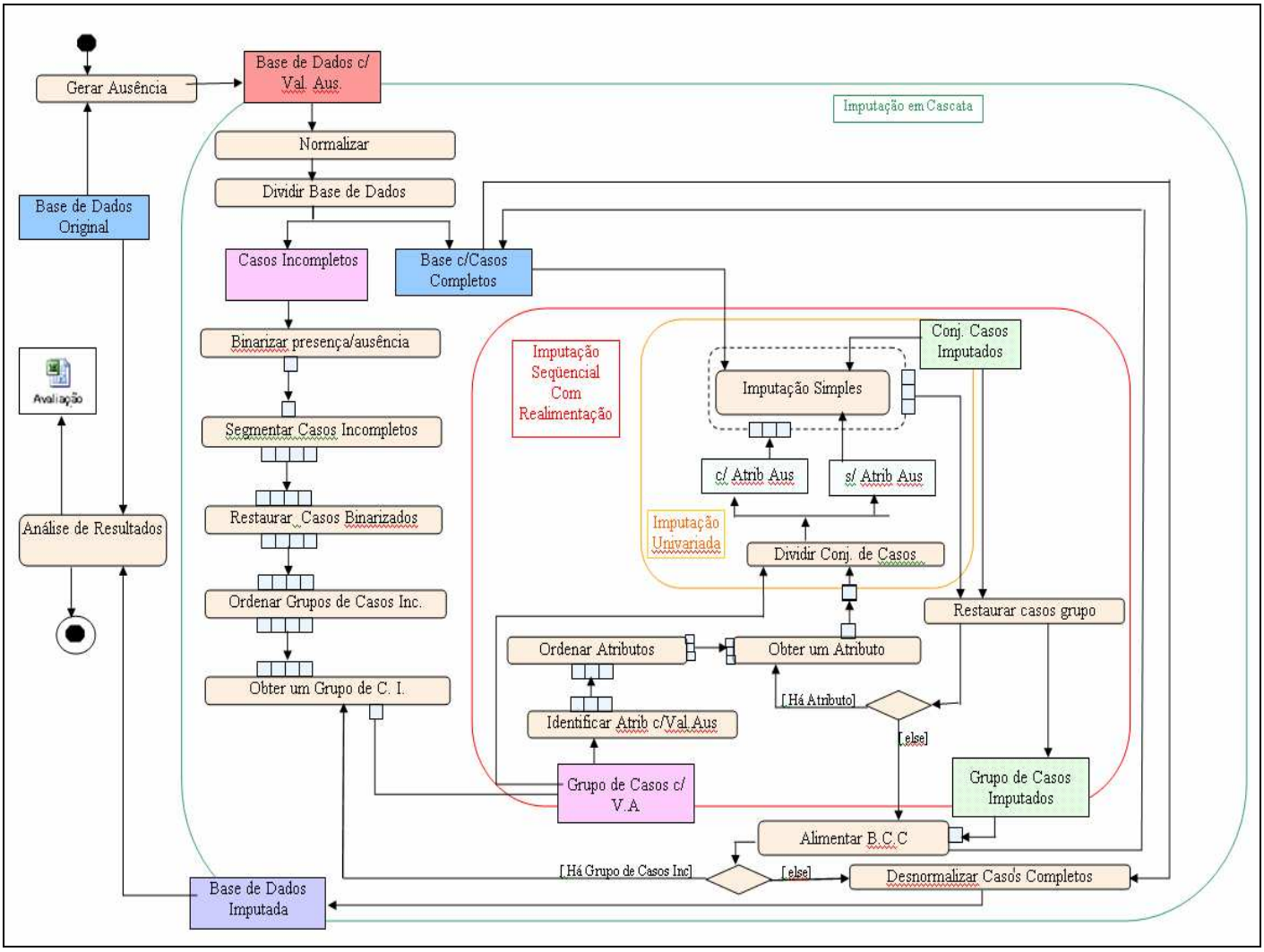


Figura 4.10 - Processo completo da Imputação em Cascata

4.3. Plataforma de desenvolvimento

A plataforma de desenvolvimento deste trabalho incorpora as macros tarefas vitais para estudos na área de complementação de valores ausentes: simulação de bases com valores ausentes, o processo de imputação (com a determinação de métodos, ordens, parâmetros, etc.), a persistência de resultados , a medição da qualidade do dado imputado e a aferição de modificações nas relações originais da base (como correlação de atributos, entre outros). Estas tarefas são realizados por 3 componentes:

1) o módulo de E*xecução de planos de imputação*, uma ferramenta desenvolvida que permite a condução dos experimentos de forma flexível e simples, seguindo alguns princípios de workflow. Embora não apresente todas as características básicas de um workflow científico, tem como principal

vantagem a geração e execução automática de todas as combinações dos parâmetros ajustáveis no processo de imputação.

2) o módulo de *Análise*, que verifica a qualidade do processo de complementação

3) o módulo *Eraser*, que simula valores ausentes em uma base de dados segundo um mecanismo e um percentual de ausência definidos pelo usuário.

Para a execução dos experimentos, os 3 componentes que compõem a plataforma são ativados de acordo com o fluxo ilustrado na figura 4.11.

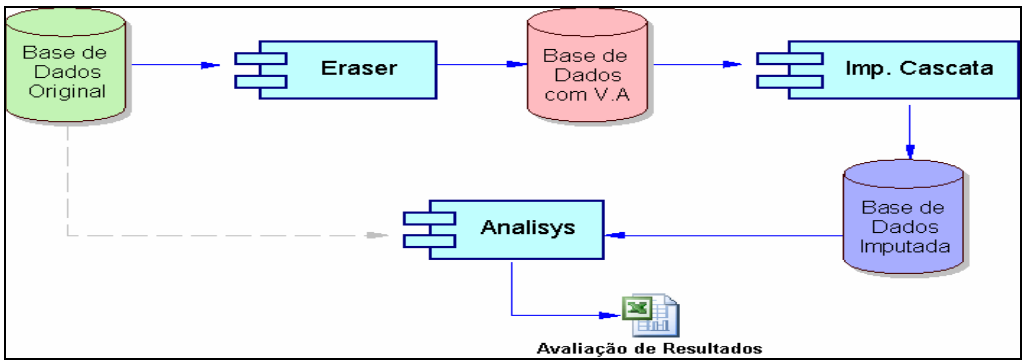


Figura 4.11. Fluxo de execução dos experimentos na plataforma desenvolvida

4.3.1 O Módulo ERASER

O módulo *Eraser* é um componente modificado do sistema *Appraisal*, (SOARES 2007) e tem o objetivo de simular valores ausentes em uma base de dados, segundo um mecanismo de ausência de dados definido.

A ausência de dados é sinalizada com um valor nulo na coluna. Quando o mecanismo de ausência é o completamente aleatório (MCAR), pode-se atribuir valores nulos a mais de um atributo, bastando, com isso, selecioná-lo no painel à esquerda, levando-os para o da direita (figura 4.10). O sistema com isso escolhe aleatoriamente um percentual, especificado na parte inferior da janela, de células (colunas) da base destes atributos. Os valores das células selecionadas são alterados para o valor nulo. Em uma mesma tupla, pode-se modificar de um a *n* dos atributos selecionados.

Quando o mecanismo de ausência escolhido é o aleatório (MAR), a seleção do atributo que terá seus valores removidos é feita no painel à esquerda. À direita, o usuário especifica as condições que farão os valores do atributo anteriormente selecionado serem alterados para nulo.

No caso onde se deseja remover atributos com o mecanismo de ausência não aleatória (NMAR), o usuário deve especificar condições independentes para cada um dos atributos que devem ter seus valores removidos.

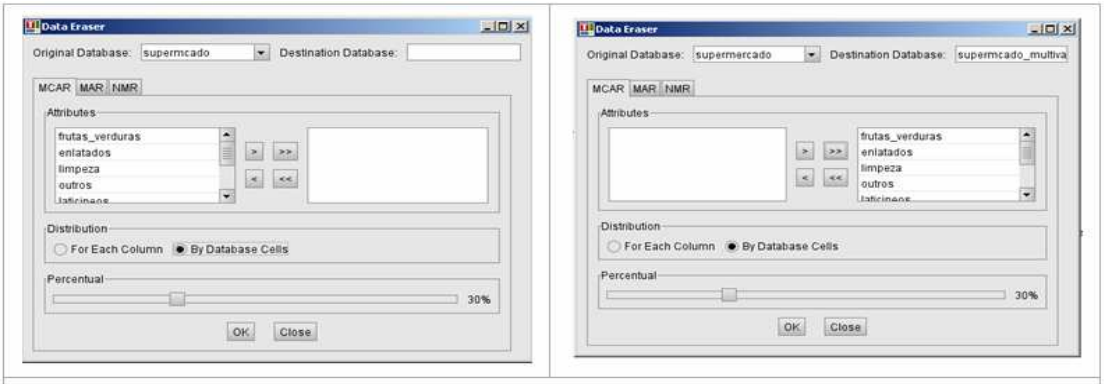


Figura 4.10: Exemplo de remoção de 30% dos valores da base, independente do atributo, com o mecanismo completamente aleatório do módulo Eraser modificado do sistema Appraisal

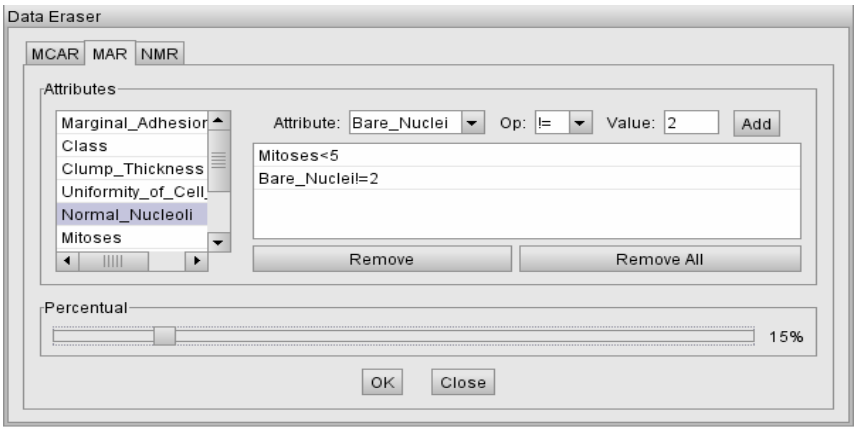


Figura 4.13: Exemplo de remoção de valores do atributo Normal_Nucleoli com o mecanismo aleatório do módulo Eraser do sistema Appraisal

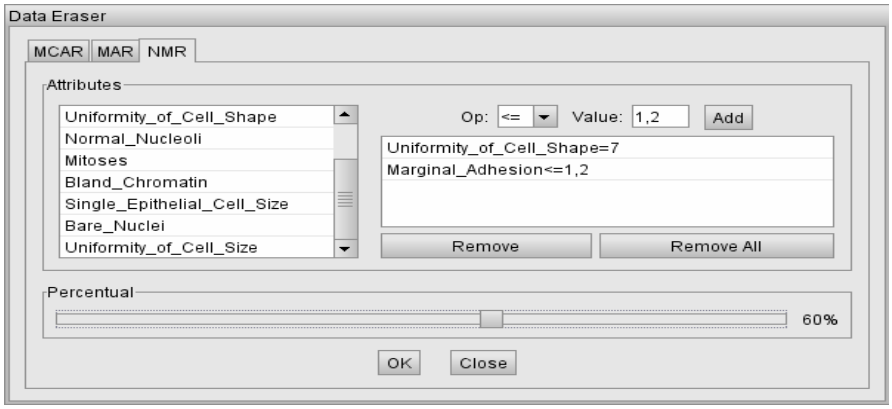


Figura 4.14: Exemplo de remoção de dois valores dos atributos Marginal_Adhesion e do atributo Uniformity_of_Cell_Shape com o mecanismo não aleatório do módulo Eraser do sistema Appraisal

4.3.2 O Módulo JBWorkflow: Ferramenta de desenvolvimento dos experimentos

Neste trabalho, para avaliarmos a abordagem proposta e analisar o impacto da ordem de imputação dos grupos e atributos na imputação multivariada, inúmeras variações de parâmetros tornam-se necessárias. Como o número total de experimentos avaliados foi muito alto, em torno de 250.000 experimentos, com diversas variações de parâmetros e tarefas de transformações de dados, decidiu-se pelo desenvolvimento de uma ferramenta com características de *workflow* para apoiar o processo de experimentação. Um *workflow* pode ser visto, de forma bem geral, como uma combinação de dados e seqüências de programas. Um experimento é caracterizado pela execução de uma seqüência de programas com suas entradas e saídas.

A utilização de *workflow*s em processos de descoberta de conhecimento em bases de dados é bastante usual. Diversas pesquisas empregam implementações de *workflows* a fim de automatizar a condução de tarefas complexas de KDD (FATTORE e ARRIGO 2005) (VAN DER AALST 2003) (GALLOUL 2005). Embora existam vários *workflows* disponíveis na literatura, optou-se pela implementação de uma ferramenta similar que possua não apenas as funcionalidades específicas da tarefa de imputação mas que também seja capaz de automatizar, com o menor esforço possível, as combinações de configuração, execução e análise comparativa dos diversos experimentos e suas variações (CASTANEDA 2008).

4.3.2.1 Requisitos Básicos Desejados

Para facilitar o entendimento dos requisitos a serem apresentados, considere o seguinte processo de negócio, relativo a imputação de dados univariada:

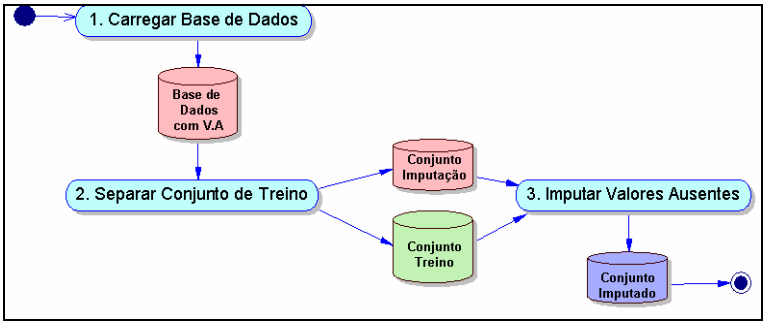


Figura 4.15 – Processo da Imputação Univariada

Cada uma das três etapas deste processo de imputação pode ser implementada de maneira diferente. Por exemplo, a etapa 1, carregar a base de dados, pode ser de um SGBD MySQL, ou Postgree; e a etapa 3, imputar valores ausentes, pode ser realizada por diversos algoritmos diferentes, como k-NN, redes neuronais , média, regressão linear, entre outros.

Ao escolher uma das alternativas possíveis, ou seja, associar componentes concretos para a realização das etapas do processo de negócio, tem-se a definição formal de um *workflow*:

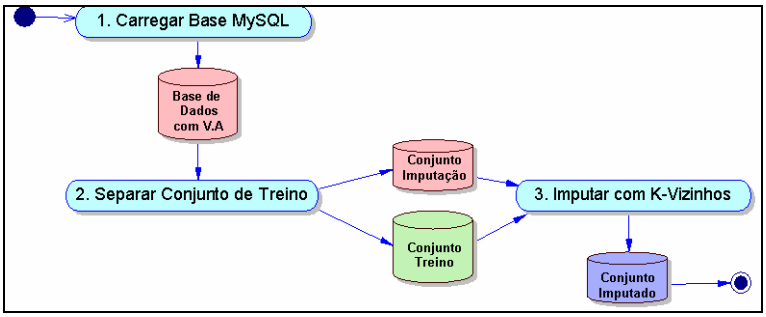


Figura 4.15 – Workflow da Imputação Univariada

Porém, apenas a definição formal de um *workflow*, não é o suficiente. Pois os componentes podem oferecer diversos parâmetros de configuração. Por exemplo, na imputação com o algoritmo dos K-Vizinhos é possível variar, entre outros parâmetrsos, o número de vizinhos e a medida de distância utilizada para o cálculo de vizinhança.

Com a definição de valores para os parâmetros de configuração de cada componente, cria-se uma instância de *workflow*, passível de execução por um WFMS:

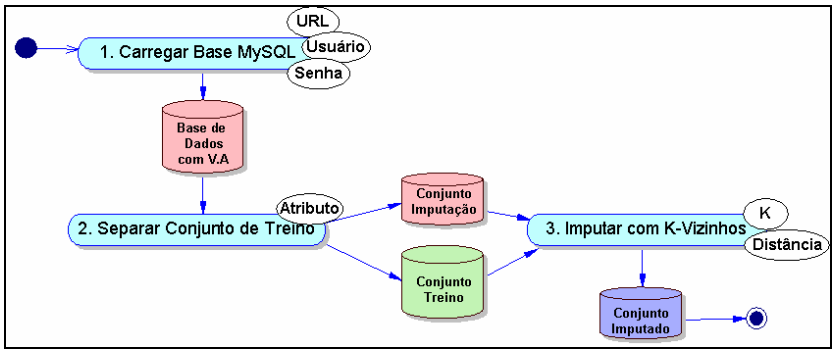


Figura 4.16 – Workflow da Imputação Univariada

Cada uma das possíveis instâncias, de cada uma das possíveis definições de *workflow* possui características próprias, gerando resultados possivelmente diferentes para o processo de imputação em uma mesma base de dados. Assim, é comum que um

analista de dados experimente diversas definições e instâncias optando por aquela que produza os melhores resultados.

Os principais requisitos levantados para o desenvolvimento da ferramenta de *workflow* que é utilizada neste trabalho estão relacionados com a automação da construção de possíveis combinações de algoritmos e parâmetros, facilitando o máximo possível o trabalho manual de experimentação por parte do analista. Portanto, são requisitos desejados:

- Construção Automática de Definições de *Workflow*: O sistema deve construir e combinar automaticamente diferentes possibilidades de definição do processo de *workflow*. Isto envolve, por exemplo, detectar diferentes implementações de cada componente, e criar definições individuais para cada combinação única de atividades.
- Construção Automática de Instâncias de *Workflow***:** O sistema deve construir e principalmente combinar automaticamente as possíveis instâncias de cada definição de *workflow* construída. Deste modo, todas as variações sobre as entradas de dados dos experimentos e parâmetros de configuração dos componentes da rede devem ser considerados. Deseja-se que a explosão combinatória provocada pela instanciação dos parâmetros, cada uma levando a uma instância executável do workflow, seja automáticamente gerada.
- Execução Automática de Instâncias: O sistema deve executar automaticamente todos os planos e instâncias construídas, em função de um único comando.
- Auditoria: O sistema deve oferecer recursos de logging, para cada execução de instância, dos componentes do *workflow*.
- Persistência**:** O sistema deve oferecer recursos de persistência individual para os resultados de cada instância de *workflow*.

4.3.2.2 Definição do processo dos experimentos da Imputação em Cascata

O fluxo que reflete os experimentos do processo de Imputação em Cascata pode ser definido de acordo com a figura 4.17. A partir da base de dados original, provoca-se um padrão de ausência em um percentual de células da base. Os casos completos são separados em uma sub-base e os incompletos são agrupados de acordo com o critério de agrupamento resultando em *m* sub-bases incompletas. Estas *m* sub-bases são

enfileiradas e processadas sequencialmente. Após a restauração de uma sub-base, esta é incorporada à base de casos completos, antes do processamento de sua sucessora.

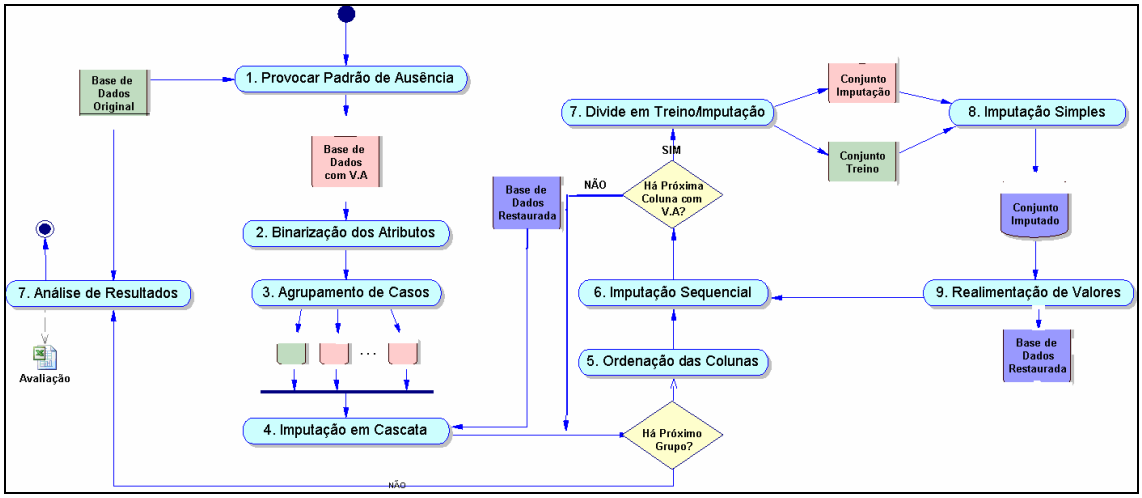


Figura 4.17 – Definição de Workflow da Imputação em Cascata

O processamento de uma sub-base envolve, inicialmente a identificação de suas colunas com valores ausentes que também são processadas sequencialmente segundo um critério de ordenação estabelecido. Para cada coluna ausente, a sub-base é, então, dividida em duas: a sub-base alvo da imputação que contém os casos com a coluna ausente, meta do preenchimento, e a outra com os demais casos. Para cada caso da sub-base alvo de imputação o valor de sua coluna meta é estimado e substituído. Antes do processamento da próxima coluna meta, pode-se substituir os casos restaurados na sub-base.

No final, os resultados, são analisados pelo erro do valor estimado e modificação na correlação original das colunas.

4.3.2.3 Usando a ferramenta

O objetivo da ferramenta é facilitar o processo de experimentação, que envolve a montagem de diferentes processos, a variação de algoritmos, a combinação de diferentes parâmetros, além da análise e comparação dos experimentos.

Os componentes do workflow obedecem a uma interface de execução que embora não seja gráfica é de fácil manipulação. São acionados através da passagem de dados e parâmetros de configuração. A saída de um ou mais componentes pode ser encadeada como entrada de outro(s), ficando a cargo da ferramenta executar o transporte destas

informações entre os passos do workflow. A estrutura funciona sobre os dois conceitos principais: (i) planos de workflow (ii) instâncias de workflow.

Um plano de workflow é a macro-estrutura do processo. Identifica os componentes que participam do processo, as regras de transformação envolvidas na execução do fluxo, e o(s) algoritmo(s) escolhidos.

Por questões de simplicidade de visualização, o exemplo de demonstração do uso da ferramenta envolve apenas a tarefa de imputação seqüencial. Esta tarefa envolve escolher uma base de dados, calcular a correlação dos atributos, ordenar os atributos e sequencialmente imputá-los (imputação simples). Para a imputação simples, a predição dos valores ausentes foi realizada pelo método k_NN com média aritmética. Portanto, não há alternativas nesta tarefa. Consequentemente este processo gera um único plano. A geração automática de vários planos acontece quando há mais de um método para uma mesma tarefa. Métodos distintos poderiam ser escolhidos para as tarefas de escolha dos possíveis doadores e imputação do atributo ausente. Esta variação está fora do escopo desta tese.

A janela superior da figura 4.16 ilustra um plano de workflow elaborado para executar a imputação seqüencial em um conjunto de dados com valores ausentes.

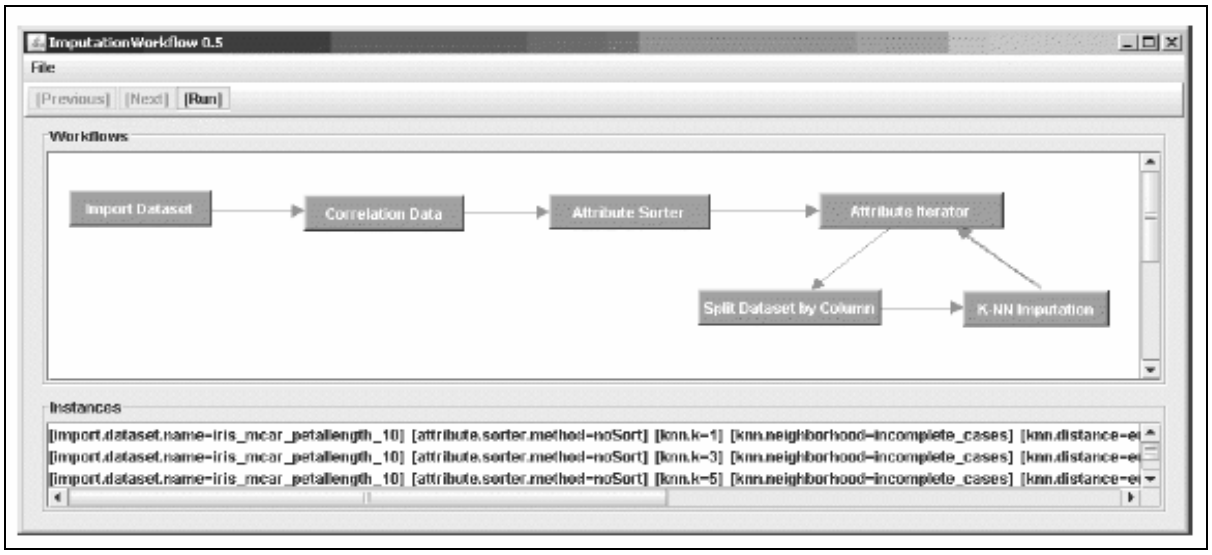


Figura 4.16. Plano de workflow para imputação no JBWorkflow

Cada plano de workflow possui uma ou mais instâncias, que são as diferentes combinações de dados e parâmetros dos componentes. Na imputação seqüencial, o caso exemplo, na subetapa *"Attribute Sorter"* pode-se escolher um ou mais critérios para determinar a ordem em que os atributos serão imputados. Para o *"k-NNImputation"*, componente escolhido para a imputação simples, há vários parâmetros ajustáveis como

o número de doadores ou casos que participam da predição, ou seja a determinação do valor de k, a métrica para similaridade, os casos que serão considerados (apenas completos ou também os incompletos), entre outros. Após a escolha do(s) valor(es) de cada um destes parâmetros, a ferramenta gera todas as possíveis instâncias, resultantes da combinação dos mesmos. Com este mecanismo é possível executar sem esforço, e de uma única vez, diversos experimentos diferentes.

No caso mostrado na figura 4.16, só o k possui alternativas de valores (1,3,5). Deste modo três instâncias foram criadas (mostradas na parte inferior da janela) que são automaticamente executadas.

A figura 4.17 ilustra o plano de workflow para a Imputação Sequencial com Realimentação. Nesta figura, é possível encontrar um componente adicional, responsável pela realimentação dos dados no experimento, antes que o laço de iteração termine.

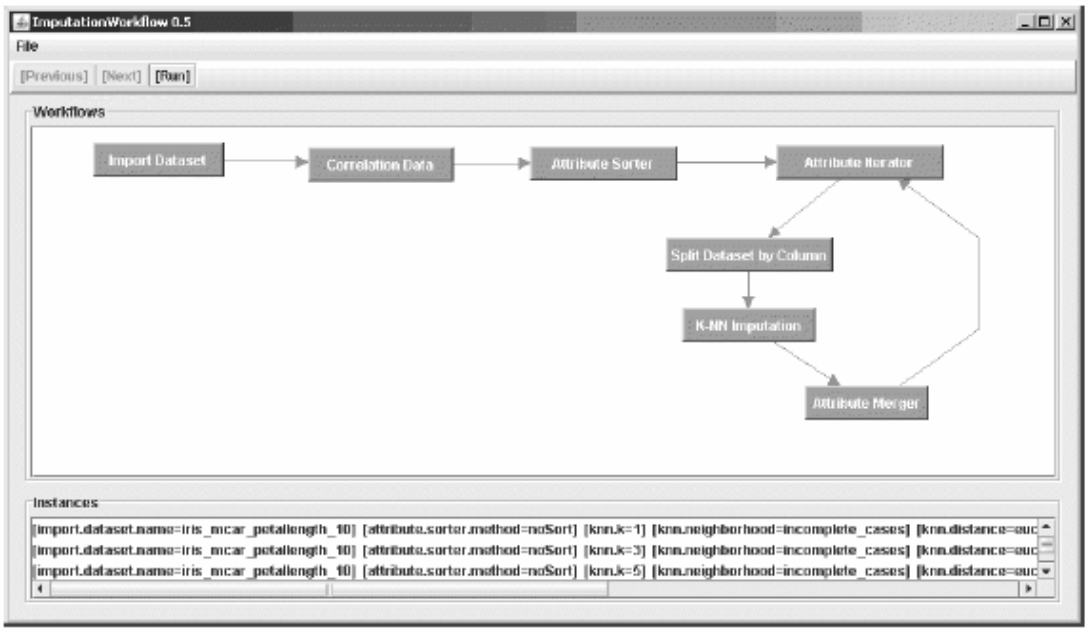


Figura 4.17. Plano de workflow para imputação com realimentação

Atualmente estes fluxos são desenhados a partir de informações armazenadas em arquivos de configuração. Os arquivos de configuração são pré-formatados, contendo entradas para as alternativas possíveis e editados via a ferramenta. Deste modo, é bastante simples habilitar os componentes e ajustar e seus parâmetros. Após o analista do processo realizar suas escolhas, a ferramenta busca os arquivos de configuração e monta todos os planos e instâncias possíveis, combinando os valores encontrados nos arquivos, exibindo-os (figuras 4.16 e 4.17). Caso o analista esteja satisfeito com os

planos/instâncias gerados, ele dispara a execução pelo botão apropriado. Os resultados da execução de cada instância são persistidos em disco, no formato de XML (para os valores imputados), e Excel (para os resultados das análises).

A figura 4.18 mostra um trecho do resultado em XML para os dados de um experimento de imputação. O XML apresenta os planos executados, e para cada plano suas instâncias. Fora os valores imputados, todos os parâmetros do contexto do workflow são armazenados, como a ordem das colunas, se houve ou não realimentação de dados, o algoritmo utilizado para imputação univariada, entre outros. Estes arquivos XML servem de base para a análise de resultados, responsável em confrontar os valores imputados com os valores originais e calcular medidas estatísticas de erro.

```
<WorkflowPlan name="iterative_knn' date="1209733343099">
  <WorkflowInstance dataset='iris_multivariate_01">
    <IterativeImputation sortedColumns="[petalwidth, petallength]" sortMethod='lessMissing' retrofeed="feed'>
      <SingleImputation column='petalWidth'>
        <ImputationAlgorithm imputationAlgorithm="knn" k='1' neighborhood='incomplete_cases' distance="euclidian" consolidation="avg"/>
        <ImputationResult>[ID|petalwidth][ID=6|0.20][ID=7|0.20][ID=8|0.40]</ImputationResult>
      </SingleImputation>
      <SingleImputation column='petallength'>
        <ImputationAlgorithm imputationAlgorithm="knn" k='1' neighborhood='incomplete_cases' distance="euclidian" consolidation="avg"/>
        <ImputationResult>[ID|petallength][ID=3|1.60][ID=4|1.40][ID=5|1.40][ID=8|1.60]</ImputationResult>
      </SingleImputation>
    </IterativeImputation>
  </WorkflowInstance>
```

Figura 4.18. Arquivo XML gerado

4.3.3 O Módulo Analysis

O módulo Analysis é responsável pela compilação dos resultados gerados pelas instâncias de workflows. A partir do arquivo XML resultante da execução da ferramenta JBWorkflow, uma planilha Excel é criada com resumos dos experimentos e cálculo dos erros.

O módulo Analysis, lê todos os arquivos XML que porventura estejam no diretório de entrada vinculados à base escolhida e gera tantas linhas quantas forem os experimentos presentes em todos os arquivos XML em um arquivo Excel. Neste arquivo

entre várias informações vitais para identificar as condições dos experimentos, há resumos da formação dos grupos e cálculo do erro por atributo, grupo e instância de workflow.

Aqui é possível escolher a medida de erro e a base a analisar.

CAPÍTULO 5

ANÁLISE DE RESULTADOS

5.1 Metodologia

5.1.1 Bases de dados Utilizadas

Os experimentos conduzidos para avaliar a imputação multivariada em cascata foram realizados em cinco bases existentes no repositório da Universidade da Califórnia, Irvine (MERZ 1998): *Iris Plants*, *Pima Indians Diabetes* e *Wisconsin Breast Cancer*, *Wine e Computer Hardware Data Set*. Entre as razões para a escolha de tais bases pode-se citar:

1. Esta tese concentra-se em atributos numéricos e as bases escolhidas possuem esta característica. A imputação multivariada sobre domínios mistos pode envolver, entre outras escolhas, a aplicação de diferentes métodos de imputação o que levaria a uma explosão combinatória podendo, inclusive, comprometer as análises pretendidas e a avaliação da abordagem. Deste modo, conjuntos de dados com atributos de tipos diferentes são objetivos de trabalhos futuros;

2. São conjuntos de dados que participam do *benchmark* da grande maioria dos trabalhos relacionados à complementação de dados ausentes e representam dados reais;

3. Apresentam características quantitativas e estruturais distintas. Esta diversidade permite avaliar a sensibilidade da imputação em cascata a domínios distintos. Embora desconsiderado, em todas as bases, exceto a *Computer Hardware*, existiu um atributo classificador dos registros. Esta informação poderia ser utilizada no processo de validação, quando, em trabalhos futuros, o intuito fosse verificar se os valores estimados modificam a classificação original. No contexto do presente trabalho, este não é o foco de avaliação.

5.1.2 Descrição das Bases

Uma vez que atributos-chave da tabela, ou qualquer outro identificador único que a tabela apresente são irrelevantes ao processo de complementação, foram eliminados dos testes.

5.1.2.1 Iris Plants

A base *Iris Plants*, construída por R. A. Fischer em meados dos anos 30 (WITTEN, FRANK, 2005) é referenciada em praticamente todos os trabalhos relacionados. É uma base com poucos atributos e bem comportada. Registra as medidas de comprimento e largura de pétalas e caule de três tipos de plantas: *Virginica*, *Versicolor* e *Setosa*. Há 150 tuplas na base, eqüi-distribuídas nas três classes, ou seja, cada classe tem 50 representantes. Nenhum registro apresenta valores ausentes.

O atributo-classe armazena o tipo da planta, em função das medidas de caule e pétalas. O atributo *sepallength* armazena o comprimento do caule enquanto o *sepalwidth* sua largura. Em relação às pétalas, *petallength* representa o comprimento e o *petalwidth* a largura das mesmas.

A descrição dos atributos é a seguinte:

	Tipo	Unidade	Valor mínimo	Valor máximo	Média	Desvio-Padrão
1. sepallength	Real	Cm	4.3	7.9	5.84	0.83
2. sepalwidth	Real	Cm	2.0	4.4	3.05	0.43
3. petallength	Real	Cm	1.0	6.9	3.76	1.76
4. petalwidth	Real	Cm	0.1	2.5	1.20	0.76

Os valores máximo e mínimo das colunas mostram uma base com um domínio pequeno de valores para os atributos. Esta pouca diversidade em cada atributo, entre atributos e a forte correlação (descrita abaixo) de seus atributos são suas principais características. Outra característica peculiar desta base é a existência de um atributo patognomônico para a classe *Setosa*. Quando o atributo *petalwidth* possui valores menores ou iguais a 0.6 cm este caso pertence à classe *Setosa*, independente dos demais atributos.

A matriz de correlação entre os atributos é a mostrada abaixo:

	sepallength	*sepalwidth*	*petalwidth*	*petallength*
sepallength	1.00	-0.11	0.82	0.87
sepalwidth	-0.11	1.00	-0.36	-0.42
petalwidth	0.82	-0.36	1.00	0.96
petallength	0.87	-0.42	0.96	1.00

A matriz nos mostra que os atributos *petallength*, *petalwidth* e *sepallength* possuem correlações significativas entre si, entre 0.82 e 0.96 (a mais alta, entre *petallength* e *petalwidth*). Por outro lado, os baixos valores de correlação entre atributo *sepalwidth* e os demais revelam que suas medidas não são influenciadas pelos valores dos demais atributos.

5.1.2.2 Pima Indians Diabetes

A base *Pima Indians Diabetes*, cedida por Vincent Sigillito, armazena dados de mulheres com diabetes e mais de 21 anos da tribo *Pima* do Arizona, EUA. Os registros contêm um atributo-classe que indica se a paciente é ou não diabética. A base apresenta 768 registros, sendo 500 de mulheres que não possuem diabetes, e 268 com resultado positivo para esta doença. Os atributos são *age (*idade), *blood pressure (*pressão sanguínea diastólica), *body mass (*índice de massa corporal - peso/altura^2), *Glucose concentration (*nível de glicose no sangue duas horas após a ingestão de glicose concentrada de um teste de tolerância à glicose), *pedigree function (*função de características hereditárias da diabetes), *pregnancy times (*número de vezes que a paciente engravidou) *serum insulin(*nível de insulina no sangue), *skin fold thickness (*espessura da pele do tríceps). Outras características sobre esta base são:

	Tipo	Unidade	Valor mínimo	Valor máximo	Média	Desvio-Padrão
1. age	int		21	81	30,86	10,18
2. Blood pressure	int	mm Hg	24	110	70,66	12,48
3. Body mass	real	Kg/m^2	18,2	67,1	33,08	7,01
4. Glucose concentration	int		56	198	122,62	30,82
5. Pedigree function	real		0,085	2,42	0,52	0,34
6. Pregnancy times	int		0	17	3,30	3,20
7. Serum insulin	int	mu U/ml	14	846	156,05	118,69
8. Skin fold thickness	int	Mm	7	63	29,14	10,50

Os valores máximo e mínimo das colunas mostram uma base com um domínio mais amplo de valores para os atributos do que a *Íris Plants*. Há uma boa diversidade de valores entre os atributos sendo que alguns (*Serun insulin e Glucose Concentration)* apresentam um desvio padrão significativo. Os atributos da base são bem independentes conforme relatado adiante.

Embora todos os atributos estejam preenchidos, constata-se a presença de ruído em alguns deles, ou seja, ocorrências de atributos com valor zero onde este valor não pertence ao domínio do mesmo. A situação encontrada foi a seguinte: *blood_pressure*: 35 registros com valor zero; *body_mass*: 11 registros com valor zero; *glucose_concentration*: 5 registros com valor zero. *skin_fold_thickness*: 227 registros com valor zero;*serum_insulin*: 374 registros com valor zero. No total, encontram-se 376 registros que possuem um destes atributos com valor zero. Há dúvidas se este valor é ruído ou valores não medidos e neste caso considerados ausentes. Esta hipótese justifica-se uma vez que não é possível a existência de pessoas com massa corporal ou pressão sanguínea diastólica igual a zero. Portanto, uma vez que para a validação dos experimentos a base original deve estar completa e ser confiável, as tuplas que apresentaram valor zero em quaisquer dos atributos numéricos foram retirados, à exceção do atributo *pregnancy_times*, já que é perfeitamente plausível uma paciente não ter gestações. Com isso, a base passou a ter 392 registros, com 262 casos de diabetes comprovadas, e 130 sem apresentar a doença (SOARES, 2007).

A matriz de correlação da base *Pima Indians Diabetes* mostra que os atributos têm pouca correlação entre si, ou seja, são bastante independentes. As maiores medidas de correlação não chegam a 70%, sendo que as maiores delas ocorrem entre os atributos *age* e *pregnancy_times*, de 0.68; *body_mass*, *skin_fold_thickness*, com 0.66; e *serum_insulin* e *glucose_concentration*, com 0.58. Esta característica, unida ao fato do mecanismo de ausência dos dados ser completamente aleatório (MCAR), pode dificultar o processo de complementação de dados nesta base.

A matriz de correlação desta base é a representada a seguir.

Atributo	*Pedgr.Func*	*Glucos Conctr*	*Body mass*	*Skin thickns*	*Blood pres*	*Age*	*Serum insulin*	*Pregn times*
Pedigree function	1.00	0.14	0.16	0.16	-0.02	0.09	0.14	0.01
Glucose concentration	0.14	1.00	0.21	0.20	0.21	0.34	0.58	0.20
Body mass	0.16	0.21	1.00	0.66	0.30	0.07	0.23	-0.03
Skin old thickness	0.16	0.20	0.66	1.00	0.23	0.17	0.18	0.09
Blood Pressure	-0.02	0.21	0.30	0.23	1.00	0.30	0.10	0.21
Age	0.09	0.34	0.07	0.17	0.30	1.00	0.22	0.68
Serum Insulin	0.14	0.58	0.23	0.18	0.10	0.22	1.00	0.08
Pregnancy Times	0.01	0.20	-0.03	0.09	0.21	0.68	0.08	1.00

5.1.2.3 Wisconsin Breast Cancer

Esta base, cedida por Dr. William H. Wolberg do hospital da Universidade de Wisconsin, registra dados sobre pacientes com câncer de mama. Representa um problema relativamente complexo com nove atributos contínuos e duas classes que indicam um neoplasia benigna ou maligna. Há 682 casos completos (de 699 existentes), onde 239 indicam pacientes que apresentam neoplasia de mama maligna e 443 que não apresentam. Os demais atributos são: *Uniformity of Cell Size, Clump Thickness, Bland Chromatin, Uniformity of Cel Shape, Marginal Adhesion, Mitoses, Bare Nuclei, Normal Nucleoli, Single Epithelial Cell Size.*

Os dados relativos aos atributos são os seguintes:

	Tipo	Valor mínimo	Valor máximo	Média	Desvio-Padrão
1. Bare_Nuclei	int	1	10	3,54	3,64
2. Bland_Chromatin	int	1	10	3,44	2,44
3. Clump_Thickness	int	1	10	4,43	2,82
4. Marginal_Adhesion	int	1	10	2,83	2,86
5. Mitoses	int	1	10	1,60	1,73
6. Normal_Nucleoli	int	1	10	2,87	3,05
7. Single_Epithelial_Cell_Size	int	1	10	3,23	2,22
8. Uniformity_of_Cell_Shape	int	1	10	3,21	2,98
9. Uniformity_of_Cell_Size	int	1	10	3,15	3,06

Os valores máximo e mínimo das colunas mostram uma característica muito interessante: todos os atributos variam na mesma faixa de valores com médias e desvios padrões similares, sugerindo uma base bastante coesa, onde há necessidade de uma maior granularidade para detectar diferenças.

A matriz de correlação é a que se segue:

Atributo	*Unif Cell Size*	*Clump Thick*	*Bland Chrm*	*Unif Cel Shp*	*Margl Adh*	*Mito ses*	*Bare Nuc*	*Norm Nuci*	*Single Epith CelSz*
Uniformity CelSize	1.00	0.64	0.76	0.91	0.71	0.46	0.69	0.72	0.75
Clump Thickness	0.64	1.00	0.55	0.65	0.49	0.35	0.59	0.53	0.52
Bland Chromatin	0.76	0.55	1.00	0.74	0.67	0.35	0.68	0.67	0.62
Uniformity of Cell Shape	0.91	0.65	0.74	1.00	0.69	0.44	0.71	0.72	0.72
Marginal Adhesion	0.71	0.49	0.67	0.69	1.00	0.42	0.67	0.60	0.59
Mitoses	0.46	0.35	0.35	0.44	0.42	1.00	0.34	0.43	0.48
Bare Nuclei	0.69	0.59	0.68	0.71	0.67	0.34	1.00	0.58	0.59
Normal Nucleoli	0.72	0.53	0.67	0.72	0.60	0.43	0.58	1.00	0.63
Single Epithelial Cell Size	0.75	0.52	0.62	0.72	0.59	0.48	0.59	0.63	1.00

Observa-se que o conjunto de dados *Wisconsin Breast Cancer* apresenta uma correlação significativa entre seus atributos. Oito de seus nove atributos estão diretamente correlacionados, com índices maiores do que 0.5. O atributo *Mitoses* é a exceção e possui correlações menores que 0.5 com todas as demais colunas. A maior correlação entre colunas acontece entre os atributos *Uniformity_of_Cell_Size* e *Uniformity_of_Cell_Shape* (0.91), relação bastante explorada na citopatologia.

5.1.2.4 Computer Hardware

A base *Computer Hardware*, também conhecida como *Machine*, foi construída por Phillip Ein-Dor e Jacob Feldmesser, em 1987, e mantém registros do desempenho da CPU em função do período do relógio, tamanho da memória e número de canais. Não é uma base muito usual, mas possui características distintas das demais: poucos casos, uma maior quantidade de atributos e não possui atributo-classe. Em vez de uma classe, esta base armazena para cada caso, dois valores numéricos que representam medidas de desempenho da CPU em função de outros componentes físicos de um computador, uma publicada pelo fornecedor do equipamento e outra estimada pelos autores da base. O atributo categórico que registra o nome do fabricante foi desconsiderado.

Há 209 tuplas e nenhuma apresenta valores ausentes. Os demais atributos são: *MachineCycleTime (*tempo de um ciclo), *MemMin* (tamanho mínimo da memória),

MemMax (tamanho máximo da memória), *Cache* (tamanho da memória cache), *ChannelsMin* (número mínimo de canais), *ChannelsMax* (número máximo de canais), *PublRelPerf* (desempenho relativo publicado) e *EstRelPerf* (desempenho relativo estimado). Escalonando em intervalos o desempenho fornecido pelo fabricante, tem-se a seguinte distribuição de casos:

Intervalo de valores	**Número de casos**
0-20	31
21-100	121
101-200	27
201-300	13
301-400	7
401-500	4
501-600	2
Acima de 600	4

A descrição dos atributos é a seguinte:

	Tipo	Unidade	Valor mínimo	Valor máximo	Média	Desvio-Padrão
1.MemMax	int	Kb	64	64.000	11.796,1	11.726,6
2. MemMin	int	Kb	64	32.000	2.868	3.878,7
3.ChannelsMax	int	Unid	0	176	18,2	26,0
4.ChannelsMin	int	Unid	0	52	4,7	6,8
5.EstRelPerf	int		15	1.238	99,3	154,8
6.MachineCycleTime	int	μs	17	1.500	203,8	260,3
7.PublRelPerf	int		6	1.150	105,6	160,8
8. Cachê	int	Kb	0	256	25,2	40,6

Os valores máximo e mínimo das colunas desta base demonstram tratar-se de uma base bastante diversificada, apresentando atributos com uma faixa de valores no domínio dos atributos bastante ampla e outras bem estreitas. Há atributos com valores inconsistentes dentro da faixa possível. Por exemplo, sabe-se que os valores de memória são medidos em potências de 2, o que delimita a diversidade dos valores. Esta característica também deve ser um complicador para o processo de imputação que não considera esta peculiaridade do atributo (este problema já pode ser observado no valor médio dos atributos mostrado na tabela) e pode estimar ruídos. O desvio padrão também é bem expressivo.

A matriz de correlação entre os atributos é a mostrada abaixo:

	Mem Max	*Mem Min*	*Chan Max*	*Chan Min*	*Estim Rel Perf*	*Mach Cycle Time*	*Publ Rel Perf*	*Cachê*
Mem Max	1.00	0.76	0.53	0.56	0.90	-0.38	0.86	0.54
Mem Min	0.76	1.00	0.27	0.52	0.82	-0.34	0.79	0.53
Channels Max	0.53	0.27	1.00	0.55	0.59	-0.25	0.61	0.49
Channels Min	0.56	0.52	0.55	1.00	0.61	-0.30	0.61	0.58
Estimated RelPerf	0.90	0.82	0.59	0.61	1.00	-0.29	0.97	0.65
Machine CycleTime	-0.38	-0.34	-0.25	-0.30	-0.29	1.00	-0.31	-0.32
Publicated RelPerf	0.86	0.79	0.61	0.61	0.97	-0.31	1.00	0.66
Cache	0.54	0.53	0.49	0.58	0.65	-0.32	0.66	1.00

Neste conjunto de dados há sete atributos diretamente correlacionados e um (*Machine Cycle Time*) inversamente correlacionado com todos os demais. Os atributos diretamente correlacionados apresentam uma correlação significativa entre seus atributos (com índices em sua maioria maiores do que 0.5) enquanto o *Machine Cycle Time* (inversamente correlacionado) o é com menor expressão (índices inferiores a 0.4).

5.1.2.5 Wine

Esta base, cedida por M. Forina em 1991, registra dados sobre análises químicas de vinhos provenientes da mesma região da Itália, mas derivados de três cultivadores distintos. As análises determinam as quantidades de treze componentes encontrados em cada um dos três tipos de vinho Representa um problema relativamente complexo com 13 atributos contínuos e três classes que indicam o tipo do vinho. Há 59 casos da primeira classe, 71 da segunda e 48 da terceira. Os demais atributos são: *Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline.*

Os dados relativos aos registros são os seguintes:

	Tipo	Valor mínimo	Valor máximo	Média	Desvio-Padrão
1. Alcohol	real	11,03	14,83	12,98	0,89
2. Malic acid	real	0,74	4,43	1,90	0,78
3. Ash	real	1,36	3,23	2,34	0,30
4. Alcalinity of ash	real	10,60	30	18,65	3,29
5. Magnesium	int	70	162	100,72	15,39
6. Total phenols	real	1,10	3,88	2,53	0,56
7. Flavanoids	real	0,57	5,08	2,49	0,75
8. Nonflavanoid phenols	real	0,13	0,66	0,33	0,11
9. Proanthocyanins	real	0,41	3,58	1,75	0,54
10. Color intensity	real	1,28	8,90	4,27	1,63
11. Hue	real	0,70	1,71	1,07	0,16
12. OD280/OD315 of diluted wines	real	1,59	4,00	2,95	0,48
13. Proline	int	278	1680	811,66	349,77

Os valores máximo e mínimo das colunas desta base demonstram um conjunto com onze atributos com uma faixa de valores não muito grande e nem muito diferentes entre si. Os atributos *Proline e Magnesium* apresentam domínio bem mais amplo, sendo o último com um grande desvio padrão.

A matriz de correlação é a que se segue:

	Prot Cyan.	*Magn.*	*hue*	*Non Flav.*	*ash*	*Alcalin ash*	*Total phen*	*Pro line*	*Alco hol*	*Color intens*	*Flava noids*	*OD280 OD315*	*Malic acid*
Proantho Cyanin	1.00	0.24	0.30	-0.37	0.01	-0.20	0.61	0.33	0.14	-0.03	0.65	0.52	-0.22
Magnesium	0.24	1.00	0.06	-0.26	0.29	-0.08	0.21	0.39	0.27	0.20	0.20	0.07	-0.05
Hue	0.30	0.06	1.00	-0.26	-0.07	-0.27	0.43	0.24	-0.07	-0.52	0.54	0.57	-0.56
Nonflavanoid phenols	-0.37	-0.26	-0.26	1.00	0.19	0.36	-0.45	-0.31	-0.16	0.14	-0.54	-0.50	0.29
Ash	0.01	0.29	-0.07	0.19	1.00	0.44	0.13	0.22	0.21	0.26	0.12	0.00	0.16
Alcalinity of ash	-0.20	-0.08	-0.27	0.36	0.44	1.00	-0.32	-0.44	-0.31	0.02	-0.35	-0.28	0.29
Total phenols	0.61	0.21	0.43	-0.45	0.13	-0.32	1.00	0.50	0.29	-0.06	0.86	0.70	-0.34
Proline	0.33	0.39	0.24	-0.31	0.22	-0.44	0.50	1.00	0.64	0.32	0.49	0.31	-0.19
Alcohol	0.14	0.27	-0.07	-0.16	0.21	-0.31	0.29	0.64	1.00	0.55	0.24	0.07	0.09
Color intensity	-0.03	0.20	-0.52	0.14	0.26	0.02	-0.06	0.32	0.55	1.00	-0.17	-0.43	0.25
Flavanoid	0.65	0.20	0.54	-0.54	0.12	-0.35	0.86	0.49	0.24	-0.17	1.00	0.79	-0.41
OD280/OD315 of diluted wines	0.52	0.07	0.57	-0.50	0.00	-0.28	0.70	0.31	0.07	-0.43	0.79	1.00	-0.37
Malic Acid	-0.22	-0.05	-0.56	0.29	0.16	0.29	-0.34	-0.19	0.09	0.25	-0.41	-0.37	1.00

Neste conjunto de dados, a maioria dos atributos é baixamente correlacionado. Há atributos diretamente correlacionados, inversamente correlacionados e alguns independentes (correlação inferior a |0.15|, inclusive 0). Os atributos *Proantho Cyanins*, *Flavanoids, OD280/OD315 of diluted wines* e *Total phenols* se distinguem dos demais e são bastante dependentes entre si, apresentando um índice de correlação superior a 0.6.

5.1.3 Parâmetros relativos à ausência dos dados

Com uso do módulo *Eraser*, e a partir das bases de dados originais, foram geradas versões com padrão de ausência multivariado. O mecanismo de ausência escolhido foi o MCAR e, para garanti-lo, após a determinação do percentual de ausência, 10%, 20% e 30% , este índice é aplicado nas células da base. Por exemplo, a base *Íris Plants*, possui 150 registros e 4 atributos, totalizando 600 células. Ao determinar-se um percentual de 10% de ausência, o módulo *Eraser* escolhe aleatoriamente 60 células para “sujar”, independente de registro. Sendo assim, algumas tuplas, terão um valor ausente, outras dois valores ausentes até um máximo de três valores ausentes, ou seja, n-1 colunas (sendo n o número de atributos da base). Para cada percentual de ausência foram criadas três versões de cada base. Este processo de criação de bases resultou em 45 bases

distintas, três versões de três percentuais de ausência de cada uma das cinco bases, nas quais a imputação em cascata foi testada.

Cada base do experimento é identificada com o nome da base original, concatenado ao padrão de ausência, seguido do percentual de valores ausentes e do número da versão. Por exemplo, a base *iris__multivariate_10_01* refere-se à primeira versão da base *Íris Plants* com 10% de valores ausentes.

Os percentuais de valores ausentes 10%, 20% e 30% foram escolhidos pela suas ocorrências nos trabalhos relacionados, principalmente na área de Bioinformática. Outros percentuais, maiores e intermediários, são alvos de trabalhos futuros. A escolha de diferentes índices de ausência, em detrimento de diferentes versões de bases com mesmo percentual, deve-se ao fato de ser de interesse o estudo do desempenho de combinações distintas da ordem de imputação à medida que a quantidade de valores ausentes aumenta..

Os demais mecanismos de ausência (MAR e NMAR) são intenções de trabalhos futuros.

5.1.4 Parâmetros dos algoritmos

5.1.4.1 A imputação de uma célula

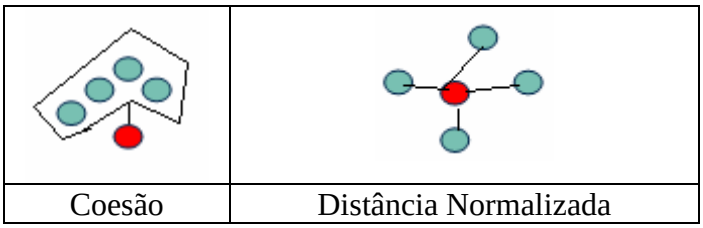
Conforme descrito no capítulo três, há diversos algoritmos para preencher uma lacuna da base, isto é, o valor de um atributo de uma tupla (uma célula da base). Nesta fase, optou-se pela simplicidade computacional da média dos valores do atributo alvo nos casos mais similares à tupla alvo, e que compõem o conjunto de casos selecionados.

Mesmo descartando os demais algoritmos e considerando apenas a média há alternativas. As principais opções são: média aritmética determinística e média estocástica. Na segunda opção, é usual modificar a média aritmética acrescentando-lhe (ou diminuindo) um valor aleatoriamente gerado (em geral delimitado pela variância do atributo) procurando não reduzir a variância dos dados e assim suavizar este problema inerente à média determinística e tornando-a menos tendenciosa. No entanto, no trabalho de SOARES (2007), do qual este é uma continuidade, a média estocástica nas bases experimentadas não obteve os melhores resultados, conforme resumo abaixo que mostra o índice segundo o qual cada uma das opções acima obteve o melhor desempenho:

	Média Determinística	Média Estocástica
Iris Plants	93,75%	6,25%
Pima Indians	95,63%	4,38%
Breast Cancer	83,89%	16,11%

Com base nestes resultados, embora esta tendência tenha sido observada no escopo dos experimentos da referida tese e a princípio não pode ser generalizada para qualquer tipo de base de dados (SOARES, 2007), optou-se pelo cálculo do valor estimado apenas pela média aritmética.

Ensaios foram realizados atribuindo pesos que expressem a credibilidade nos valores que participam da média, uma vez que alguns destes valores podem ser resultantes de imputações prévias. A idéia básica no cálculo destes pesos reside no fato de que se o grupo de casos que participam do cálculo da média for similar ao da tupla alvo, a crença que o valor estimado é próximo do real, aumenta. Em anexo a este pensamento, como a similaridade é calculada, em geral, por alguma métrica de distância, e são baseadas em médias, pode-se considerar, também, o quão coeso é o grupo conforme ilustração 5.1:



Coesão	Distância Normalizada

Figura 5.1. Representação de métricas de peso

Desta forma, dois cálculos de pesos foram ensaiados: distância normalizada e coesão. O peso denominado distância normalizada, leva em consideração uma média normalizada da distância da tupla alvo para as tuplas que participaram do cálculo do valor e pode ser definida por: $peso\_celula_i = \frac{1}{N}\sum_{i=1}^{N}\frac{distância_i - \min dist}{\max dist - \min dist}$ onde N é o número de tuplas doadoras. Portanto, o peso associado aos valores dos casos que participaram do cálculo da média desta célula e da $distância_i$ são normalizados em relação à faixa de valores possíveis para o atributo alvo. A segunda forma de calcular o

peso é: $peso\_celula_i = 1 - \frac{\sigma}{\mu}$ onde são consideradas a média e o desvio padrão dos valores das células dos casos selecionados. A relação $\frac{\sigma}{\mu}$ é considerada na estatística como uma *medida de espalhamento*, portanto, 1 – *espalhamento* foi considerado coesão.

Nos primeiros testes, a média ponderada pelo peso normalizado apresentou um comportamento um pouco melhor que as demais. No entanto, as variações estão em torno de 0.01%, e a explosão combinatória que estas medidas introduzem nos experimentos, desencorajou a exploração destas formas de ponderação, ficando, também, como trabalhos futuros.

5.1.4.2 A imputação de um Atributo

5.1.4.2.1 A ordem de imputação de um atributo

A imputação multivariada de *n* atributos foi tratada como uma seqüência de *n* imputações univariadas. Ao dividir o problema deste modo, introduz-se a necessidade de estabelecer em que ordem os atributos serão imputados. Esta ordenação impacta em todo o processo de complementação. Inicialmente estas decisões foram influenciadas pelos trabalhos relacionados que, em geral, ordenam os atributos em função da quantidade de células ausentes no mesmo. No entanto, acredita-se que outros fatores adicionais à freqüência de ausência devam ser considerados, sendo estes, também, alvo de estudos. Portanto, há cinco ordens possíveis (foram denominadas como se apresentam na ferramenta desenvolvida):

1. *lessCorrelation* – inicia com o atributo menos correlacionado, (soma do valor absoluto da correlação deste com os demais) até o mais correlacionado.
2. *moreCorrelation* – inverso do anterior. Inicia com o mais correlacionado até o menos correlacionado.
3. *lessMissing* – critério usual. Inicia pelo atributo que tem menos células ausentes até o atributo com mais células ausentes.
4. *moreMissing* - inverso do anterior. Inicia com o atributo com mais células com valores ausentes para o atributo com menos células ausentes.
5. *noSort* - utiliza a ordem definida pelo esquema da base de dados.

5.1.4.2.2 Parâmetros do algoritmo dos k vizinhos mais próximos

5.1.4.2.2.1 Determinação do valor de k adotado

Não há consenso no valor do k. Muitos autores sugerem que o valor ideal de k deve ser a raiz quadrada do número N de casos completos, arredondado superiormente ($k = \left| \sqrt{N} \right|$), entre eles, Jönsson e Wohlin (2004). Conforme Soares (2007), Cartwright, Shepperd e Song (2003) sugerem k = 1 ou 2, embora k = 1 torne o algoritmo sensível a *outliers*. MYRTVEIT et al (2000) e Huisman (2000) sugerem k = 1. Já BATISTA e MONARD (2001) e vários autores da Bioinformática adotam como ideal o valor de k = 10. Há afirmações, entre elas a do grupo de Troyanskaya (2001), que a escolha do k não interfere no desempenho do algoritmo. Afirmam, também, que a medida que o k cresce o método aproxima-se do valor produzido pela média da coluna (o que é bastante intuitivo).

Deste modo, optou-se por quatro possíveis valores de k: 1,3,5,10. Valores superiores a 10 foram testados inicialmente mas, como comprovado também na literatura, não apresentaram modificações no resultado final. (BATISTA e MONARD ,2003a) (SEHGAL et al 2005)(WONG et al 2007).

Uma escolha mais customizada do k poderia ser: a) utilizar a heurística do $k = \left| \sqrt{N} \right|$, (embora nos experimentos de SOARES (2007), esta heurística tenha falhado experimentalmente) considerando N o número de casos do grupo ou b) aplicação de um algoritmo genético, ou c) avaliar o sub-grupo escolhido em função de índices de avaliação de agrupamentos. Estas variações estão entre os objetivos futuros.

5.1.4.2.2.2 Desempenho nas bases dos valores adotados de k

Agrupando as instâncias de workflow, por percentual de ausência em cada uma das bases e selecionando o *k* com melhor desempenho para estas configurações (o com menor erro de imputação) obteve-se a seguinte distribuição:

- Base *Iris Plants*

	Percentual de Ausência		
k	10	20	30
1	11.66%	2.75%	16.22%
3	18.91%	22.43%	26.21%
5	35.49%	50.71%	20.24%
10	33.94%	24.11%	37.33%

Nesta base, o k= 10 só consegue vencer os demais quando o percentual de ausência é máximo. O comportamento do k para percentual de ausência de 20% difere dos demais e o k=5 se sobressai muito. Acredita-se que esta situação está relacionada à alta correlação da base, com valores muito próximos e principalmente à pequena quantidade de registros. Em índices altos de ausência, os possíveis doadores podem estar bastante incompletos e a similaridade é obtida em segmentos da tupla. Portanto, uma maior quantidade de doadores suaviza o impacto da semelhança parcial. No entanto, com percentuais de ausência mais baixos, existe mais abundância de doadores completos, mas devido à pequena quantidade de casos da base para obter-se 10 vizinhos é necessário um relaxamento no critério de similaridade, obrigando o método de imputação incorporar casos não tão similares no cálculo do valor estimado.

Base *Pima Indians*

	Percentual de Ausência		
k	10	20	30
1	0.17%	0.12%	0.26%
3	6.88%	7.09%	5.89%
5	18.32%	14.95%	19.82%
10	74.63%	77.84%	74.03%

Nesta base, o k= 10 é sem dúvida a escolha correta. Acredita-se que a razão desta supremacia deve-se às seguintes características da base: maior quantidade de registros, baixa correlação entre os atributos e a maior variabilidade no domínio de valores dos atributos.

- Base *Breast Cancer*

	Percentual de Ausência		
K	10	20	30
1	3.08%	2.98%	2.38%
3	15.96%	10.71%	12.66%
5	26.08%	19.26%	37.98%
10	54.88%	67.05%	46.98%

Nesta base, também, o k= 10 é a escolha correta. Embora aqui, os atributos sejam correlacionados e a base seja compacta, há registros suficientes para encontrar vizinhos de boa qualidade, melhorando o processo de imputação.

- Base Wine

	Percentual de Ausência		
k	10	20	30
1	0.81%	2.79%	7.24%
3	8.30%	12.19%	17.76%
5	20.07%	32.92%	36.40%
10	70.82%	52.09%	38.60%

Nesta base, também, o k= 10 é a escolha correta. Acredita-se que a quantidade de registros, 172, a correlação entre os atributos e a o domínio dos valores não ser muito amplo (mas com faixas significativamente maiores que a da *Íris*) impacta no desempenho do k=10, quando há 30% de ausência.

- Base *Hardware Machine*

	Percentual de Ausência		
k	10	20	30
1	47.41%	49.40%	25.96%
3	22.07%	15.53%	36.05%
5	19.28%	18.96%	24.58%
10	11.24%	16.11%	13.42%

O k nesta base tem um comportamento inverso das demais. Acredita-se que se deve ao fato explanado anteriormente, ou seja, há valores inconsistentes na faixa possível do domínio (por exemplo, tamanho de memória). Deste modo, ao calcular a média de k vizinhos, o valor estimado não é um valor válido sendo melhor replicar o valor de seu vizinho mais similar.

- *Geral*

K	Desempenho
1	11.55%
3	15.86%
5	26.23%
10	46.37%

Independente de qualquer parâmetro ou característica da base observa-se a predominância do k=10, em concordância com resultados relatados na literatura.

5.1.4.2.2.3 Tipo de distância

O algoritmo *k*-NN apóia-se no conceito de similaridade para construir a fila de objetos (os vizinhos) de onde são extraídos os candidatos para imputação. O conceito de similaridade, por sua vez, apóia-se na idéia de distância. A distância Euclidiana é a preferida na literatura, talvez por ter sido a originalmente proposta ou por sua simplicidade de cálculo. A distância Euclidiana, no entanto, é sensível a *outliers* e não captura adequadamente a variância e a correlação dos atributos (SALTON 1988). Há alguns trabalhos, entre eles os de WONG et al (2007), VERBOVEN et al (2007), HRUSCHKA, HRUSCHKA JR. e EBECKEN (2003b), que adotam outras medidas ou modificam a distância Euclidiana buscando por melhores resultados em seus trabalhos.

Deste modo, sobre a hipótese de que na imputação multivariada a correlação dos atributos deva ser considerada, optou-se pela escolha da distância Euclidiana e da distância de Mahalanobis que permite considerar correlações entre os atributos dos objetos Ao selecionar a distância de Mahalanobis pretende-se considerar correlações entre as características dos objetos, questão considerada de suma importância no processo de complementação.

5.1.4.2.2.4 Desempenho nas bases das distâncias escolhidas

Os resultados dos testes das duas distâncias escolhidas agrupados por base e considerando a melhor instância, isto é, a que apresenta o menor índice de erro de similaridade relativo, estão resumidos abaixo:

Iris	percentual de ausência		
Distância	10	20	30
Euclidiana	67.05%	67.05%	74.16%
Mahalanobis	32.95%	32.95%	25.84%

Pima	percentual de ausência		
Distância	10	20	30
Euclidiana	41.68%	66.69%	49.37%
Mahalanobis	58.32%	33.31%	50.63%

Breast	percentual de ausência		
Distância	10	20	30
Euclidiana	41.69%	45.04%	34.26%
Mahalanobis	58.31%	54.96%	65.74%

Machine	percentual de ausência		
Distância	10	20	30
Euclidiana	80.29%	87.91%	83.16%
Mahalanobis	19.71%	12.09%	16.84%

Wine	percentual de ausência		
Distância	10	20	30
Euclidiana	73.79%	80.22%	88.36%
Mahalanobis	26.21%	19.78%	11.64%

Distância	Percentual
Euclidiana	64.38%
Mahalanobis	35.62%

Os resultados dos testes contrariam a hipótese inicial, onde se acreditava que a distância de Mahalanobis traria grandes benefícios ao processo de complementação por incluir a correlação dos atributos no cálculo da medida. Exceto para a base *Breast* de ausência, seu desempenho foi bastante inferior ao da distância Euclidiana. É possível que o bom desempenho da distância Mahalanobis na base *Breast*, seja pela união dos seguintes fatores: a faixa de valores de todos os atributos é a mesma, oito de seus 9 atributos são correlacionados com um desvio–padrão significativo e há bastante casos. Portanto, a forma da distribuição dos casos é mais elipsoidal que as demais, favorecendo a distância de Mahalanobis. No caso da Pima, a pequena supremacia no índice de10%, necessita de um estudo bem mais criterioso, envolvendo a formação dos grupos. A hipótese inicial, mas que precisa ser avaliada com mais cuidado, é que a morfologia da ausência interna aos grupos esteja influenciando, pois há quatro atributos correlacionados 2 a 2.

5.1.4.2.2.5 Possíveis Vizinhos

Como muitas vezes o número de casos originalmente completos é pequeno, comprometendo a viabilidade e veracidade do processo de complementação, optou-se por considerar como possíveis vizinhos também casos incompletos de acordo com a definição de JÖNSSON e WOHLIN (2004).

5.1.4.3 A imputação de um Grupo

5.1.4.3.1 Parâmetros do algoritmo de agrupamento: rede SOM

No procedimento *hot-deck* (FORD, 1983) a imputação deve ser precedida por algum algoritmo de agrupamento. Todavia, apesar de ser uma solução já consolidada na área, a determinação da quantidade de grupos continua a ser um desafio e, dentro da tarefa de complementação de valores ausentes, postergada para segundo plano. Neste contexto, também, o algoritmo mais popular é o *K*_means onde a determinação do número de grupos é um parâmetro de entrada. Intuitivamente, o número máximo de grupos possíveis equivale ao número máximo de combinações de morfologias de ausência, ou seja, 2^k. No entanto, conforme frisado no capítulo 4, uma separação menos granular é desejada e que reflita as relações intrínsecas do domínio de entrada. Portanto, a escolha do método de agrupamento foi dirigida por estes dois aspectos, determinação automática da quantidade de grupos, e que estes preservem a formação topológica de ausência, optando-se pela rede SOM. Os agrupamentos mapeados pela rede SOM são automaticamente segmentados pelo algoritmo Costa-Netto (vide apêdice III)

5.1.4.3.1.1 A topologia da rede

O número de nós do mapa deve ser configurado a priori e influencia a resposta da rede. Normalmente esta escolha é empírica, embora haja uma corrente que utiliza a heurística do número de nós do mapa ser múltiplo de $\sqrt{N}$, onde N é o número de objetos da base (VESANTO e ALHONIEMI, 2000).

Com intuito de tentar encontrar heurísticas para a topologia da rede SOM, quando o propósito é um agrupamento grosseiro dos objetos, norteado pelo princípio de divisão e conquistas, e mantendo-se o mais independente possível do domínio as cinco topologias abaixo, propostas e justificadas no capítulo 4 foram experimentadas:

1. *Número de atributos* - o mapa é composto por 2k nós, onde k representa o número de atributos da base. Deste modo, há um protótipo no mapa para cada combinação possível de ausência.
2. *Número de tuplas* - um nó para cada caso existente na base.
3. *Número de células ausentes* - um nó para cada célula ausente.

4. *Número de tuplas com valores ausentes* - um nó por tupla que apresenta algum valor ausente.

5. *Média de atributos* - quantidade de atributos ausentes / quantidade de atributos da base nós.

5.1.4.3.1.2 Desempenho das topologias de redes adotadas

Os gráficos 5.1 a 5.5 abaixo demonstram o desempenho das topologias e são resultados da freqüência da colocação da topologia alvo em relação às demais, considerando o melhor resultado para uma mesma configuração nos demais parâmteros das instâncias de workflow. Portanto, a metodologia utilizada para medir o desempenho das arquiteturas foi: dada uma configuração de parâmetros, comparar os melhores resultados obtido por cada uma das arquiteturas, determinar a colocação, contabilizar a colocação.

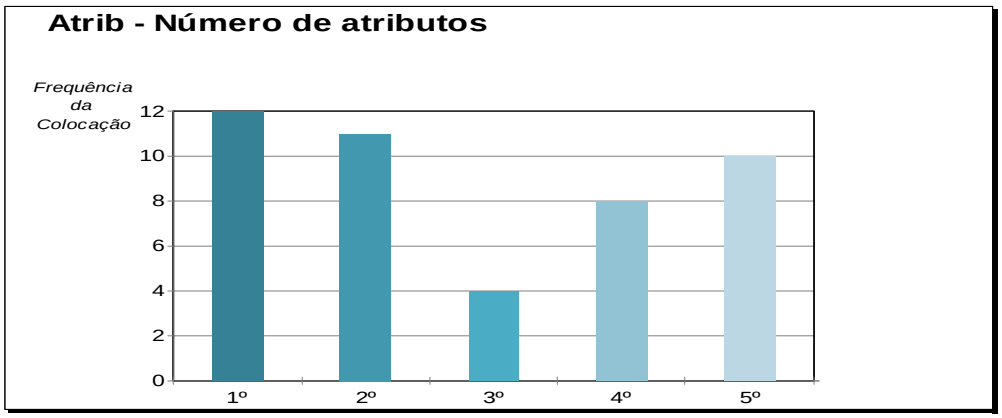


Gráfico 5.1 – Freqüência da colocação da topologia Número de Atributos

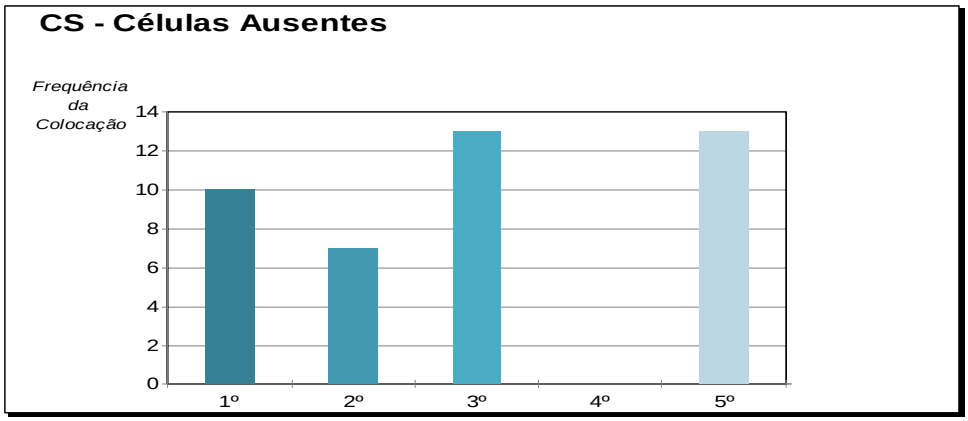


Gráfico 5.2 – Freqüência da colocação da topologia Células Ausentes

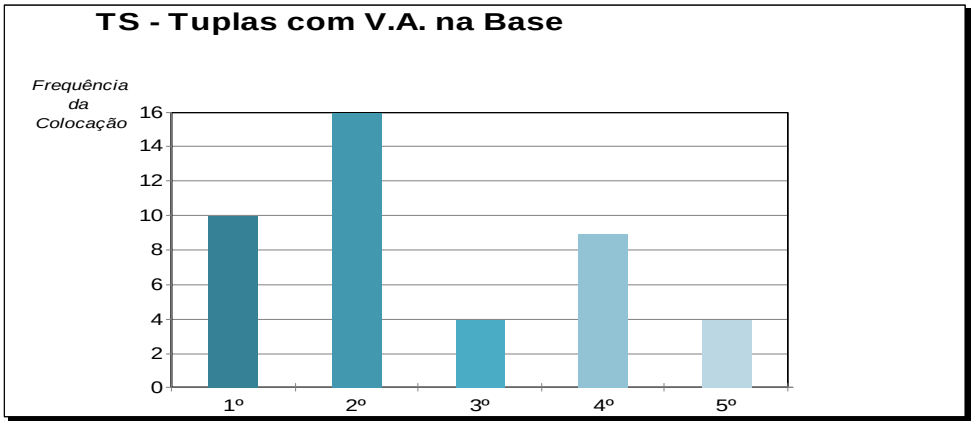


Gráfico 5.3 – Freqüência da colocação da topologia Número de Tuplas com valores Ausentes

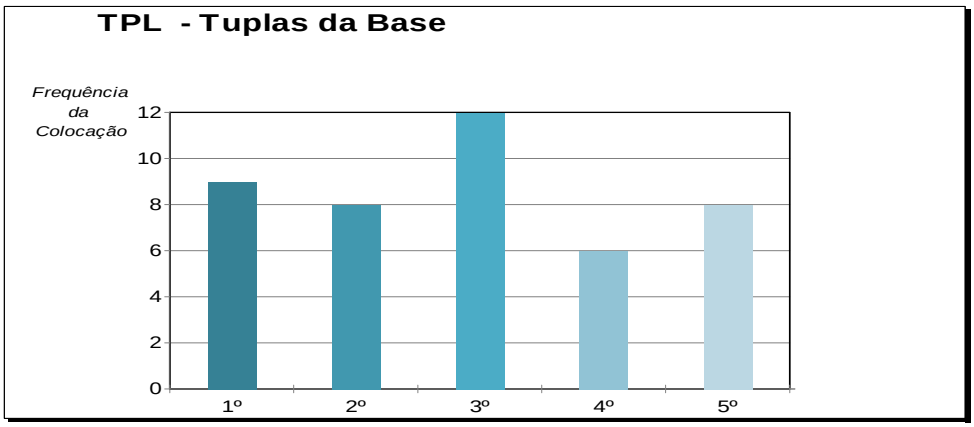


Gráfico 5.4 – Freqüência da colocação da topologia Número de Tuplas da Base

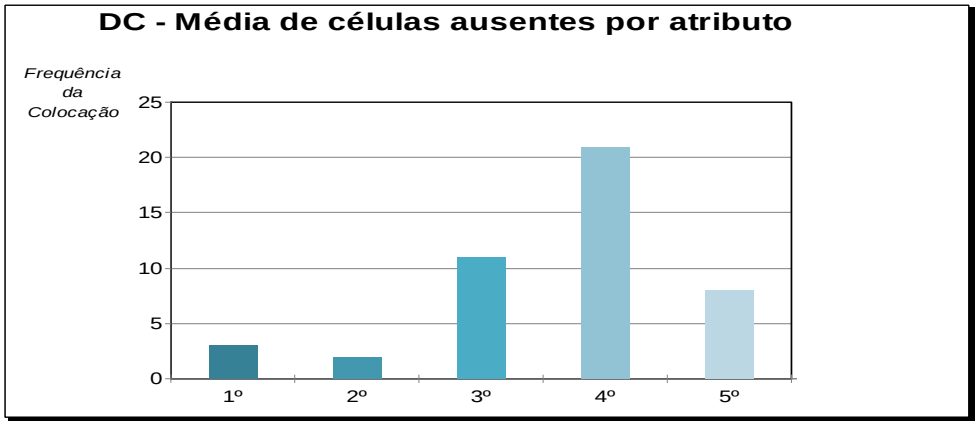


Gráfico 5.5 – Freqüência da colocação da topologia Média de Células Ausentes por Atributo

Considerando apenas a freqüência no primeiro lugar, as topologias podem ser ordenadas do seguinte modo: 1º) *Número de atributos* (12 vezes), *2º) Número de células ausentes e Número de tuplas com valores ausentes* (10 vezes), *4º) Número de tuplas na base* (9 vezes) *e 5º) Média de atributos* (3 vezes). No entanto, a variação entre os quatro primeiros colocados é tão pequena que é imperativo avaliá-las de forma mais

ampla,como por exemplo, a freqüência nos dois primeiros lugares e a estabilidade do desempenho. A estabilidade do desempenho é uma relaçao de recompensa (pelas boas colocações) – punição (pelas péssimas colocações) e foi medida porque acredita-se que é melhor ter uma topologia que sempre apresente um resultado mediano do que uma que apresenta, numa mesma freqüência, os melhores e piores resultados.

O gráfico 5.6 ordena as topologias considerando suas frequências nos dois primerios lugares. Pode-se observar uma mudança de posição no desempenho das topologias. A arquitetura *Número de tuplas com valores ausentes*, que estava empatada em segundo lugar, torna-se a vencedora com 26 colocações, seguida de *2º) Número de atributos* (23 vezes), *3º) Número de células ausentes e Número de tuplas da base* (17 vezes) e novamente em último lugar *a Média de atributos* (3 vezes).

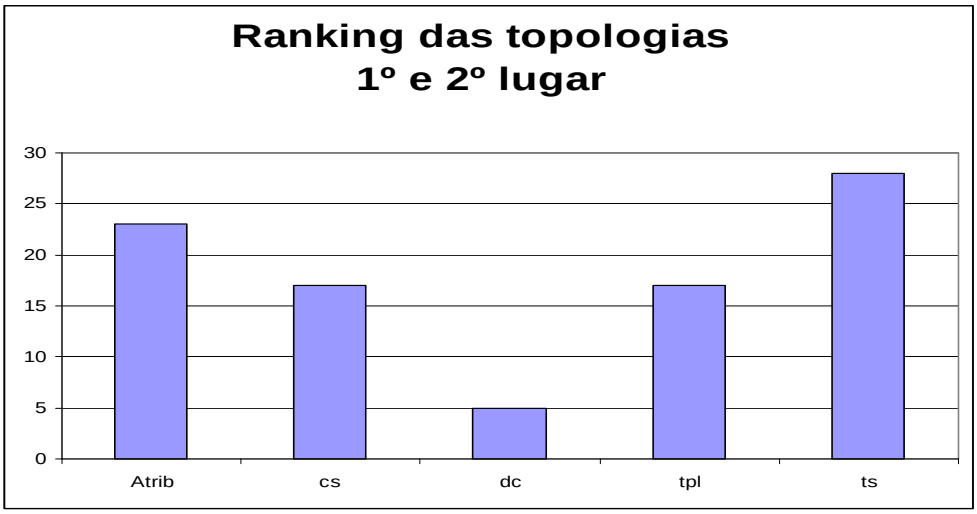


Gráfico 5.6 Ranking das topologias considerando a freqüência nos dois primeiros lugares

A supremacia da topologia *Número de tuplas com valores ausentes* fica mais evidenciada considerando a constância do desempenho e a topologia *Número de células ausentes* apresenta um resultado levemente superior ao da *Número de tuplas da base*, conforme pode ser observado nos gráficos 5.7 e 5.8.

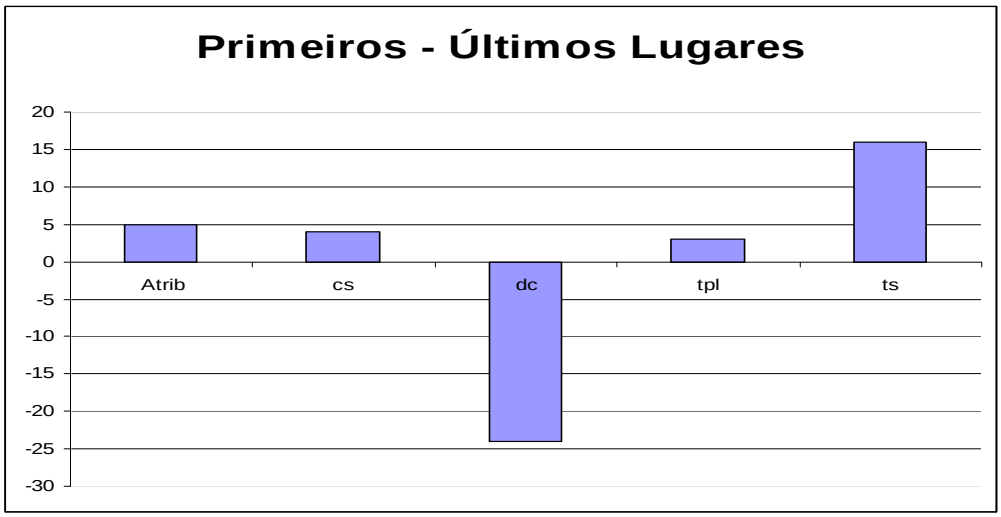


Gráfico 5.7 – Medida da constância do desempenho das topologias

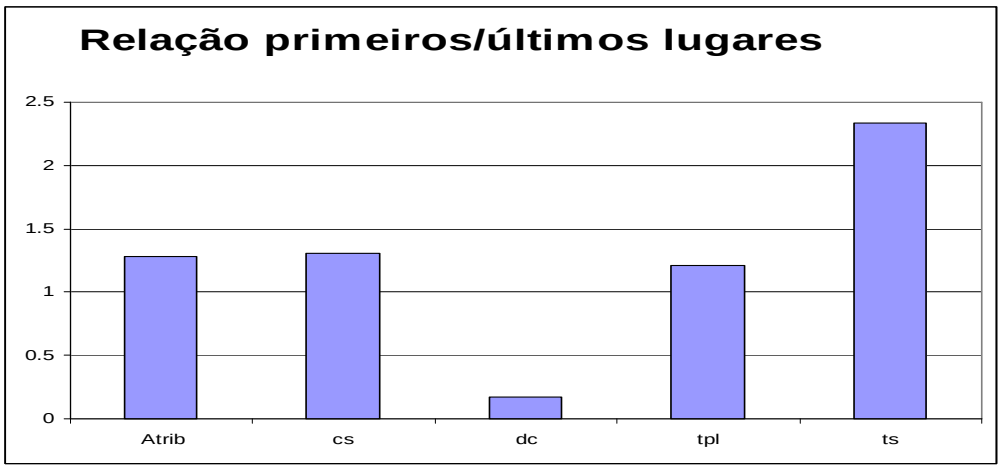


Gráfico 5.8- Ranking das topologias considerando a estabilidade do desempenho

Desta forma, pode-se concluir o seguinte ranking para as topologias: 1º) *Número de tuplas com valores ausentes, 2º) Número de atributos, 3º) Número de células ausentes, 4º) Número de tuplas na base e 5º) Média de atributos.* Vale salientar que em relação à estabilidade, as arquiteturas *Número de células ausentes* e *Número de tuplas na base* apresentam resultados quase idênticos.

5.1.4.3.1.3 Demais parâmetros da rede

Escolhidos por seu uso na literatura e por serem padrões de ferramentas disponíveis como o JOONE. O pacote JOONE – *Java Object Oriented Neural Engine* (MARRONE, 2007) é um *framework*, bastante difundido, desenvolvido em Java para construir e executar aplicações de Inteligência Artificial, baseadas em redes neuronais.

- *Número de ciclos:* O número de ciclos do treinamento também foi variado nos experimentos. Influencia a formação dos grupos e o tempo de

treinamento da rede. Foram escolhidos como valores de ciclo: 1000, 5.000, 10.000 e 15.000.

- *Tipo de aprendizado:* aprendizado em lote (vide capítulo 2)
- *Determinação de vizinhança:* de acordo com a distribuição Gaussiana (vide capítulo 2)
- *Inicialização dos protótipos:* aleatória
- *Taxa de aprendizado*: 0.7.

5.1.4.3.2 A ordenação dos segmentos

A imputação em cascata defende a idéia da reutilização dos valores. Os registros incompletos são inicialmente extraídos da base e agrupados para depois incrementalmente serem reincorporados. Portanto a determinação da ordem de imputação destes grupos é muito relevante. É necessário um equilíbrio entre a quantidade de registros disponíveis para a imputação e a "qualidade" destes registros. Almeja-se que os valores estimados sejam exatamente iguais (ou mais similar possível) ao original, mas dependendo do número de registros na base fonte (ou treinamento), a quantidade de possíveis "doadores" pode ser insuficiente para gerar um valor de qualidade. Portanto, há sete ordens possíveis cujas justificativas estão no capítulo 4 (foram denominadas como se apresentam no workflow desenvolvido):

1. *tupleLessMissing* – Inicia com o grupo que possui a menor quantidade de tuplas ausentes até o com maior quantidade de tuplas ausentes;
2. *tupleMoreMissing* – Inverso do anterior. Inicia com o grupo com maior quantidade de tuplas ausentes até o grupo com menor quantidade de tuplas ausentes;
3. *fieldLessMissing* - Inicia com o grupo com menos células ausentes até o com mais células ausentes;
4. *fieldMoreMissing* - Inverso do anterior. Inicia com o grupo com mais células ausentes até o grupo com menos células ausentes;
5. *fieldPerTupleLessMissing* - Do cluster com menor média de células ausentes por tupla até o grupo com maior média de células ausentes por tupla;

6. *fieldPerTupleMoreMissing* - Do cluster com maior média de células ausentes por tupla até o com menor média de células ausentes por tupla;
7. *noSort* – aleatória. Utiliza a ordem de geração dos grupos.

Neste trabalho somente aspectos quantitativos simples foram avaliados. Outros critérios, passíveis de serem utilizados foram descritos no capítulo 4 e são alvos de estudos futuros. Devido sua importância no contexto desta tese, uma análise do impacto da ordenação dos grupos e dos atributos dentro dos grupos, bem como as combinações possíveis é realizada na seção 5.2.3.

5.1.5 Métricas

5.1.5.1 Medida do erro do processo de imputação

As principais escolhas como medidas de erro na literatura são: a *medida do erro relativo absoluto* (RAD – *Relative Absolute Deviation*), a *medida de erro absoluto* e a *medida de erro de similaridade.*

O erro absoluto é calculado por:

$$E_{ABS} = |a\text{-}a'|$$

enquanto que o erro relativo absoluto é calculado da seguinte forma:

$$RAD_X(i) = \frac{\left|a - a'\right|}{a}$$

onde *a* é o valor original do atributo *X* da tupla *i* e *a'* é o valor imputado do atributo *X* nesta tupla *i*.

Por exemplo, considere o valor original = 2 e valor imputado = 3 e outro atributo *y*, com valor original = 60 e valor imputado = 80. O erro relativo absoluto e erro absoluto para *x e y é:*

$$RAD_x(i) = \frac{|2-3|}{2} = 50\% \quad \text{EABSx}(i) = |\,2-3\,| = 1$$

$$RAD_y(i) = \frac{|60-70|}{60} \approx 16{,}6\% \qquad \text{EABS}_y(i) = |\,60-70\,| = 10$$

Na avaliação do processo de imputação, caso o erro absoluto tivesse sido considerado, tem-se que o erro na imputação de *y* é 10 vezes superior ao de *x*, enquanto ao considerarmos o erro absoluto relativo, esta proporção cai para aproximadamente 1/3 pois está expressando o quão distante o dado imputado está do valor original diminuindo a interferência do domínio dos valores. No entanto, o erro relativo absoluto não pode ser calculado quando o valor mínimo do atributo é zero, valor válido, por exemplo, para o atributo número de gravidez da base *PIMA*.

MAGNANI e MONTESI (2004) em seu trabalho, tratado no capítulo 3, propuseram uma medida de similaridade para a tarefa de agrupamento. Esta métrica foi adaptada por SOARES (2007) como medida de erro. A função de similaridade proposta mede, para cada atributo de um registro, o quão parecidos são os registros e é expressa por:

$$sim(a_k, b_k) = 1 - \frac{|a_k - b_k|}{|\max_X(k) - \min_X(k)|}$$

Aqui, *a* e *b* são duas tuplas de uma tabela, *a*(*k*) indica o atributo *k* das tuplas *a* e *b*, $\max_X(k)$ e $\min_X(k)$ são respectivamente o maior e o menor valores apresentados na tabela para a coluna *k* para todas as tuplas *x* da tabela. É facilmente verificado que quando dois casos são idênticos a função $sim(a_k, b_k) = 1$, pois o valor da diferença em módulo $|a_k - b_k|$ é igual a zero. Logo os atributos a_k e b_k possuem o mesmo valor. Portanto, o segundo termo desta expressão ($\frac{|a_k - b_k|}{|\max_X(k) - \min_X(k)|}$) expressa o quão diferente são os atributos a_k e b_k , relativos aos valores extremos encontrados na coluna. Se considerarmos a_k o valor original e b_k o valor imputado, este termo pode ser considerado como uma possível medida de erro pois mede o quão longe o valor imputado está do valor original. Esta medida, chamada de **erro de similaridade** está expressa abaixo, e foi a medida escolhida.

$$E_{SIM} = \frac{|x - x'|}{|\max_x - \min_x|}$$

Para visualizar erros médios, é calculada a média aritmética simples, definindo assim o erro de similaridade médio de um processo de imputação em uma tabela. Este erro médio pode ser expresso da seguinte forma:

$$\overline{Esim} = \frac{1}{m}\sum_{i=1}^{m} Esim(i)$$

Este erro médio é calculado tanto para análise do processo de imputação de uma coluna (atributo), como para a análise de imputação de agrupamentos (erro médio total da imputação de todos os atributos deste agrupamento) como para demais análises avaliadas.

5.1.5.2 Medida de preservação da correlação entre os atributos

É intuitivo que a reutilização de valores no processo de complementação reforça algumas relações existentes entre os atributos originais, uma vez que os valores estimados são derivados dos valores existentes e a seguir considerados como originais (ou parcialmente se forem ponderados) para as próximas imputações. É um efeito cascata similar ao "juros sobre juros". Portanto, uma medida que se deseja analisar é o quanto a base imputada preserva a correlação original dos atributos.

O coeficiente de correlação entre dois atributos A e B é dado por:

$$\rho_{A,B} = \frac{cov(A,B)}{\sigma_A \sigma_B}$$

onde cov(A,B) é a matriz de covariância entre os atributos A e B e σ_A é o desvio padrão do atributo A e σ_B do atributo B. Este coeficiente varia no intervalo de [-1..1], onde -1 significa a máxima correlação inversa dos atributos, 0 significa que não há correlação entre os atributos e 1 uma perfeita correlação direta. Por exemplo, considere os seguintes atributos em relação a alunos em uma disciplina prática: Número de Faltas, Horas de Prática Extra Classe, Cor do Cabelo e Rendimento, pode-se dizer que o Rendimento é inversamente correlacionado ao Número de Faltas, diretamente correlacionado ao número de Horas de Prática Extra Classe e não correlacionado com a Cor do Cabelo.

A matriz de correlação de uma base de dados é uma matriz quadrada simétrica, NxN, onde N representa o número de atributos, onde cada célula M[*i*,*j*] armazena a correlação do atributo K_i com o atributo K_j, ou seja:

$$\begin{pmatrix} \rho(K_1,K_1) & \rho(K_1,K_2) & \cdots & \rho(K_1,K_N) \\ \rho(K_2,K_1) & \rho(K_2,K_2) & \cdots & \rho(K_1,K_N) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(K_N,K_1) & \rho(K_N,K_2) & \cdots & \rho(K_N,K_N) \end{pmatrix}$$

A correlação total de um atributo K é definida como a correlação média deste atributo em relação aos demais atributos Por exemplo, a correlação total do atributo K1 é:

$$OC(K_1) = \frac{\rho(K_1,K_2) + \rho(K_1,K_3) + \cdots + \rho(K1,K_N)}{N-1}$$

Uma das metas é comparar a modificação da correlação do atributo imputado em relação ao atributo original. Esta medida, obtida pela diferença da correlação total do atributo na base imputada e na base original, representa o quão enviesada está a correlação deste atributo na base imputada e por isso foi denominada Bias da Correlação do Atributo (ACB) e é definida por:

$$ACB(k_{original}, k_{imputado}) = OC(k_{imputado}) - OC(k_{original})$$

Como medida global do impacto do processo de complementação de valores ausentes reutilizando os valores imputados na correlação da base, totaliza-se os bias dos atributos. Deste modo, esta métrica, denominada Bias da Correlação, é definida por:

$$CB = \sum_{i=1}^{N} ACB(k_{i\,imputado}, k_{i\,original})$$

5.2 Condições ambientais dos experimentos

Os testes foram realizados em um microcomputador com processador *Pentium Celeron D* 3 GHz, com 512 MB de memória principal, e 74,4 GB de disco rígido IDE, com o sistema operacional *Windows XP Professional* versão 2002 *Service Pack* 2 instalado. O *JBWorkflow, Eraser e o Analysis* foram desenvolvidos utilizando linguagem Java no ambiente Eclipse SDK versão 3.3.1.1 e JDK 1.5.0.11. Os dados estão armazenados em um Sistema Gerenciador de Banco de Dados MySQL versão 5.0.45-win32.

5.2.1 Estatísticas dos experimentos

Os seguintes fatores são responsáveis pela explosão combinatória resultando em gerou 252.000 experimentos (instâncias de workflows):

- *Número de Bases Utilizadas*

Ao total foram avaliadas 45 bases, porque para cada uma das cinco bases escolhidas (*Iris Plants*, *Pima Indians Diabetes* e *Wisconsin Breast Cancer*, *Wine e Computer Hardware Data Set)* testou-se com três percentuais distintos de ausência: 10%, 20% e 30%. Para cada um destes percentuais, três versões distintas foram criadas.(vide seção 5.1.3)

- *Número de redes SOM na tarefa de agrupamento*

Os experimentos foram realizados sobre 20 agrupamentos distintos gerados pela variação dos parâmetros da rede SOM. Para cada uma das cinco topologias propostas, quatro ciclos de treinamento foram avaliados. (vide seção 5.1.4.3)

- *Número de configurações resultantes do algoritmo k_NN na imputação de um atributo*

Os parâmetros do algoritmo k_NN, utilizado na imputação de uma célula e detalhados na seção 5.1.4.2.2 , resultaram em 8 combinações possíveis, pois para cada valor escolhido de k (1,3,5,10), duas distâncias foram testadas (distância Euclidiana e de Mahalanobis).

- *Número de ordenações possíveis na imputação*

Há sete ordens distintas para organizar os grupos gerados pela tarefa de agrupamento e dentro de cada um destes grupos, há cinco ordens para a seleção do próximo atributo a imputar, totalizado 35 ordens diferentes a serem testadas.(vide seção 5.1.4.2.1 e 5.1.4.3.2)

Resumindo o descrito acima, têm-se 45*20*8*35 = 252.000 distribuídos em:

1. Bases: 45
    a. Número de bases: 5 bases
    b. Número de versões: 3 versões
    c. Percentual de ausência 3 percentuais

2. Agrupamentos gerados pela rede SOM: 20
    a. Topologias: 5 topologias
    b. Ciclos: 4 ciclos
3. Método *k*-NN: 8
    a. Número de k: 4 valores
    b. Distâncias: 2 distâncias
4. Ordenações: 35
    a. Ordenações de segmentos: 7 ordens entre grupos (externa)
    b. Ordenações de atributos: 5 ordens de atributos (interna)

Todas as instâncias foram testadas e seus resultados persistidos em disco em arquivos formato XML. Os resultados dos testes alimentaram uma base de dados para análise.

O tempo de CPU utilizado pelos experimentos totalizou 69 dias 2 horas e 30 minutos distribuídos conforme tabela abaixo:

		Íris		Pima		Breast		Machine		Wine	
		min	hs	min	hs	min	hs	min	hs	min	hs
10	1	120	2.0	1543	25.72	5458	90.97	1069	17.82	834	13.90
	2	149	2.5	1712	28.53	5429	90.48	519	8.65	776	12.93
	3	100	1.7	1588	26.47	5655	94.25	651	10.85	797	13.28
		369	6.2	4843	80.72	16542	275.70	2239	37.32	2407	40.12
20	1	210	3.5	2057	34.28	6767	112.78	645	10.75	1028	17.13
	2	190	3.2	2308	38.47	6496	108.27	770	12.83	979	16.32
	3	190	3.2	2283	38.05	6254	104.23	759	12.65	1026	17.10
		590	9.8	6648	110.80	19517	325.28	2174	36.23	3033	50.55
30	1	292	4.9	2262	37.70	9819	163.65	863	14.38	1289	21.48
	2	282	4.7	2426	40.43	8216	136.93	1029	17.15	1329	22.15
	3	291	4.9	2468	41.13	8250	137.50	1089	18.15	1243	20.72
		865	4.9	7156	119.27	26285	438.08	2981	49.68	3861	64.35
		1824	20.9	18647	310.8	62344	1039.1	7394	123.2	9301	155.0
		1 dia 6hs 24min		12 dias 22hs 47m		43 dias 7hs 4m		5 dias 3hs14m		6 dias 11hs 1m	

5.3 Resultados da Imputação em Cascata

5.3.1 Comparação do desempenho da Imputação em Cascata

Os gráficos desta seção comparam o desempenho de duas abordagens similares: imputação em cascata e a imputação seqüencial. Para cada abordagem compara-se, também, a reutilização ou não dos valores imputados.

A imputação em cascata possui um passo de agrupamento pela morfologia da ausência precedendo a imputação seqüencial dos atributos. Como reutiliza os valores estimados, determina um seqüenciamento dos grupos e, internamente aos grupos, dos atributos, originando as 35 combinações descritas no item anterior.

A imputação seqüencial com realimentação de valores, também, reutiliza os valores estimados e ordena os atributos antes de preenchê-los. Esta abordagem faz parte da dissertação de mestrado de Rafael Castaneda, a ser defendida, e que utiliza duas ordenações de atributos: crescente e decrescente em relação à quantidade de células ausentes na base e usa a distância Euclidiana no algoritmo k-NN (CASTANEDA et al, 2008).

A imputação seqüencial sem realimentação transforma, também, o problema da imputação multivariada de *n* atributos em *n* imputações univariadas. A principal diferença é que todo valor estimado utiliza exclusivamente os valores originais da base.

O erro E_{SIM} foi adotado e a comparação levou em consideração as bases e os percentuais de ausência.

Na imputação em cascata para cada base/percentual de ausência há 16800 experimentos, portanto a primeira análise considera o menor erro gerado de duas formas: selecionando o menor erro todas as instâncias de workflows e o menor erro gerado pelas mesmas configurações.

A necessidade desta segunda medida deve-se ao fato que nos experimentos da imputação em cascata há diversos parâmetros adicionais sendo testados, como por exemplo, outros critérios de ordenação, de distância de topologias SOM para o agrupamento que podem melhorar ou piorar o desempenho da abordagem. Sendo assim, é mostrado, também, o comportamento do método para as mesmas condições da imputação seqüencial: dois critérios de ordenação interna e distância Euclidiana.

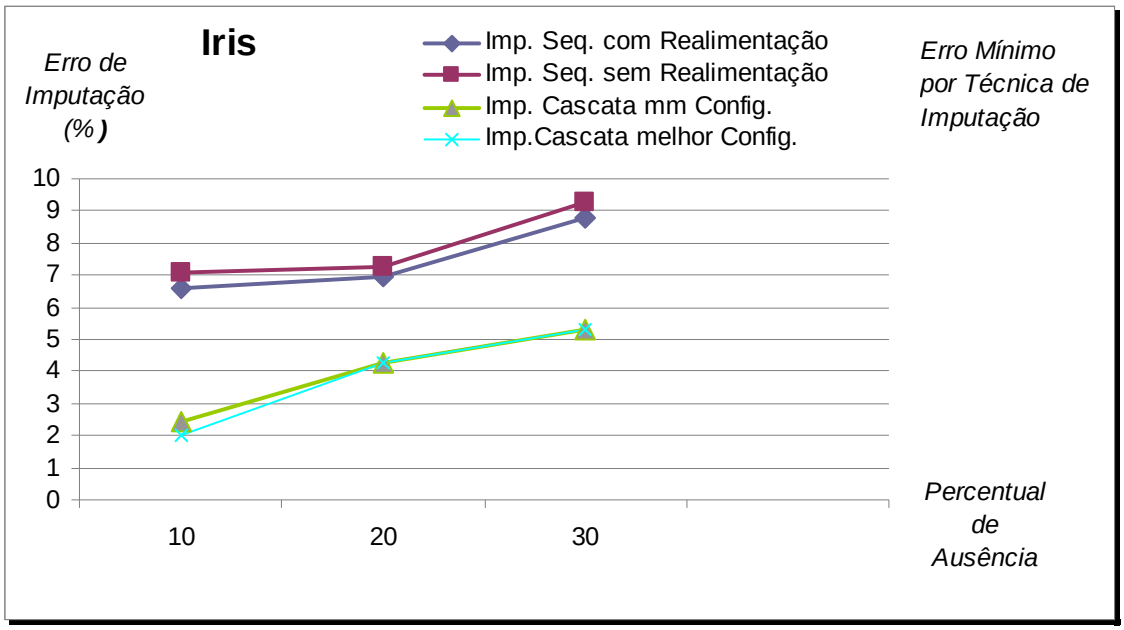


Gráfico 5.9- Erro mínimo nos métodos Imputação Seqüencial sem reutilização dos valores, Imputação Seqüencial com reutilização dos valores, Imputação em Cascata com reutilização de valores nas mesmas configurações e na melhor configuração da Imputação em Cascata na Base Íris.

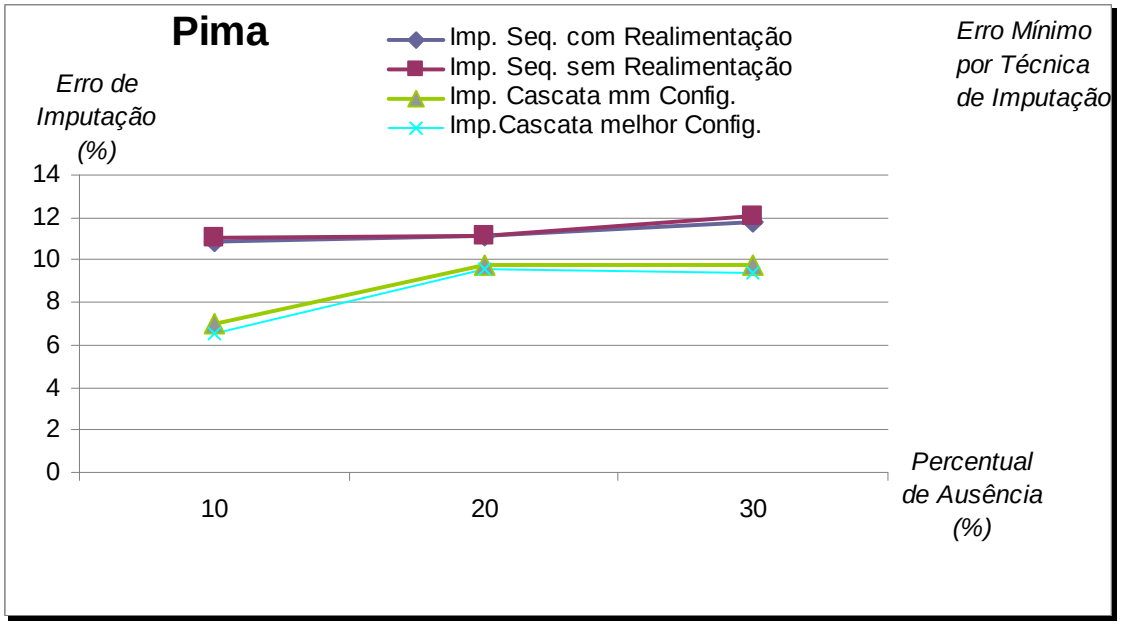


Gráfico 5.10- Erro mínimo nos métodos Imputação Seqüencial sem reutilização dos valores, Imputação Seqüencial com reutilização dos valores, Imputação em Cascata com reutilização de valores nas mesmas configurações e na melhor configuração da Imputação em Cascata na Base Pima.

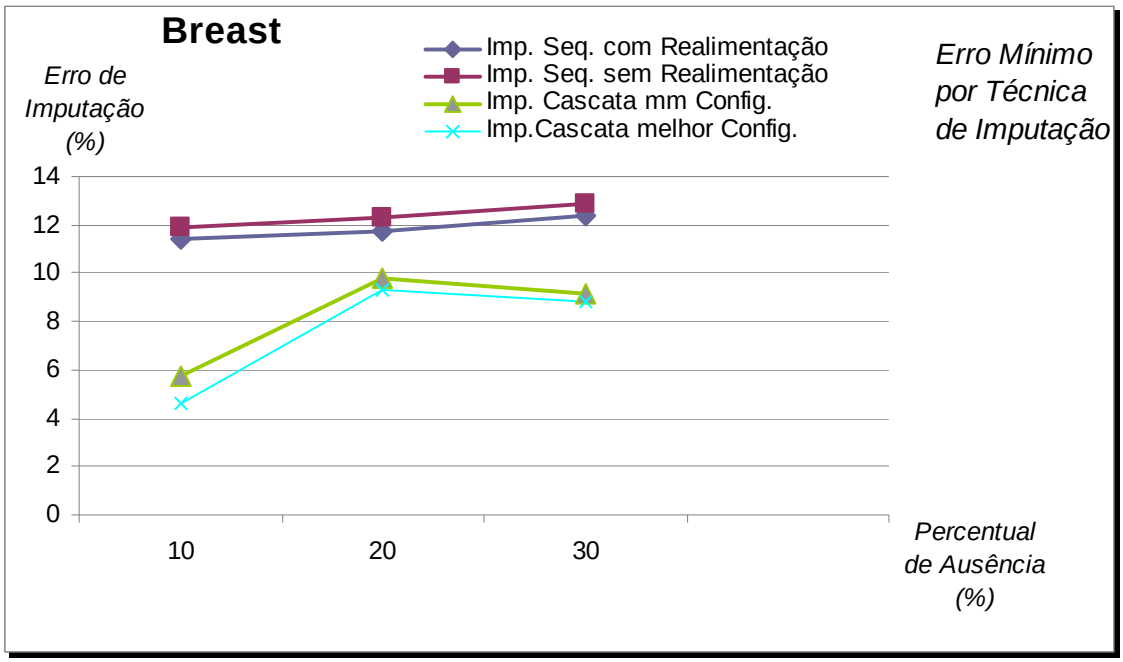


Gráfico 5.11- Erro mínimo nos métodos Imputação Seqüencial sem reutilização dos valores, Imputação Seqüencial com reutilização dos valores, Imputação em Cascata com reutilização de valores nas mesmas configurações e na melhor configuração da Imputação em Cascata na Base Breast.

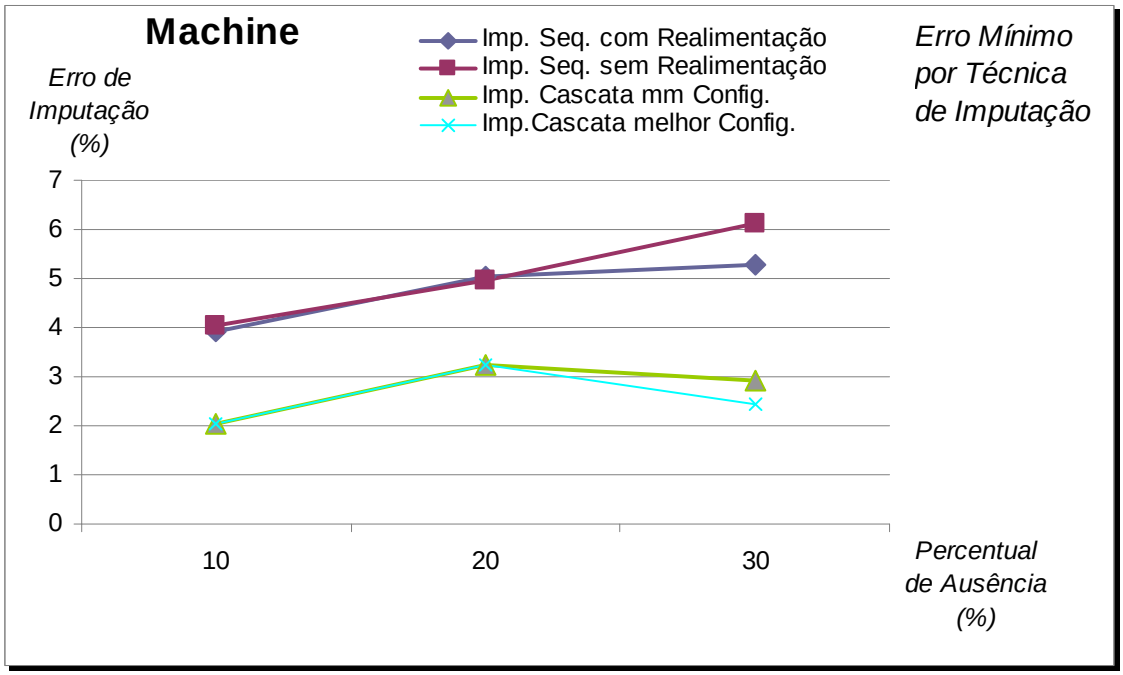


Gráfico 5.12- Erro mínimo nos métodos Imputação Seqüencial sem reutilização dos valores, Imputação Seqüencial com reutilização dos valores, Imputação em Cascata com reutilização de valores nas mesmas configurações e na melhor configuração da Imputação em Cascata na Base Computer Hardware (Machine)

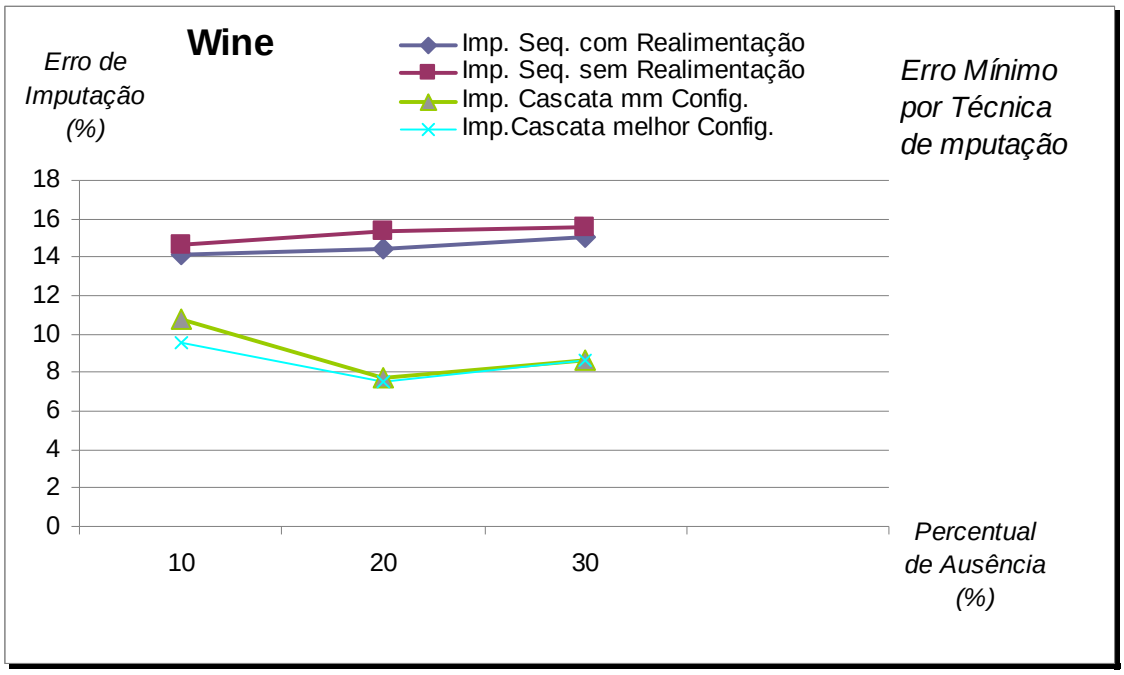


Gráfico 5.13- Erro mínimo nos métodos Imputação Seqüencial sem reutilização dos valores, Imputação Seqüencial com reutilização dos valores, Imputação em Cascata com reutilização de valores nas mesmas configurações e na melhor configuração da Imputação em Cascata na Base Wine

5.3.2 Análise e resultados da comparação do desempenho da Imputação em Cascata com demais métodos

Em relação à configuração com menor erro, a imputação em Cascata teve melhor desempenho em todas as bases. Os índices de erro diminuíram em no mínimo dois pontos percentuais em todas as situações.

Considerando as mesmas configurações e com percentual de ausência de 10%, o índice de erro diminui em média 40% em relação à imputação seqüencial com realimentação e 42% em relação à imputação seqüencial sem realimentação. Na medida em que o percentual de ausência cresce, esta relação diminui. Para índices de ausência de 20% a redução na taxa de erro é de 29% para seqüencial com realimentação e 31% para seqüencial enquanto para 30% de células ausentes é de 32% para seqüencial com realimentação e 35% para seqüencial. Um fato curioso nesta análise é a menor diferença no percentual de ausência de 20%. Acredita-se que é conseqüência de agrupamentos menos criteriosos, devendo ser melhor investigado nos trabalhos futuros. A queda desta relação justifica-se pela menor quantidade de registros para as primeiras imputações.

Nas duas bases com atributos correlacionados, *Íris Plants e Wisconsin Breast Câncer e Hardware Machine* a imputação em Cascata obteve o melhor desempenho em relação às demais abordagens diminuíndo o erro a 1/3 na *Íris* e à metade na *Breast* e a

aproximadamente à metade na *Machine*. Este contexto sugere que esta abordagem é uma boa alternativa para bases correlacionadas.

Este bom desempenho (redução de 50% na taxa de erro) também se repete para os índices de 20% e 30% de ausência na Base *Wine*, embora seja de apenas 30% para o a taxa de ausência 10% (abaixo da média). Como há quatro atributos fortemente correlacionados nesta base, acredita-se que esta é a razão do bom desempenho para os percentuais maiores de ausência.

Cabe comentar que observando os erros de todas as configurações da Imputação em Cascata e calculando uma média destes erros, a situação embora favorável à imputação em Cascata, não indica o mesmo comportamento. As diferenças são bem menores, sugerindo uma sensibilidade do método aos agrupamentos. Outro fator que influencia o cálculo da média é a quantidade de experimentos. Enquanto nos demais métodos há em torno de 60 experimentos participando da média, na imputação em cascata eles superam 2000.

5.3.3 Bias da correlação na Imputação em cascata

Os gráficos desta seção comparam o quanto a Imputação em Cascata modificou a correlação original dos atributos utilizando como métrica a medida definida no item 5.1.5.2, e chamada de Bias da Correlação. A alteração da correlação original da base é uma conseqüência possível da reutilização dos valores. A linha “zero” do gráfico representa a correlação original das bases, antes de sujá-las, ou seja, com todos os valores preenchidos. Portanto, nos demonstrativos abaixo, o ideal é estar sobre a linha zero, o que significa que não houve alteração na correlação original. Para efeitos de comparação são apresentados, também, os bias dos métodos de imputação seqüencial com e sem reutilização dos valores.

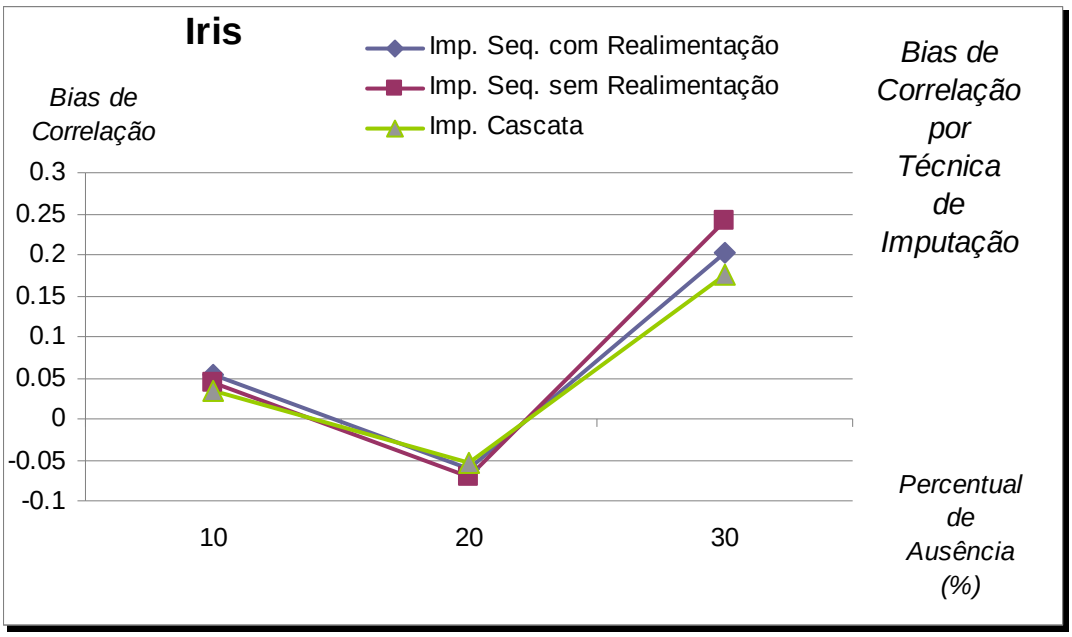


Gráfico 5.14- Tendência da alteração na correlação original dos atributos pela aplicação dos métodos Imputação Seqüencial com e sem reutilização dos valores e Imputação em Cascata com reutilização de valores nas mesmas configurações na Base Íris.

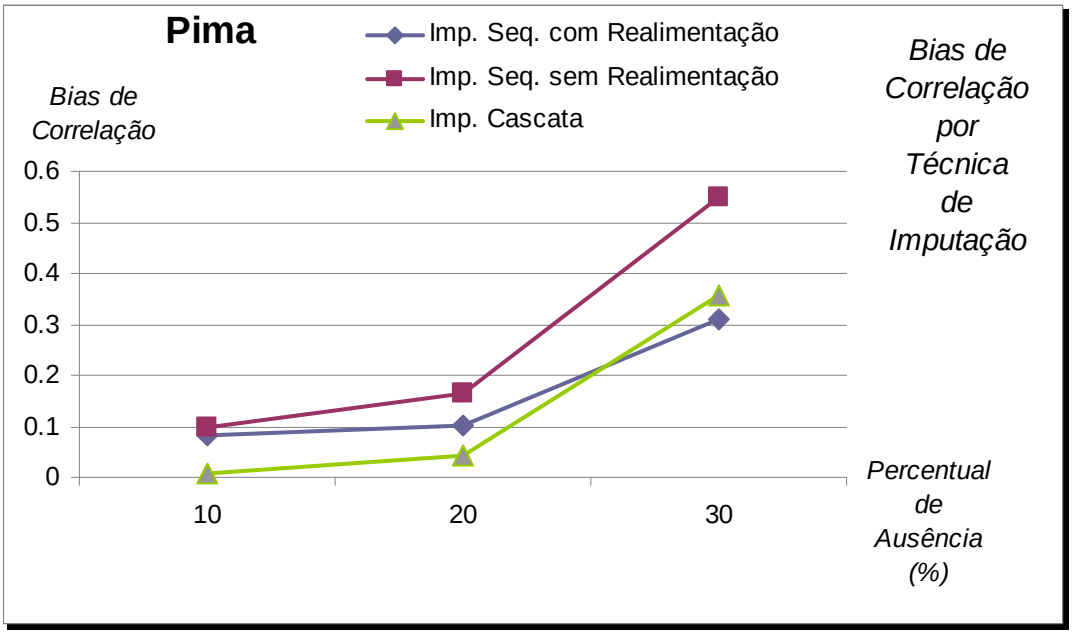


Gráfico 5.15- Tendência da alteração na correlação original dos atributos pela aplicação dos métodos Imputação Seqüencial com e sem reutilização dos valores e Imputação em Cascata com reutilização de valores nas mesmas configurações na Base Pima.

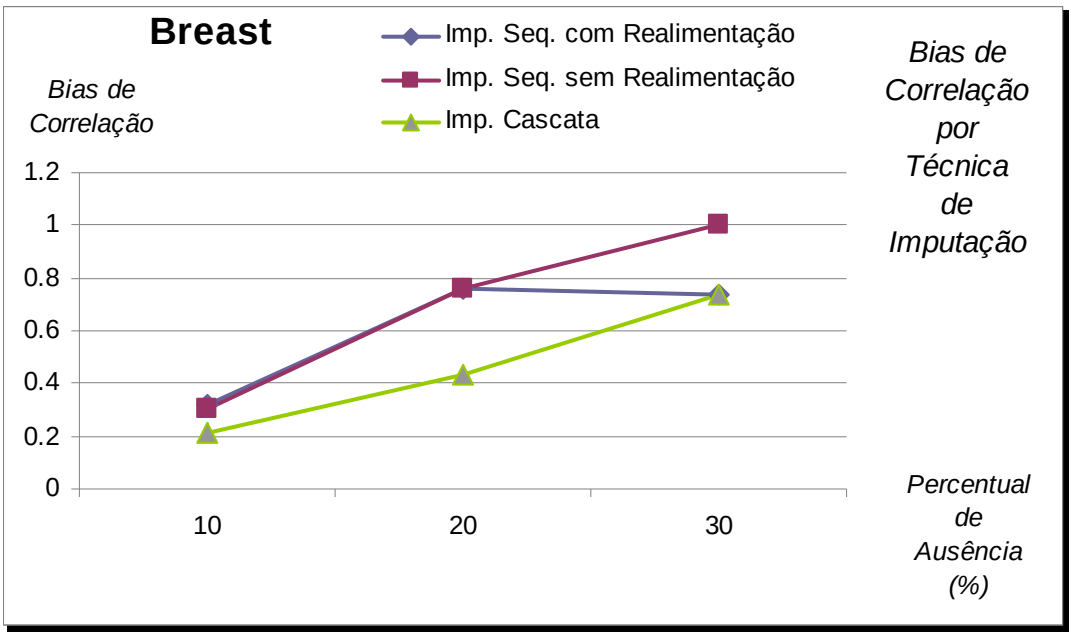


Gráfico 5.16- Tendência da alteração na correlação original dos atributos pela aplicação dos métodos Imputação Seqüencial com e sem reutilização dos valores e Imputação em Cascata com reutilização de valores nas mesmas configurações na Base Breast.

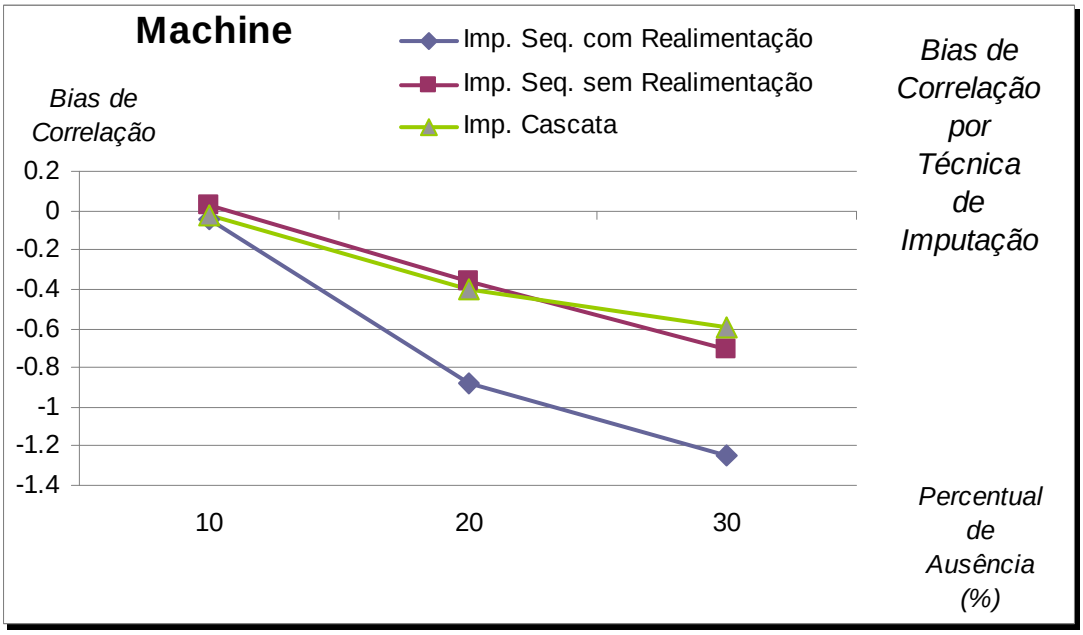


Gráfico 5.17- Tendência da alteração na correlação original dos atributos pela aplicação dos métodos Imputação Seqüencial com e sem reutilização dos valores e Imputação em Cascata com reutilização de valores nas mesmas configurações na Base Computer Hardware (Machine)

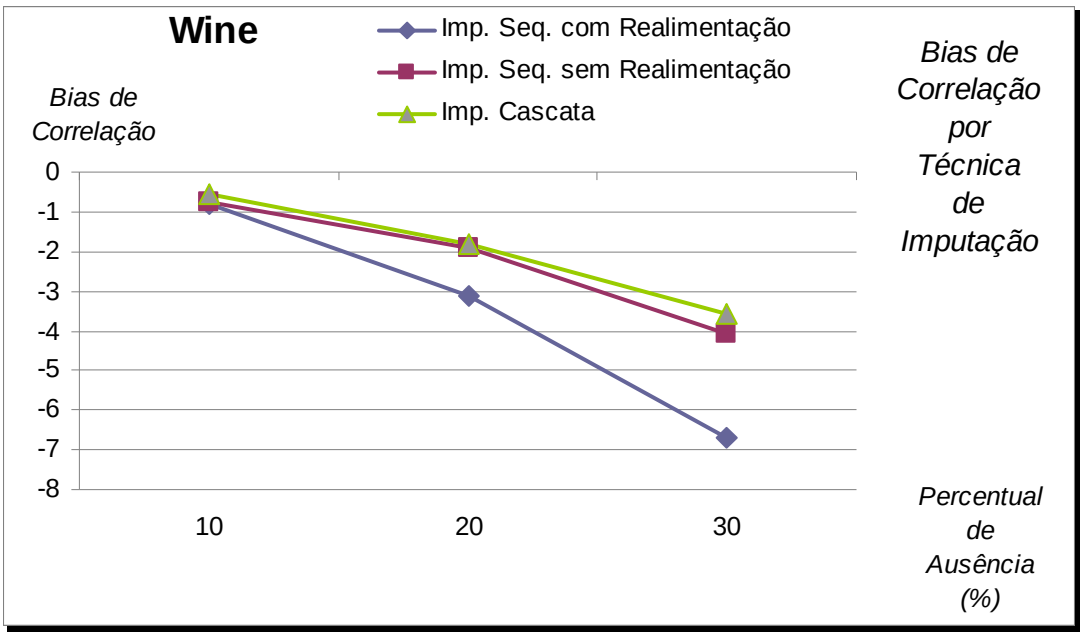


Gráfico 5.18- Tendência da alteração na correlação original dos atributos pela aplicação dos métodos Imputação Seqüencial com e sem reutilização dos valores e Imputação em Cascata com reutilização de valores nas mesmas configurações na Base Wine

5.3.4 Análise do bias da correlação na Imputação em cascata

Como pode ser observado, a imputação em Cascata foi a técnica que melhor manteve a correlação original em todas as bases. A imputação seqüencial com realimentação é a que mais interfere nesta relação. É bom salientar que no caso da *Íris Plants* -30%, na *Pima* – 10%,20% e 30%, e *Breast* - 30% a imputação seqüencial, sem realimentação alterou mais a correlação original do que a imputação seqüencial com realimentação. Além disso, modificou a correlação de todas as bases, sugerindo que a modificação da correlação é uma conseqüência da imputação e não apenas da realimentação dos valores.

Outra conclusão importante que se deduz destes gráficos é que na medida em que o índice de ausência aumenta, também aumenta a alteração da correlação mas mantendo a tendência positiva ou negativa. A Base *Íris*, tem um comportamento um pouco diferente pois com 20% a tendência da alteração da correlação muda de positiva para negativa, para em 30% voltar à tendência positiva. Acredita-se que a razão desta quebra na tendência está relacionada à construção dos grupos e com a escolha dos

doadores pelo algoritmo k_NN, pois na análise dos mesmos, a Base *Íris* com 20% de ausência têm um comportamento diferenciado.

5.3.5 Comparação do Desempenho das Ordenações nas Bases

Em duas fases distintas da Imputação em Cascata é necessário decidir a ordem de imputação. Em primeiro lugar, os grupos devem ser ordenados. Esta ordenação é denominada a Ordem Externa. Depois, dentro do grupo alvo, é escolhida a ordem em que os atributos serão regredidos, chamada de Ordem Interna. Há sete ordens externas possíveis e cinco ordens internas, totalizando 35 configurações diferentes. A tabela abaixo resume e cria uma legenda para cada umas das combinações. Estas legendas são utilizadas nos gráficos 5.19 a 5.63 que se seguem. Os gráficos, por sua vez, representam o menor erro mínimo que o par (Ordem Externa, Ordem Interna) obteve nos experimentos em cada uma das bases.

Configuração	Ordenação Externa	Ordenação Interna
a-a	fieldLessMissing	lessCorrelation
a-b	fieldLessMissing	lessMissing
a-c	fieldLessMissing	moreCorrelation
a-d	fieldLessMissing	moreMissing
a-e	fieldLessMissing	noSort
b-a	fieldMoreMissing	lessCorrelation
b-b	fieldMoreMissing	lessMissing
b-c	fieldMoreMissing	moreCorrelation
b-d	fieldMoreMissing	moreMissing
b-e	fieldMoreMissing	noSort
c-a	fieldPerTupleLessMissing	lessCorrelation
Configuração	Ordenação Externa	Ordenação Interna
c-b	fieldPerTupleLessMissing	lessMissing
c-c	fieldPerTupleLessMissing	moreCorrelation
c-d	fieldPerTupleLessMissing	moreMissing
c-e	fieldPerTupleLessMissing	noSort
d-a	fieldPerTupleMoreMissing	lessCorrelation
d-b	fieldPerTupleMoreMissing	lessMissing
d-c	fieldPerTupleMoreMissing	moreCorrelation

d-d	fieldPerTupleMoreMissing	moreMissing
d-e	fieldPerTupleMoreMissing	noSort
e-a	noSort	lessCorrelation
e-b	noSort	lessMissing
e-c	noSort	moreCorrelation
e-d	noSort	moreMissing
e-e	noSort	noSort
f-a	tupleLessMissing	lessCorrelation
f-b	tupleLessMissing	lessMissing
f-c	tupleLessMissing	moreCorrelation
f-d	tupleLessMissing	moreMissing
f-e	tupleLessMissing	noSort
g-a	tupleMoreMissing	lessCorrelation
g-b	tupleMoreMissing	lessMissing
g-c	tupleMoreMissing	moreCorrelation
g-d	tupleMoreMissing	moreMissing
g-e	tupleMoreMissing	noSort

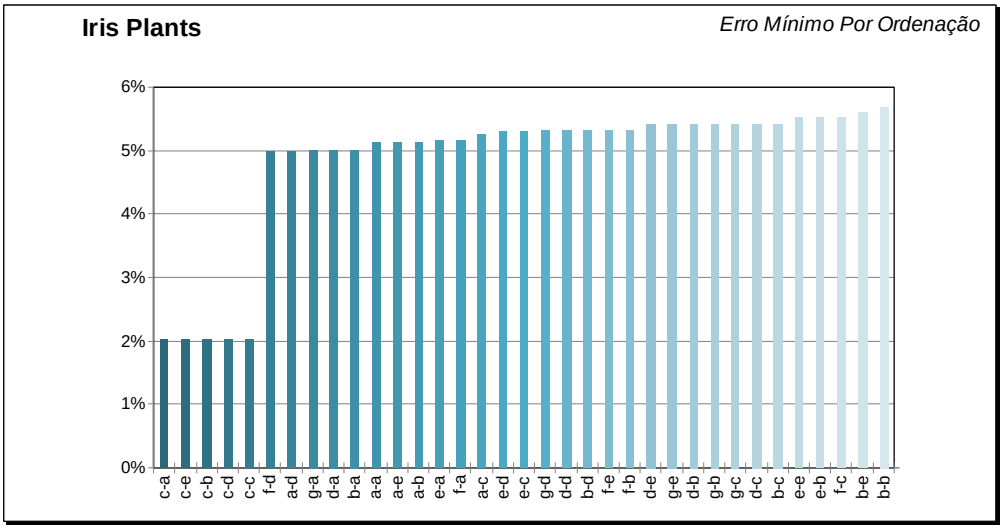


Gráfico 5.19- Erro mínimo das ordenações representadas pelo par (Ordem Externa, Ordem Interna) na Base Íris.

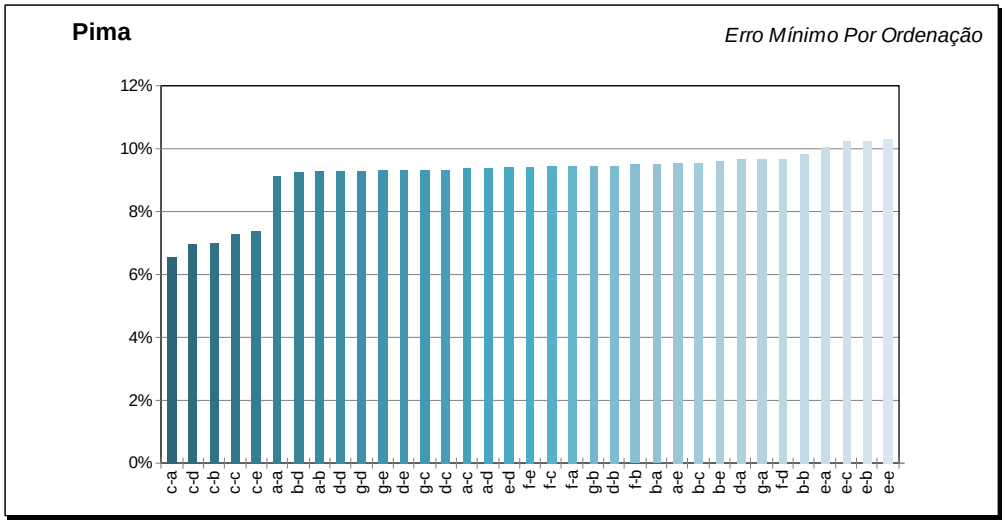


Gráfico 5.20- Erro mínimo das ordenações representadas pelo par (Ordem Externa, Ordem Interna) na Base Pima.

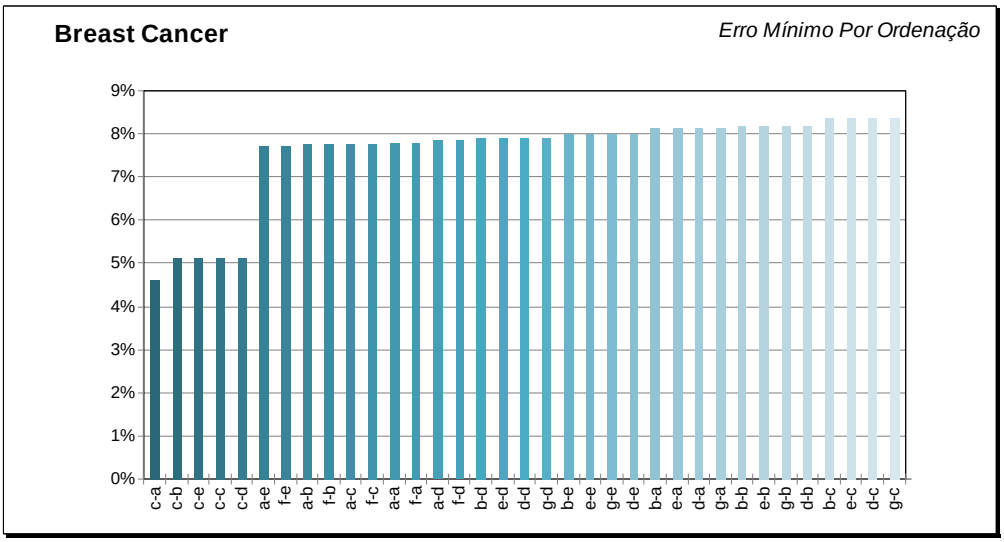


Gráfico 5.21- Erro mínimo das ordenações representadas pelo par (Ordem Externa, Ordem Interna) na Base Breast

.

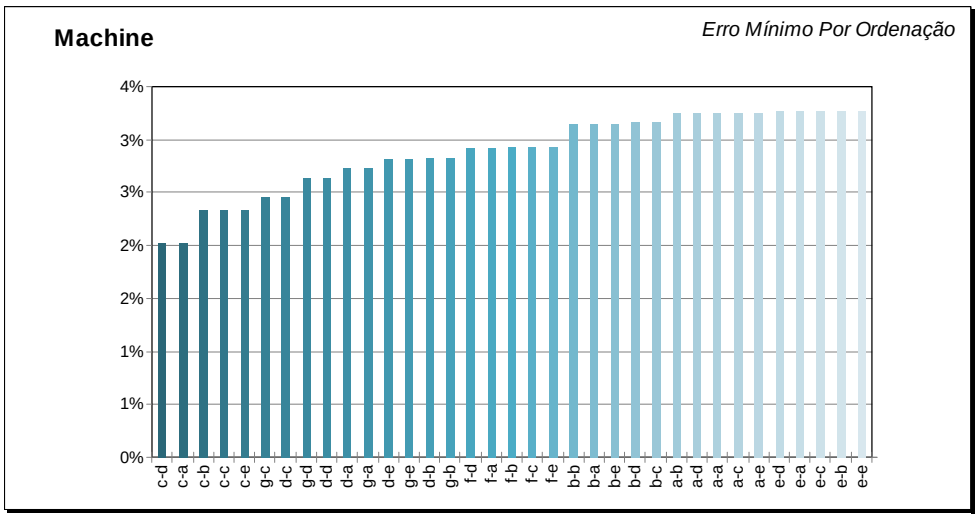


Gráfico 5.22- Erro mínimo das ordenações representadas pelo par (Ordem Externa, Ordem Interna) na Base Computer Hardware (Machine)

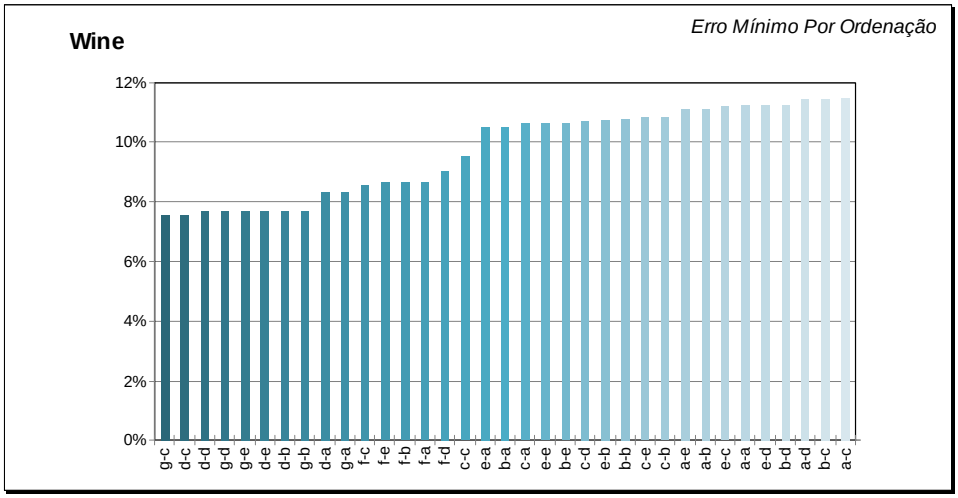


Gráfico 5.23- Erro mínimo das ordenações representadas pelo par (Ordem Externa, Ordem Interna) na Base Wine

5.3.6 Análise e resultados da comparação do desempenho das ordenações nos experimentos realizados

Exceto na base *Wine*, em todas as outras bases a ordem externa *FieldperTupleLessMissing (c)* obteve os melhores resultados.

Sendo assim a seguir são analisados os resultados das quatro bases na qual a Ordem Externa (c) venceu, para posteriormente analisar a *Wine*. Esta ordem representa que os grupos com menor média de células ausentes por tupla foram processados em primeiro lugar. A supremacia desta configuração sugere que tanto o número de células ausentes como a quantidade de registros nos grupos influencia a qualidade da imputação. Isto também indica que as morfologias mais simples, quando imputadas inicialmente, auxiliam o processo de complementação como um todo. A ordem interna não modificou os valores do erro mínimo na *Íris Plants.* Em todos os conjuntos, este valor foi 2.027. Nas demais, há um salto de 0.01 para a *Machine* 0.4 para a *Pima* e 0.5 para a *Breast* do primeiro lugar para o segundo lugar, quando então se manteve constante ou com modificações em aproximadamente 0.02. Portanto, sem perda de generalidade, pode-se afirmar que o ranking interno, ou seja, de Ordem Interna, quando a Ordem Externa é *FieldperTupleLessMissing,* nos experimentos realizados é*: a-d-b-c-e*, ou seja, l*essCorrelation, moreMissing, lessMissing, moreCorrelation, noSort*. Portanto, as variáveis com maior independência quando imputadas primeiro contribuem de forma mais significativa para obter-se melhores resultados, seguidas pelas variáveis com mais células ausentes. Esta seqüência é bem plausível, pois, acredita-se que as

variáveis independentes são mais “fáceis” de imputar. Seus valores não dependem de valores de outros atributos. Caso bons doadores estejam presentes, os valores estimados são bons também. O segundo lugar também é bastante intuitivo pois ao regredir os atributos com maior número de células ausentes, a quantidade de tuplas disponíveis aumenta diversificando os possíveis doadores para as próximas imputações. Aqui, a ordem de imputação fez diferença, pois a ordem aleatória ficou em último lugar nesta Ordem externa.

Em todas as bases a diferença da ordem Externa vencedora para as demais é bem significativa, quando então se mantém equilibradas. As ordens *tupleMoreMissing* (g) e *fieldPerTupleMoreMissing* (d) empatam no segundo lugar seguidas da *tupleLessMissing* (f) e *fieldLessMissing* (a) Na base *IrisPlants* e *Breast* há uma tendência as seguinte configuração de Ordem Interna (d) – (a) – (b) – (c): *moreMissing* - *lessCorrelation* - *lessMissing* - *moreCorrelation* que é a inversa a da Ordem Externa vencedora, mas segue a mesma lógica. Na *Machine* e na *Pima*, as ordens Internas, produzem tecnicamente o mesmo valor (variações inferiores a 0.03). Novamente a ordem de imputação fez diferença, pois a ordem aleatória (e) ficou também em último lugar.

Na base Wine, externamente as ordens vencedoras empatadas são *tupleMoreMissing* (g) e *fieldPerTupleMoreMissing* (d) seguidas da (f) *tupleLessMissing*. Internamente a seqüência *moreCorrelation* (c) - *moreMissing* (d) - *lessMissing* (b) mantém-se empatada no primeiro lugar. Embora praticamente ao inverso das demais bases, as ordens que regem os melhores resultados nesta base estão, também considerando a complexidade da morfologia da ausência (pela relação quantidade de células ausentes por tuplas) e internamente a correlação entre os atributos está influenciando mais que a quantidade de células ausentes.

5.3.7 Classificação das ordenações

A seguir, mostra-se o comportamento das configurações de Ordem Externa e Ordem Interna. Um *ranking* é uma forma de visualizar a posição dos elementos de um conjunto em relação aos demais, considerando alguma característica. Portanto, a partir desta ordenação parcial do conjunto de elementos, pode-se compará-los entre si pelo critério escolhido.

A métrica que serviu de base para os critérios foi calculada considerando-se o erro mínimo da configuração para cada base e percentual de ausência. Portanto, quarenta e

cinco rodadas foram realizadas. Em cada rodada, seleciona-se o menor erro obtido por cada uma das configurações. Este erro é usado para classificar as configurações e distribuir a pontuação. Se uma determinada configuração ficou em primeiro lugar nesta rodada, ganha um ponto no primeiro lugar, se ficou em segundo, ganha um ponto no segundo lugar e assim sucessivamente. Deste modo, toda configuração pontua em toda a rodada, o que varia é a posição da pontuação.

Os gráficos 5.24 a 5.58 mostram o desempenho de cada combinação de ordenação (Ordem Externa, Ordem Interna). Deste modo, no eixo X estão representadas as bases, no eixo Y a frequência que a ordenação ocupou o enésimo lugar representado no eixo Z. Desta forma é fácil visualizar o comportamento de cada ordenação. As melhores ordenações tem suas pontuações concentradas no início do eixo Z, as piores no final e as medianas no centro. As configurações com distribuição quase uniforme são inconclusivas.

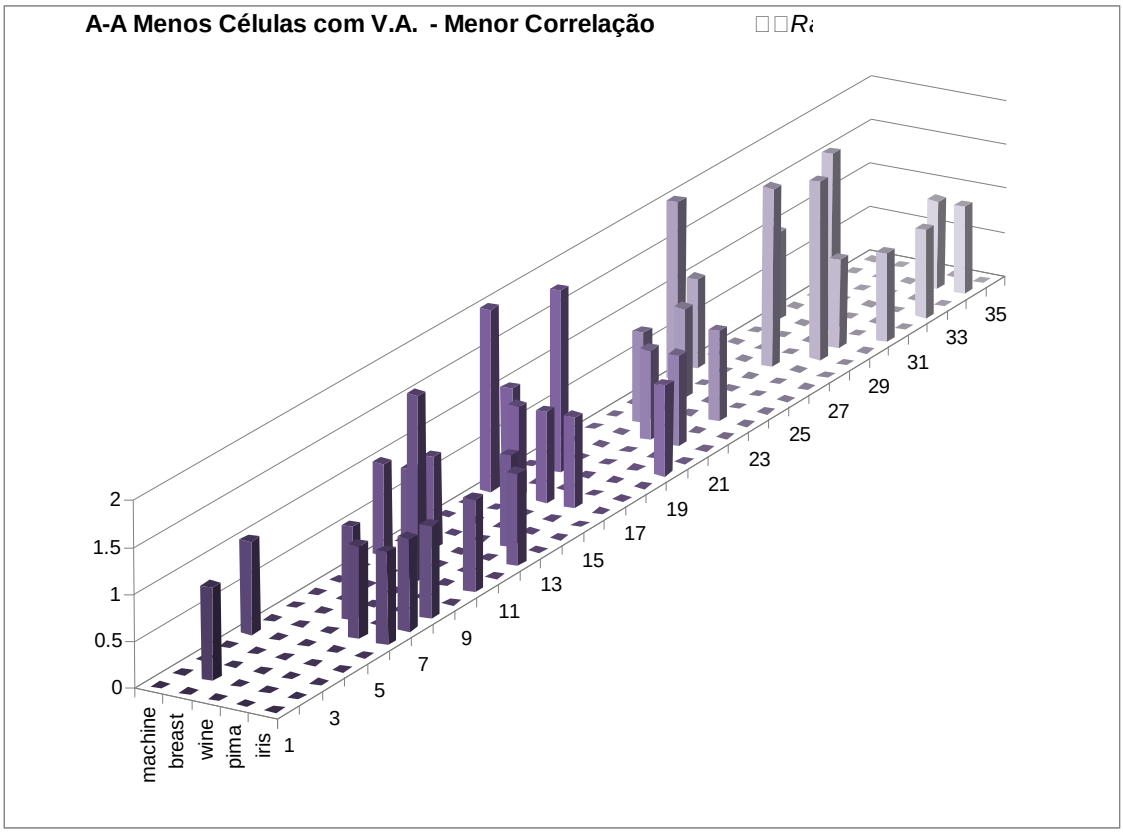


Gráfico 5.24- Ranking da ordenação (A,A) nas Bases
Critério de Ordenação dos Grupos (Externa): A - Menos Células com Valores Ausentes
Critério de Ordenação de Atributos (Interna): A - Menor Correlação.

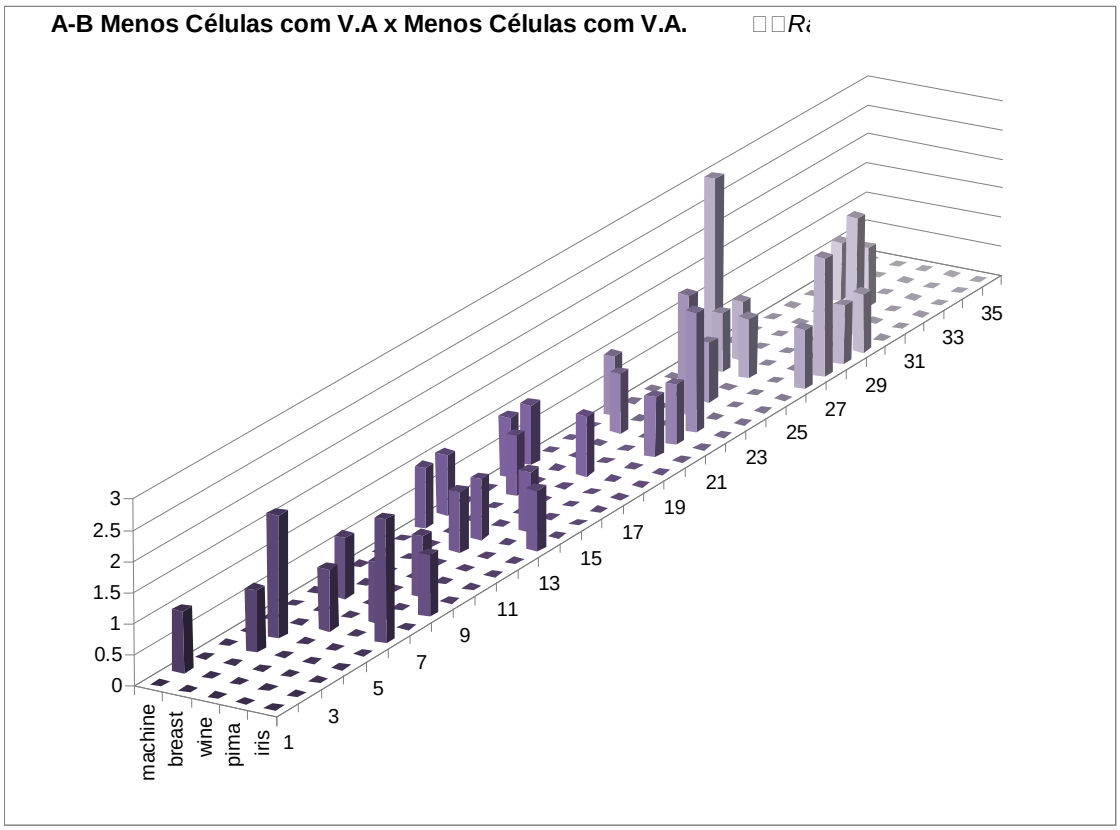


Gráfico 5.25- Ranking da ordenação (A,B) nas Bases
Critério de Ordenação dos Grupos (Externa): A - Menos Células com Valores Ausentes
Critério de Ordenação de Atributos(Interna): B – Menos Células com Valores Ausentes.

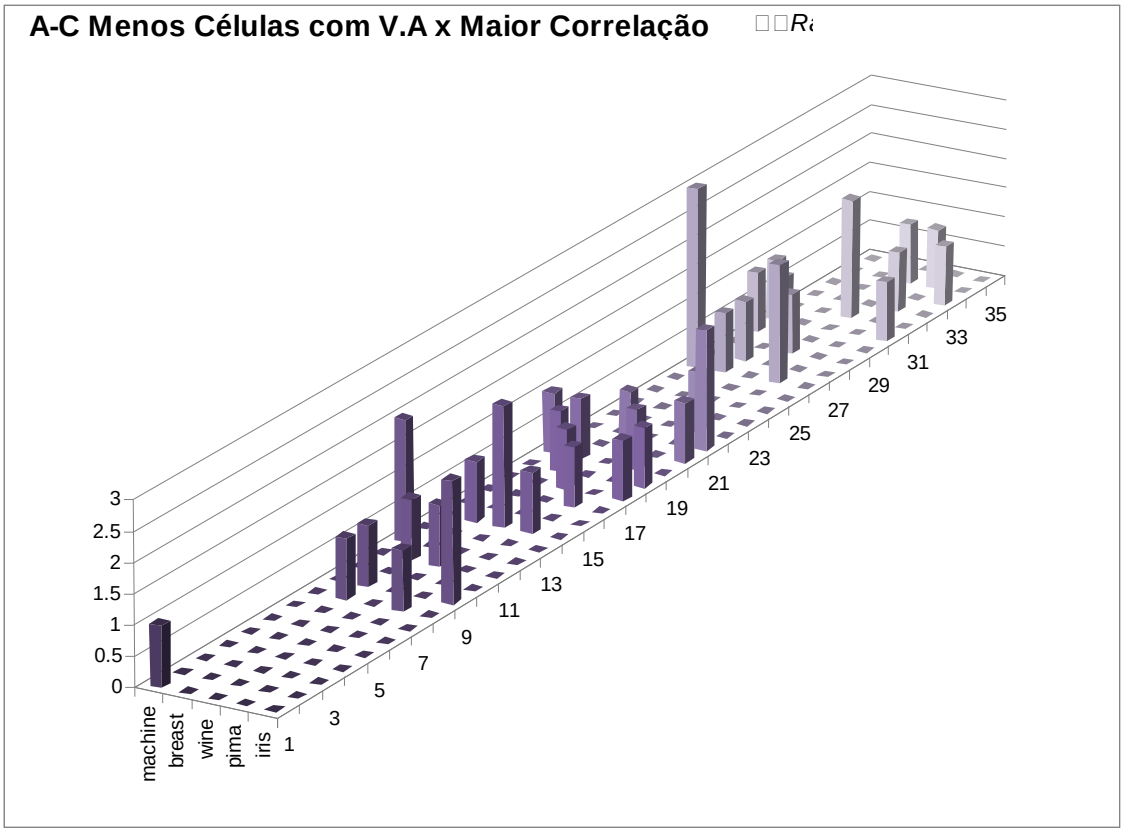


Gráfico 5.26- Ranking da ordenação (A,C) nas Bases
Critério de Ordenação dos Grupos (Externa): A - Menos Células com Valores Ausentes
Critério de Ordenação de Atributos(Interna): C – Maior Correlação.

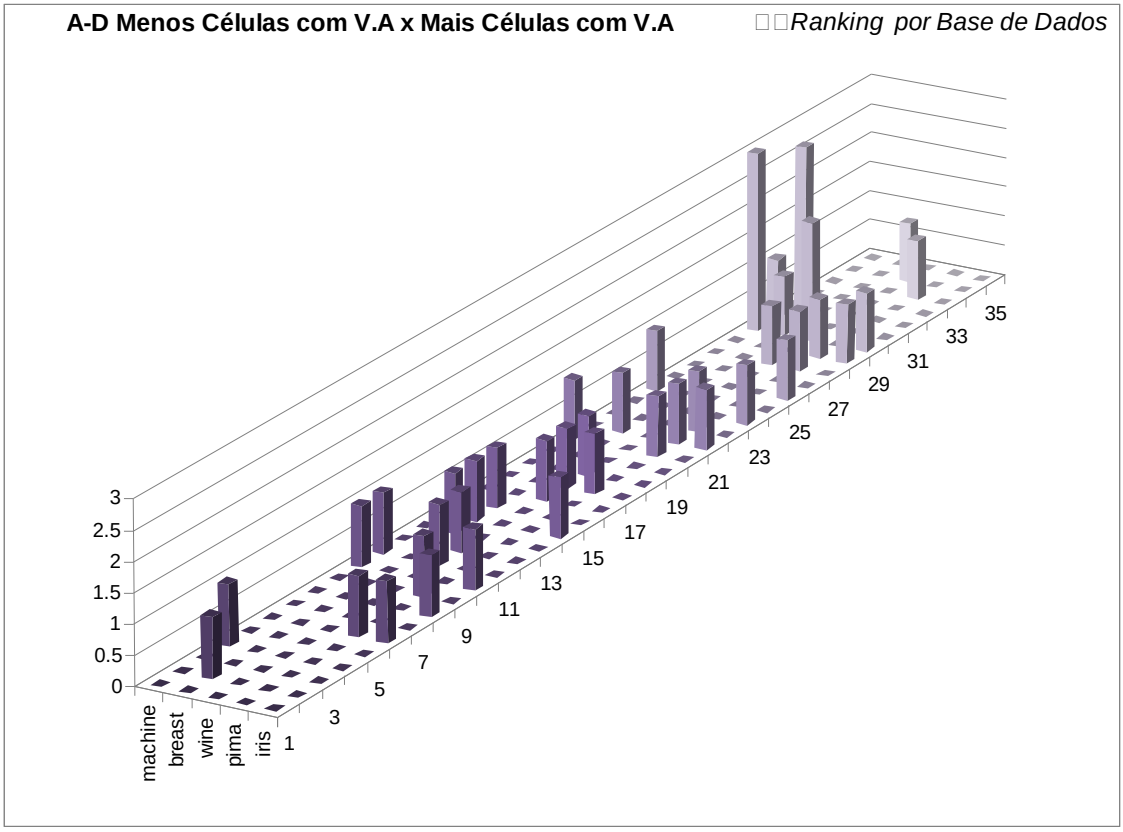


Gráfico 5.27- Ranking da ordenação (A,D) nas Bases
Critério de Ordenação dos Grupos (Externa): A - Menos Células com Valores Ausentes
Critério de Ordenação de Atributos(Interna):D – Mais Células com Valores Ausentes

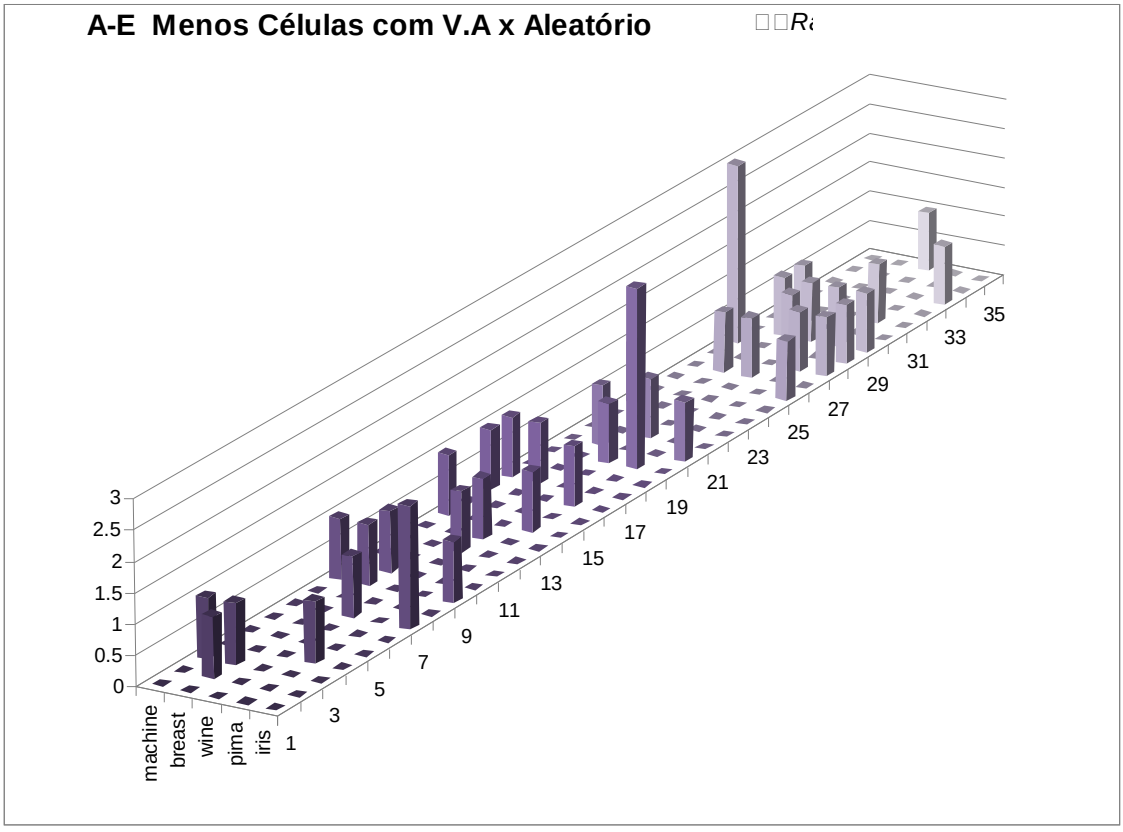


Gráfico 5.28- Ranking da ordenação (A,E) nas Bases
Critério de Ordenação dos Grupos (Externa): A - Menos Células com Valores Ausentes
Critério de Ordenação de Atributos(Interna):E – Aleatório

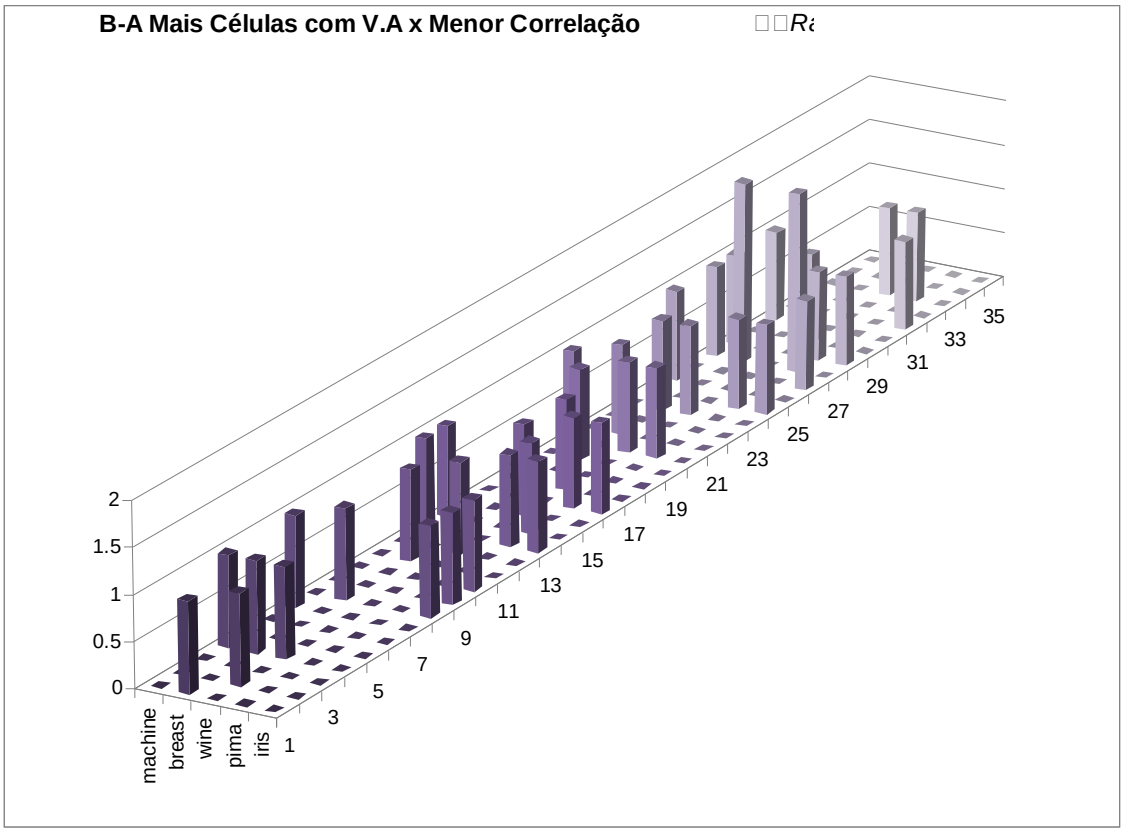


Gráfico 5.29- Ranking da ordenação (B,A) nas Bases
Critério de Ordenação dos Grupos (Externa): B - Mais Células com Valores Ausentes
Critério de Ordenação de Atributos (Interna): A - Menor Correlação.

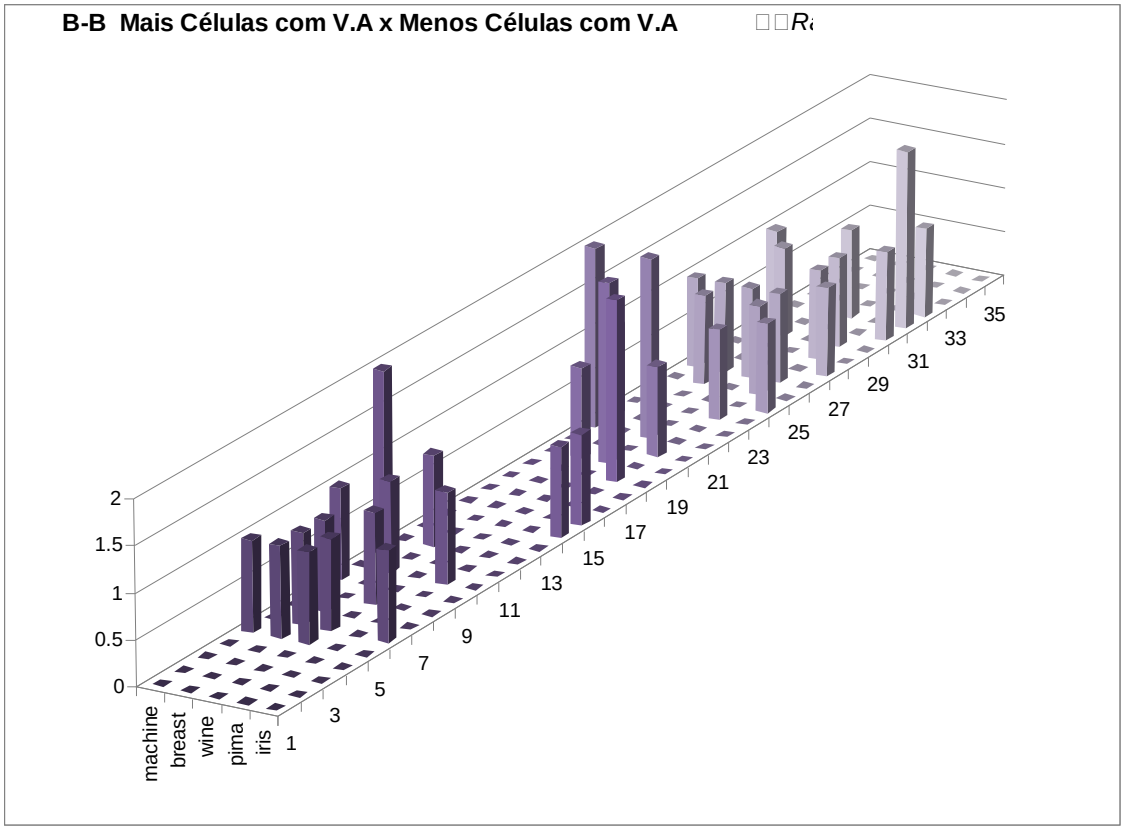


Gráfico 5.30- Ranking da ordenação (B,B) nas Bases
Critério de Ordenação dos Grupos (Externa): B - Mais Células com Valores Ausentes
Critério de Ordenação de Atributos(Interna): B – Menos Células com Valores Ausentes.

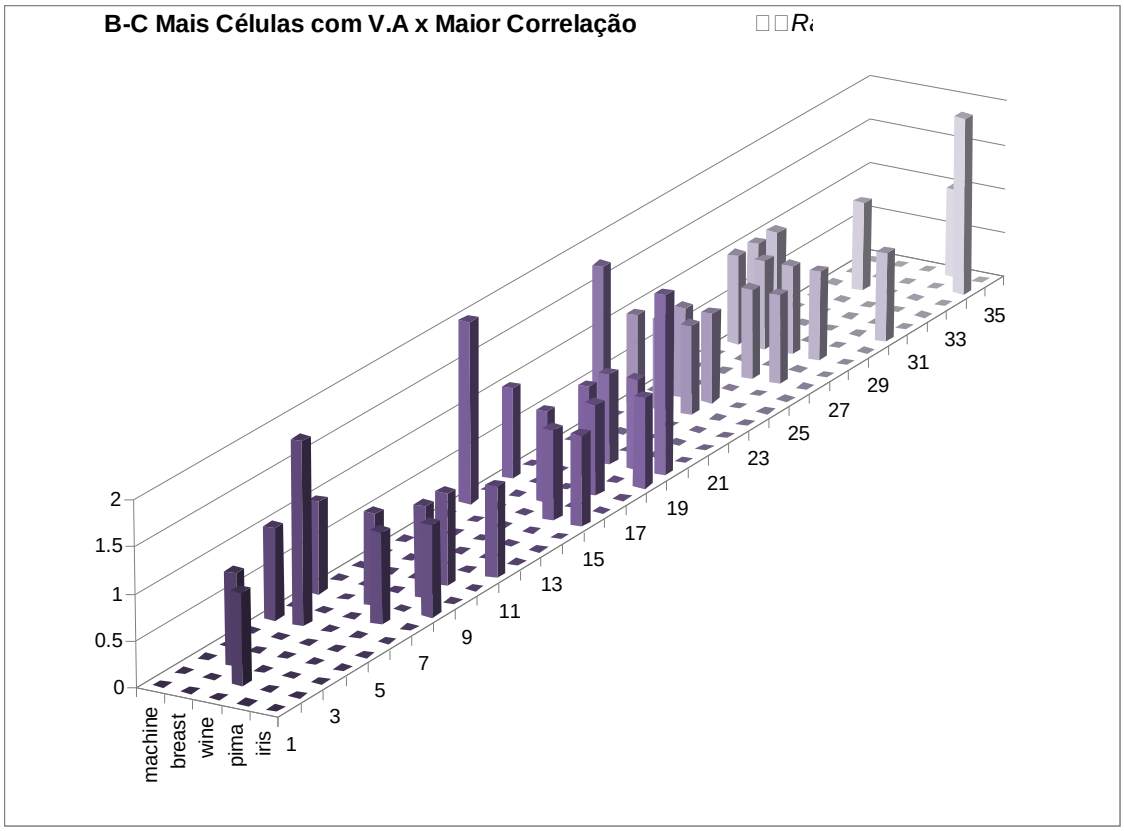


Gráfico 5.31- Ranking da ordenação (B,C) nas Bases
Critério de Ordenação dos Grupos (Externa): A - Mais Células com Valores Ausentes
Critério de Ordenação de Atributos(Interna): C – Maior Correlação.

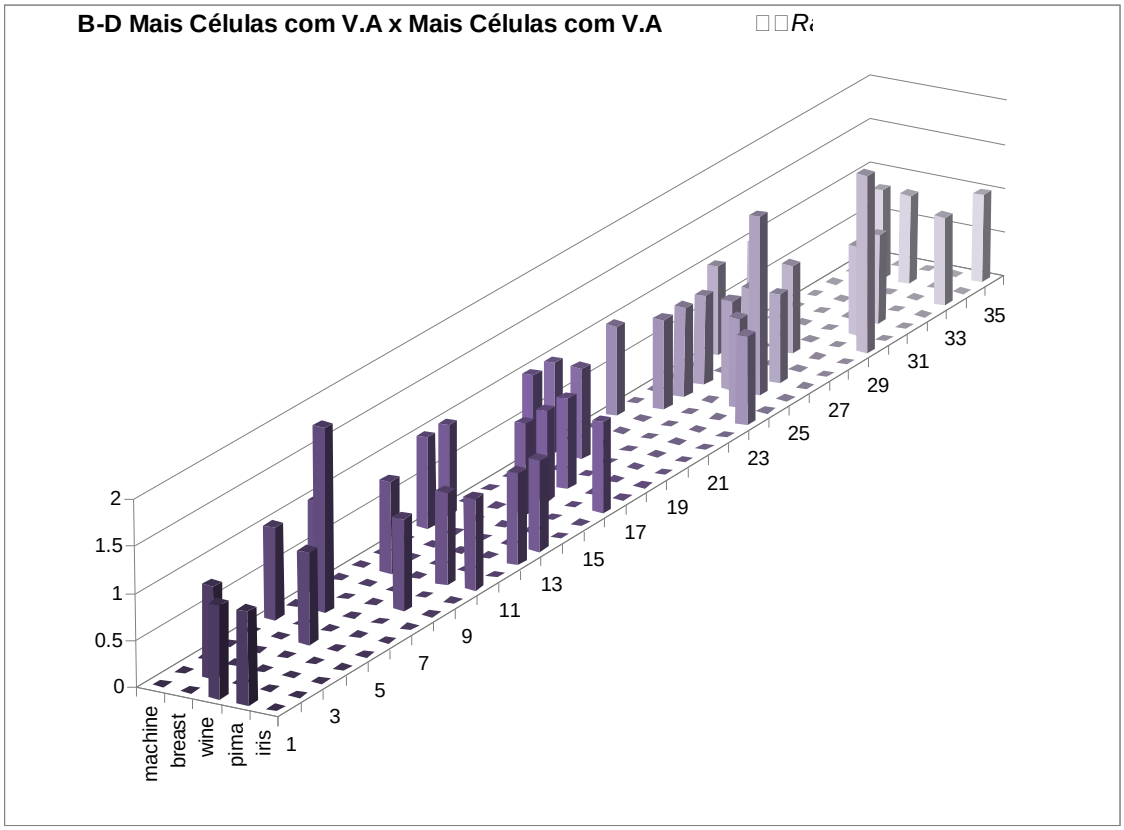


Gráfico 5.32- Ranking da ordenação (B,D) nas Bases
Critério de Ordenação dos Grupos (Externa): B - Mais Células com Valores Ausentes
Critério de Ordenação de Atributos(Interna):D – Mais Células com Valores Ausentes

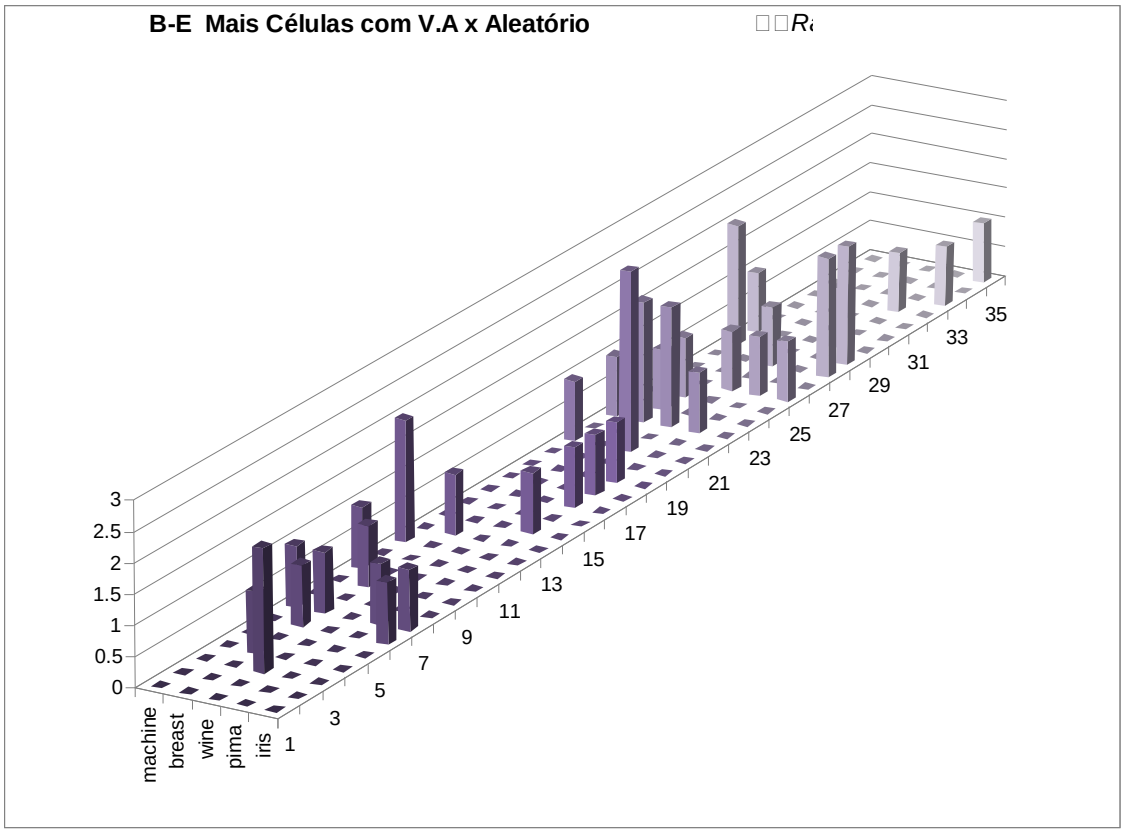


Gráfico 5.33- Ranking da ordenação (B,E) nas Bases
Critério de Ordenação dos Grupos (Externa): B - Mais Células com Valores Ausentes
Critério de Ordenação de Atributos(Interna):E – Aleatório

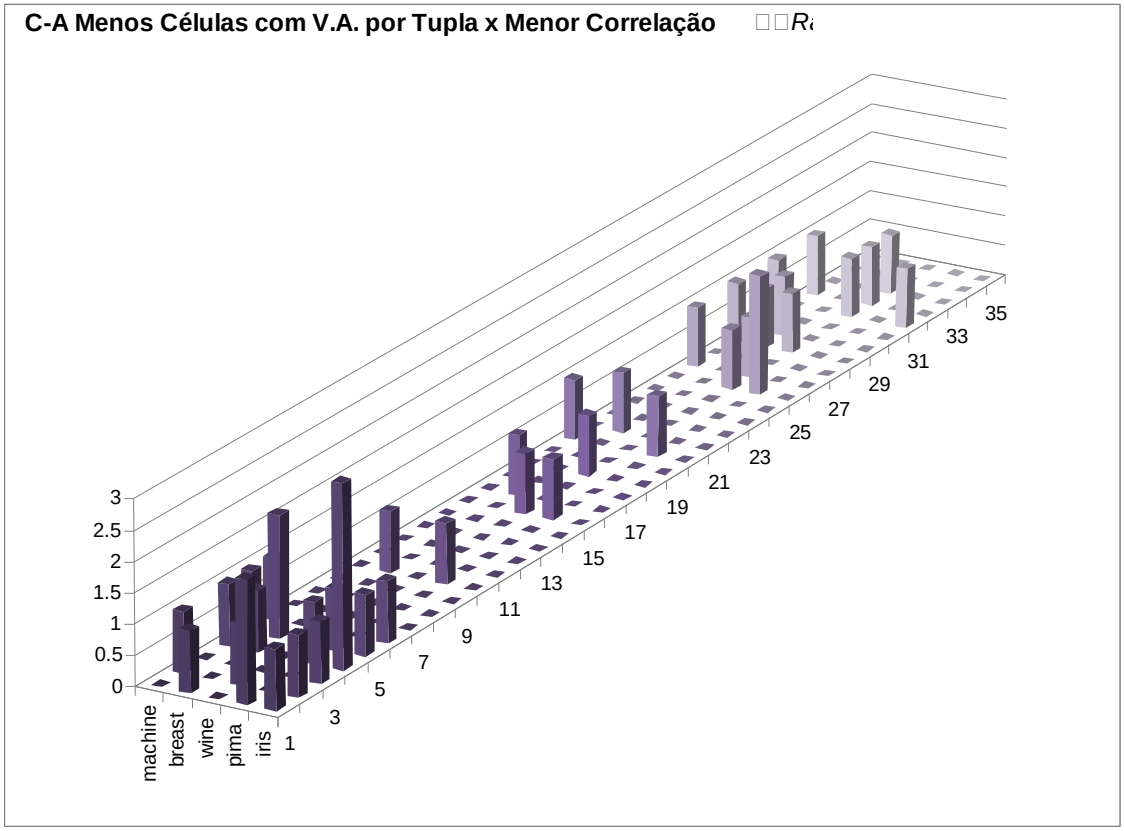


Gráfico 5.34- Ranking da ordenação (C,A) nas Bases
Critério de Ordenação dos Grupos (Externa): C - Menos Células com Valores Ausentes por Tupla
Critério de Ordenação de Atributos (Interna): A - Menor Correlação.

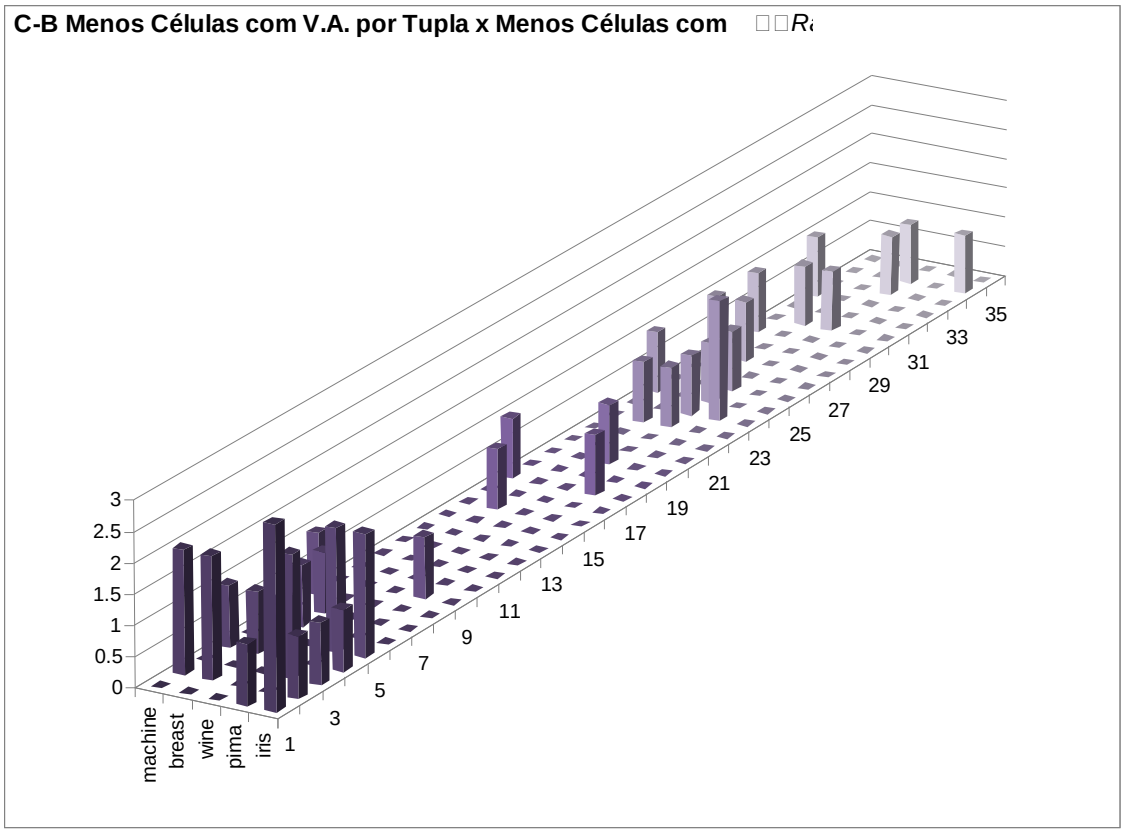


Gráfico 5.35- Ranking da ordenação (C,B) nas Bases
Critério de Ordenação dos Grupos (Externa): A - Menos Células com Valores Ausentes por Tupla
Critério de Ordenação de Atributos(Interna): B – Menos Células com Valores Ausentes.

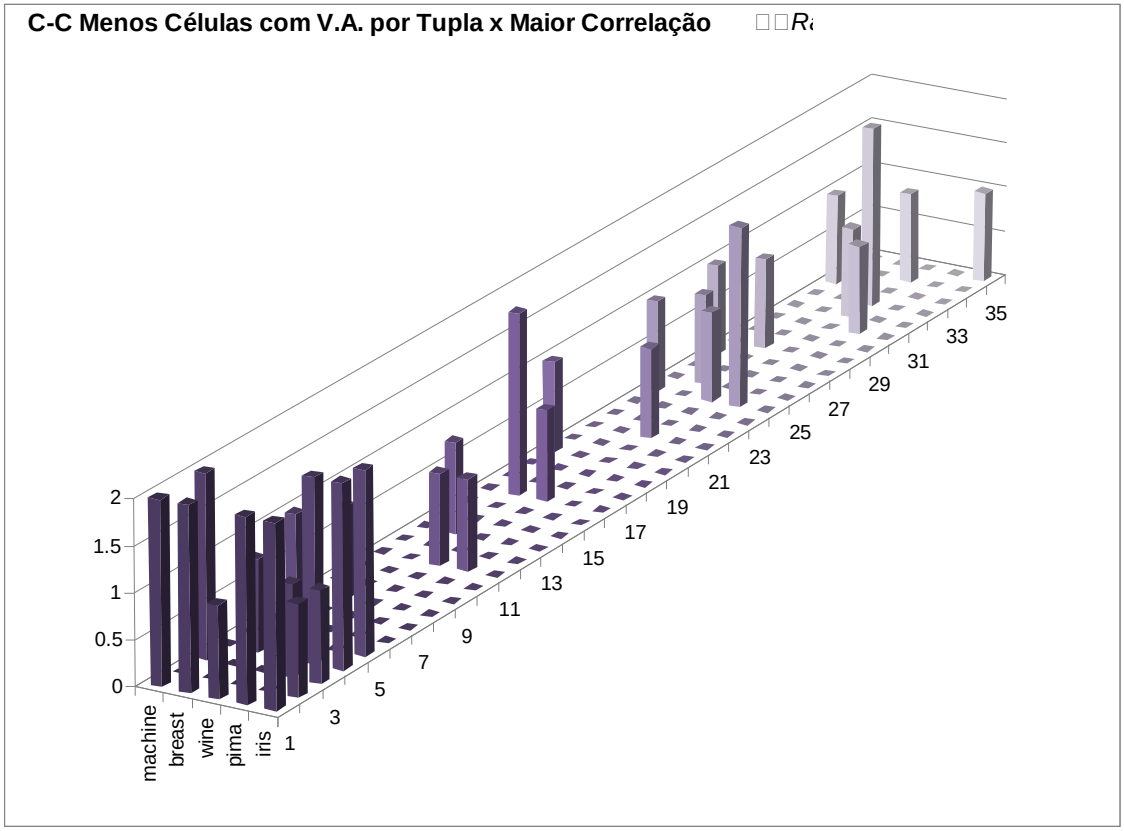


Gráfico 5.36- Ranking da ordenação (C,C) nas Bases
Critério de Ordenação dos Grupos (Externa): A - Menos Células com Valores Ausentes por Tupla
Critério de Ordenação de Atributos(Interna): C – Maior Correlação.

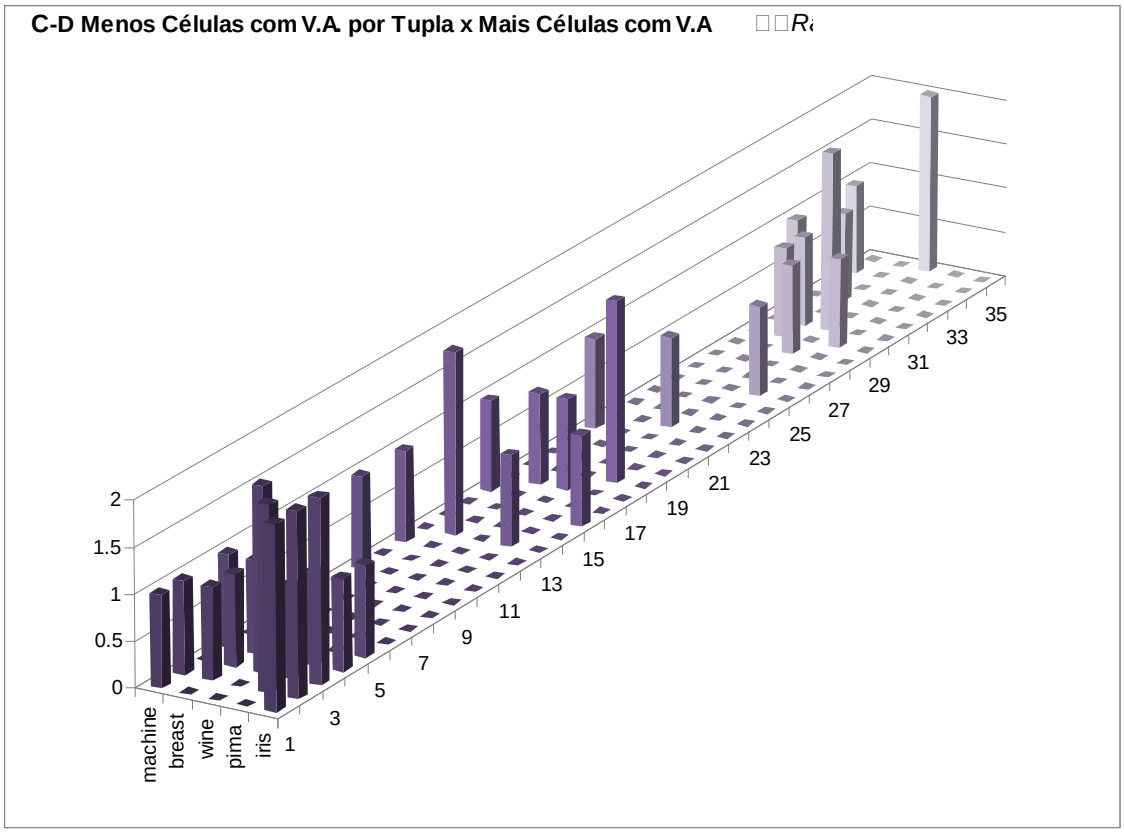


Gráfico 5.37- Ranking da ordenação (C,D) nas Bases
Critério de Ordenação dos Grupos (Externa): C - Menos Células com Valores Ausentes por Tupla
Critério de Ordenação de Atributos(Interna):D – Mais Células com Valores Ausentes

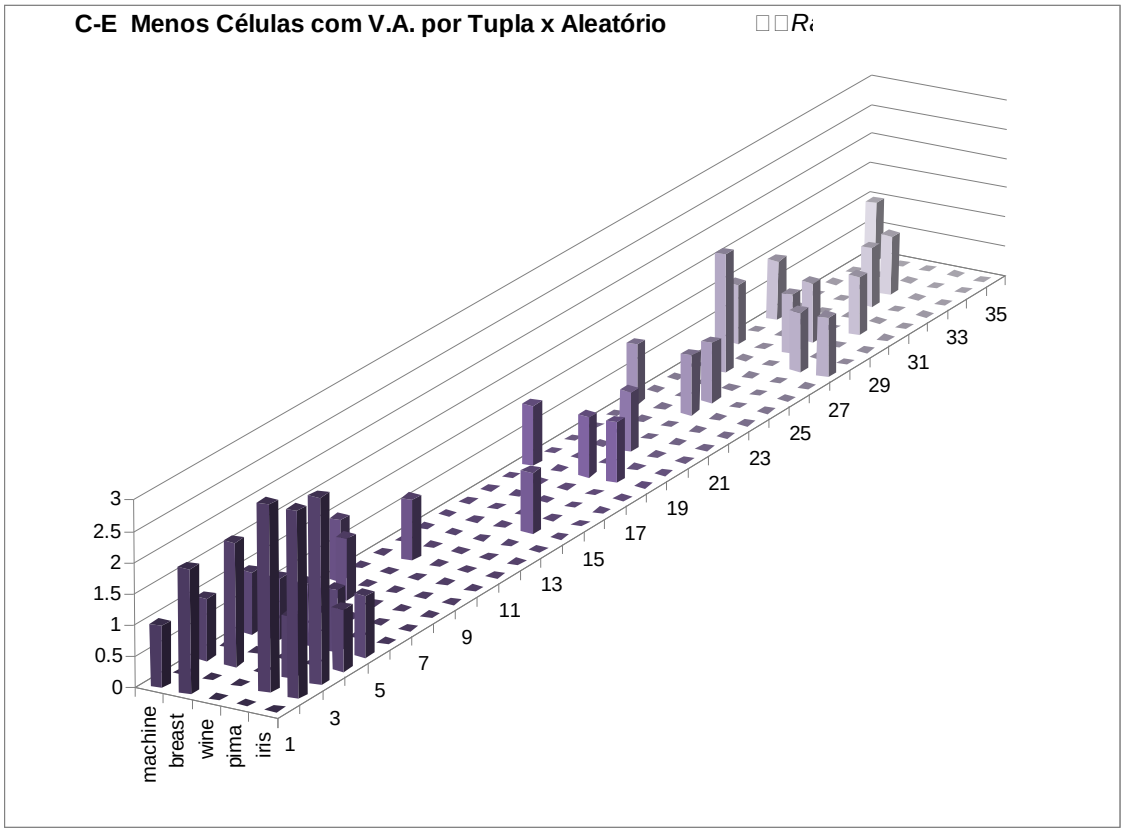


Gráfico 5.38- Ranking da ordenação (C,E) nas Bases
Critério de Ordenação dos Grupos (Externa): A - Menos Células com Valores Ausentes por Tupla
Critério de Ordenação de Atributos(Interna):E – Aleatório

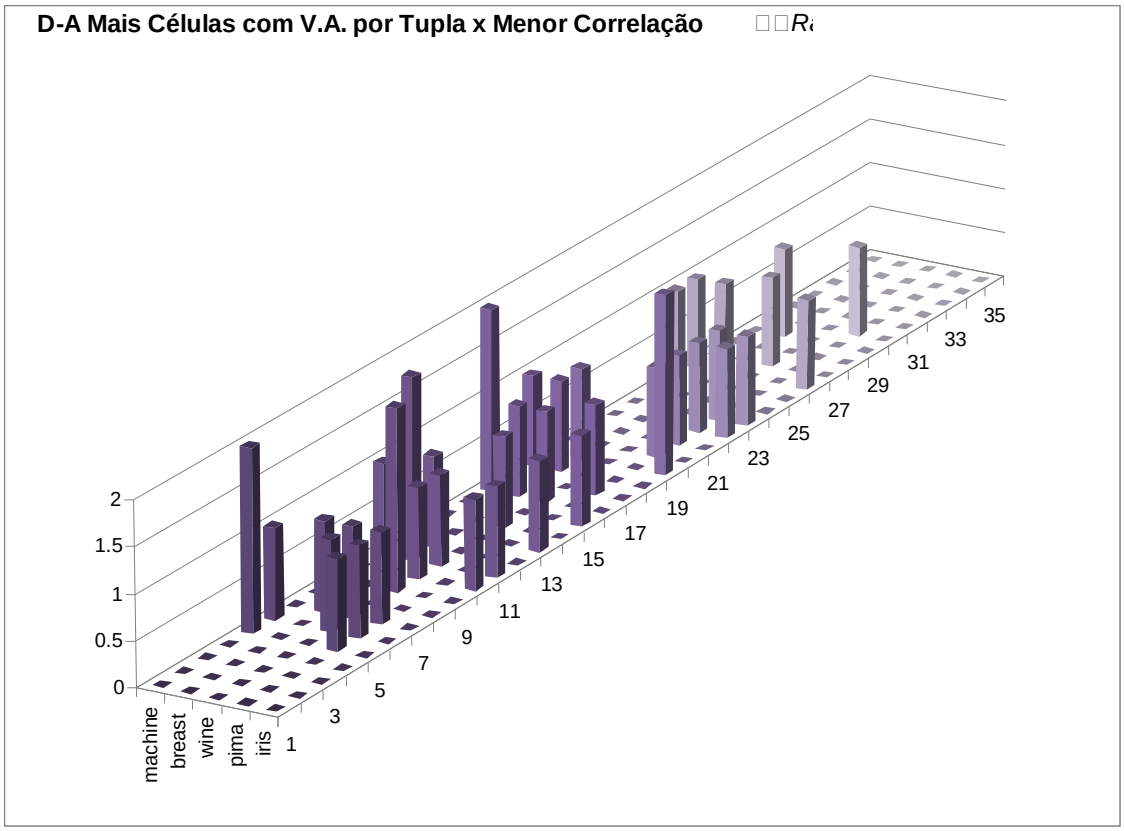


Gráfico 5.39- Ranking da ordenação (D,A) nas Bases
Critério de Ordenação dos Grupos (Externa): D - Mais Células com Valores Ausentes por Tupla
Critério de Ordenação de Atributos (Interna): A - Menor Correlação.

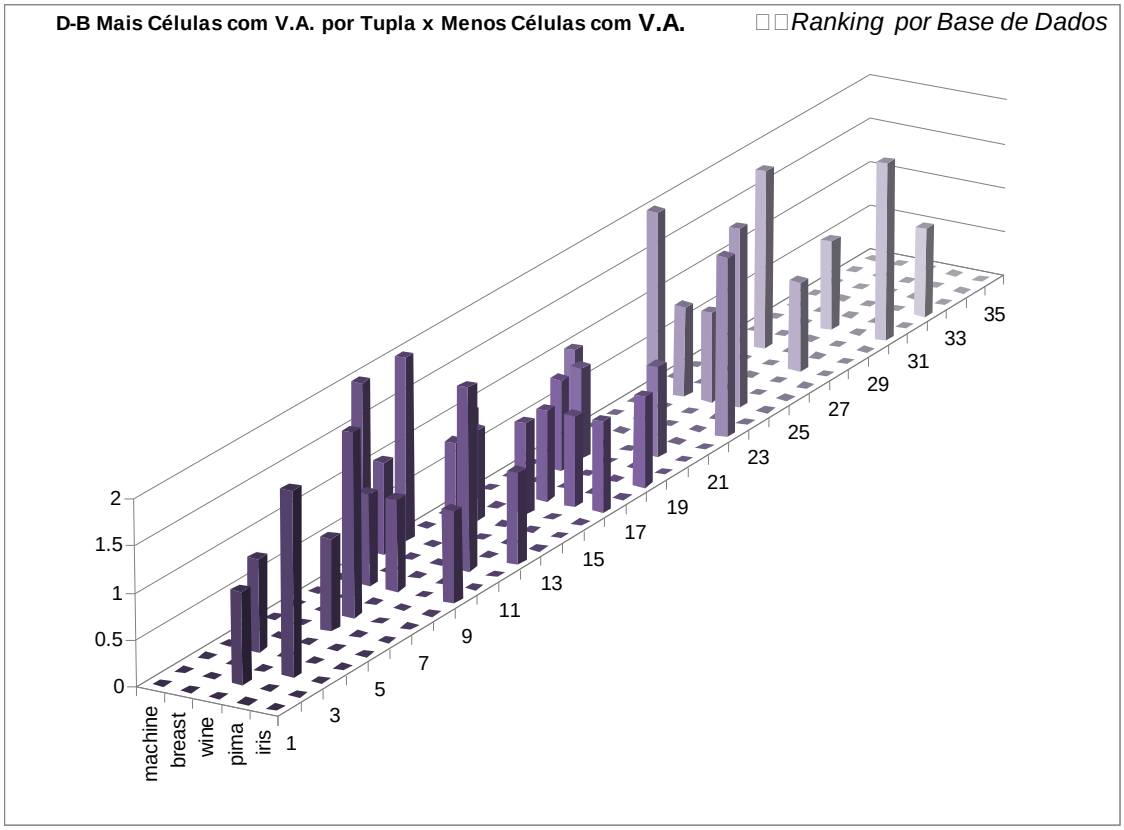


Gráfico 5.40- Ranking da ordenação (D,B) nas Bases
Critério de Ordenação dos Grupos (Externa): D - Mais Células com Valores Ausentes por Tupla
Critério de Ordenação de Atributos(Interna): B – Menos Células com Valores Ausentes.

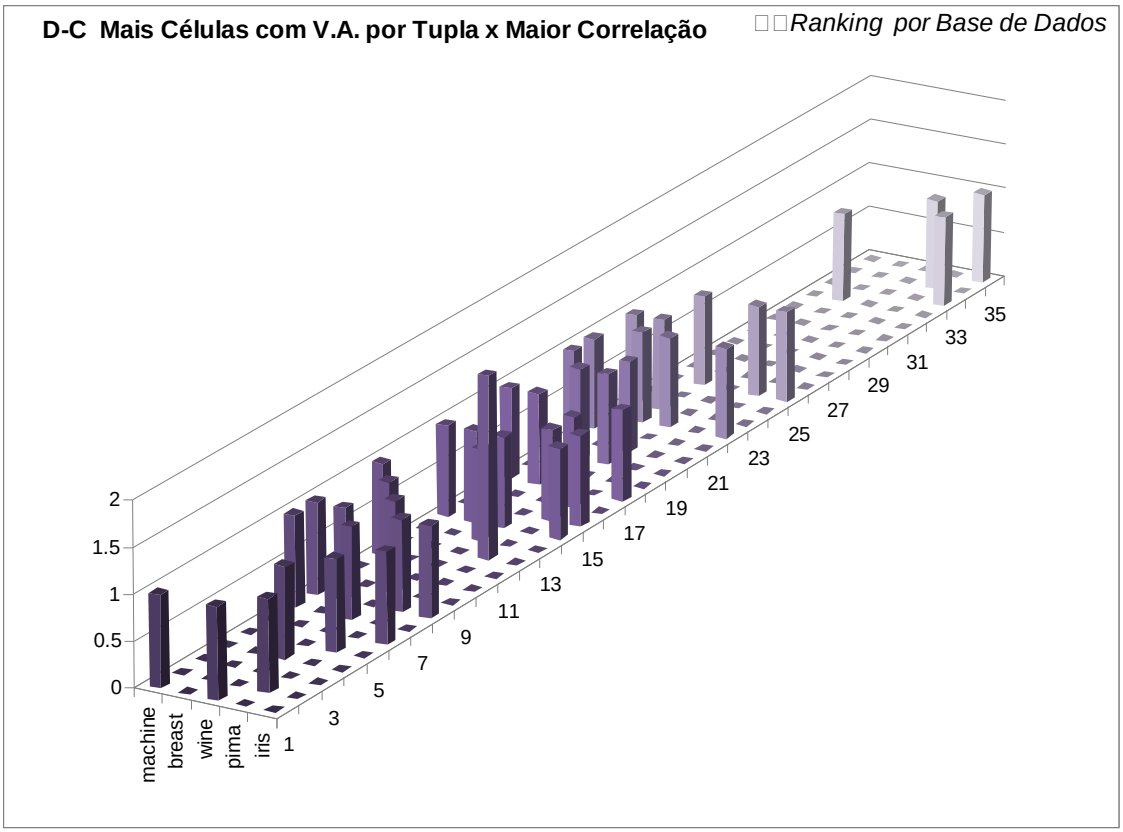


Gráfico 5.41- Ranking da ordenação (D,C) nas Bases
Critério de Ordenação dos Grupos (Externa):D - Mais Células com Valores Ausentes por Tupla
Critério de Ordenação de Atributos(Interna): C – Maior Correlação.

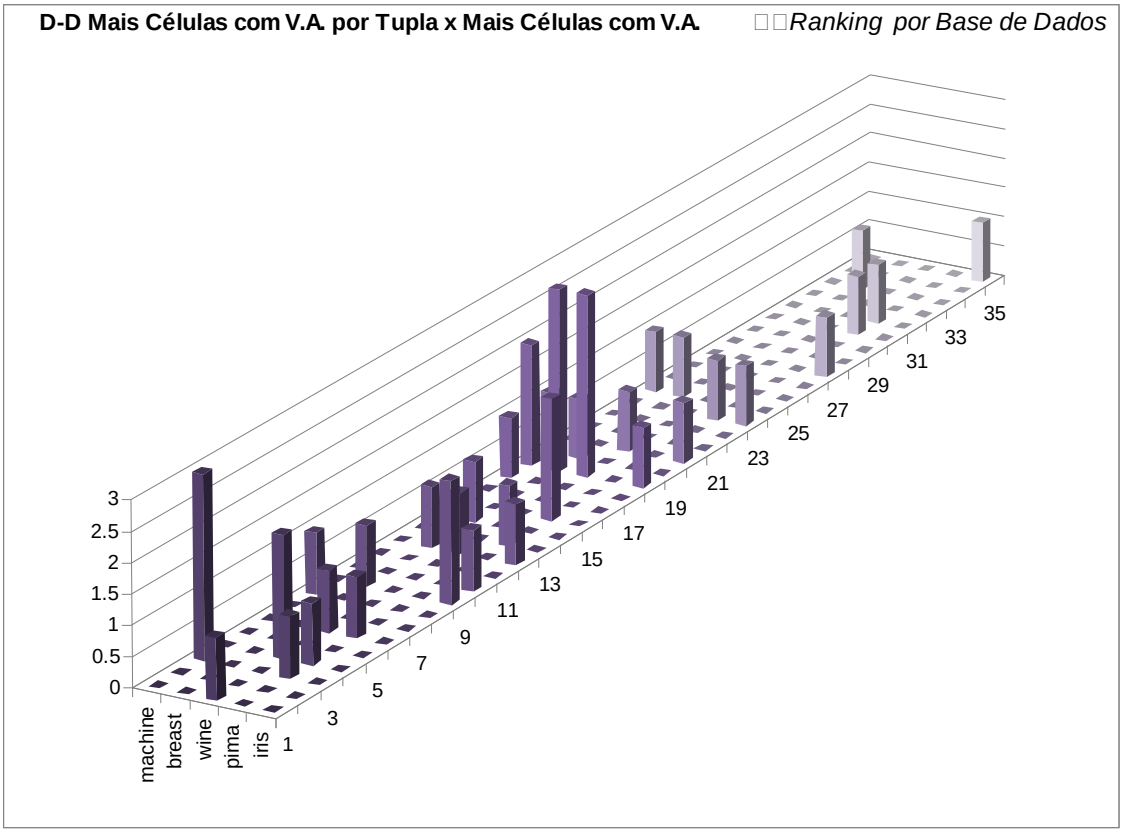


Gráfico 5.42- Ranking da ordenação (D,D) nas Bases
Critério de Ordenação dos Grupos (Externa): D - Mais Células com Valores Ausentes por Tupla
Critério de Ordenação de Atributos(Interna):D – Mais Células com Valores Ausentes

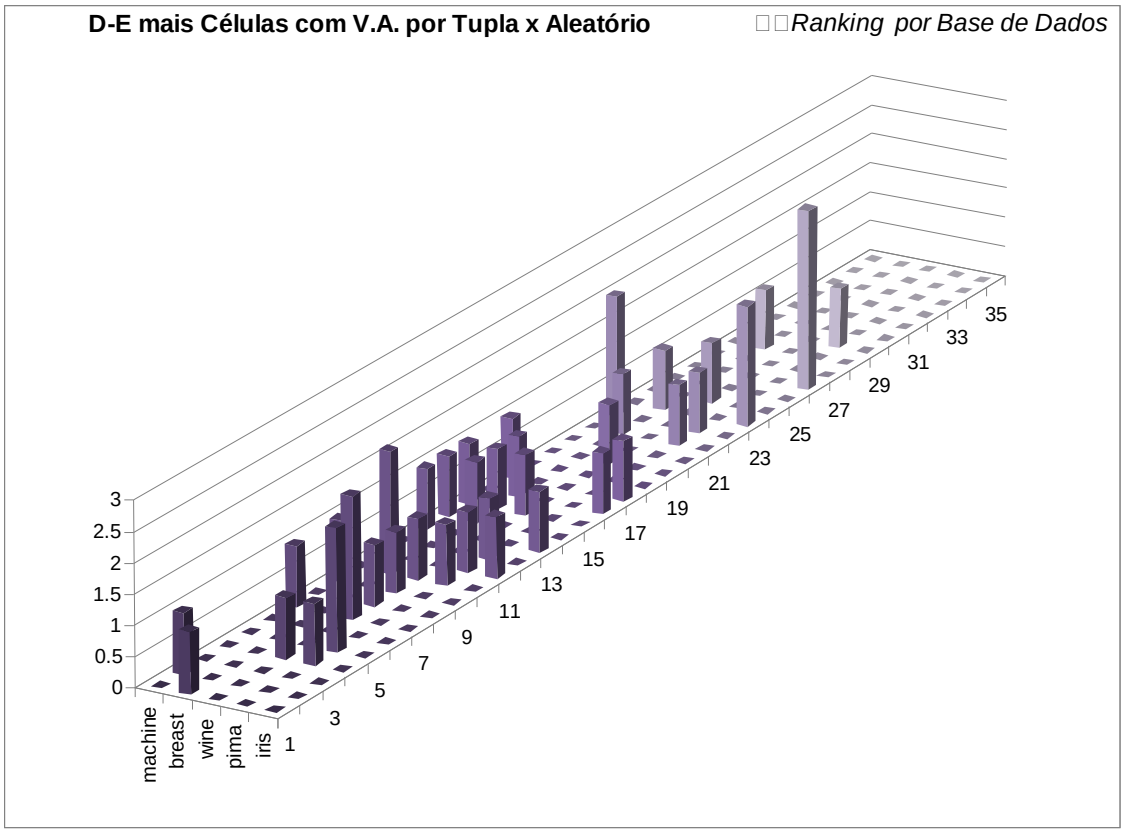


Gráfico 5.43- Ranking da ordenação (D,E) nas Bases
Critério de Ordenação dos Grupos (Externa): D - Mais Células com Valores Ausentes por Tupla
Critério de Ordenação de Atributos(Interna):E – Aleatório

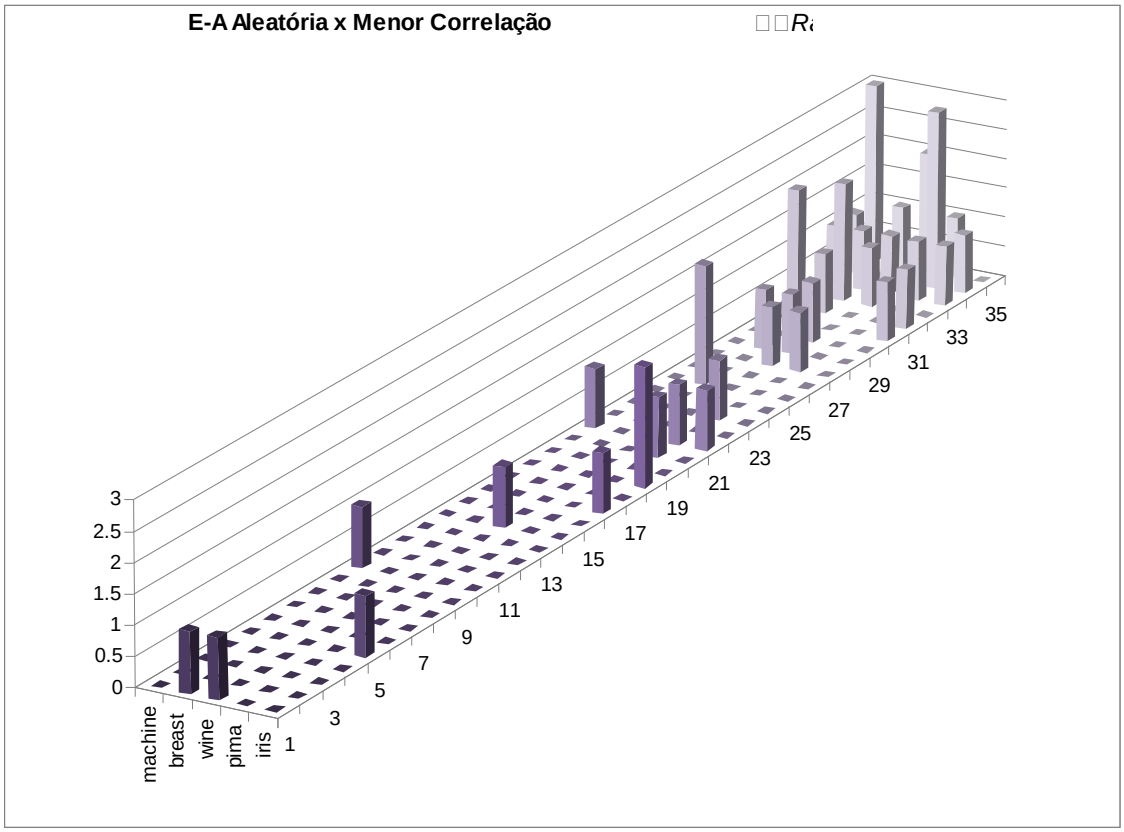


Gráfico 5.44- Ranking da ordenação (E,A) nas Bases
Critério de Ordenação dos Grupos (Externa): E- Aleatória
Critério de Ordenação de Atributos (Interna): A - Menor Correlação.

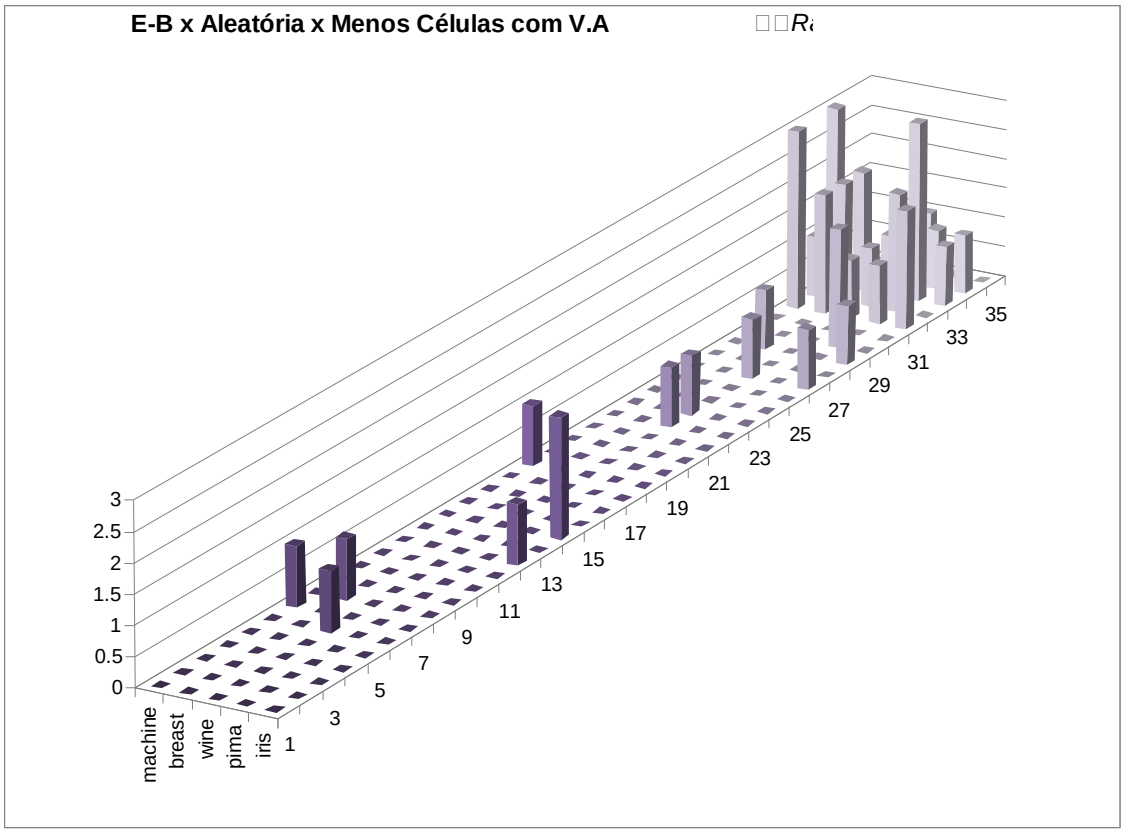


Gráfico 5.45- Ranking da ordenação (E,B) nas Bases
Critério de Ordenação dos Grupos (Externa): E-Aleatória
Critério de Ordenação de Atributos(Interna): B – Menos Células com Valores Ausentes.

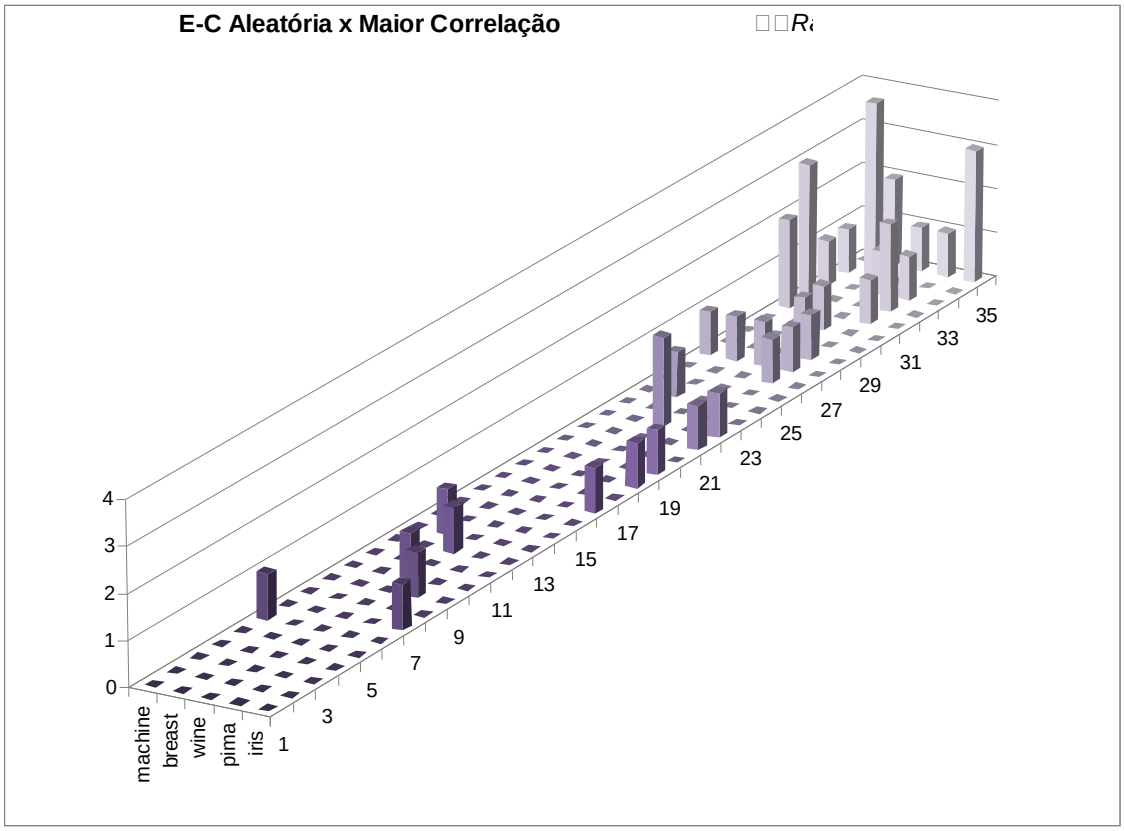


Gráfico 5.46- Ranking da ordenação (E,C) nas Bases
Critério de Ordenação dos Grupos (Externa): E-Aleatória
Critério de Ordenação de Atributos(Interna): C – Maior Correlação.

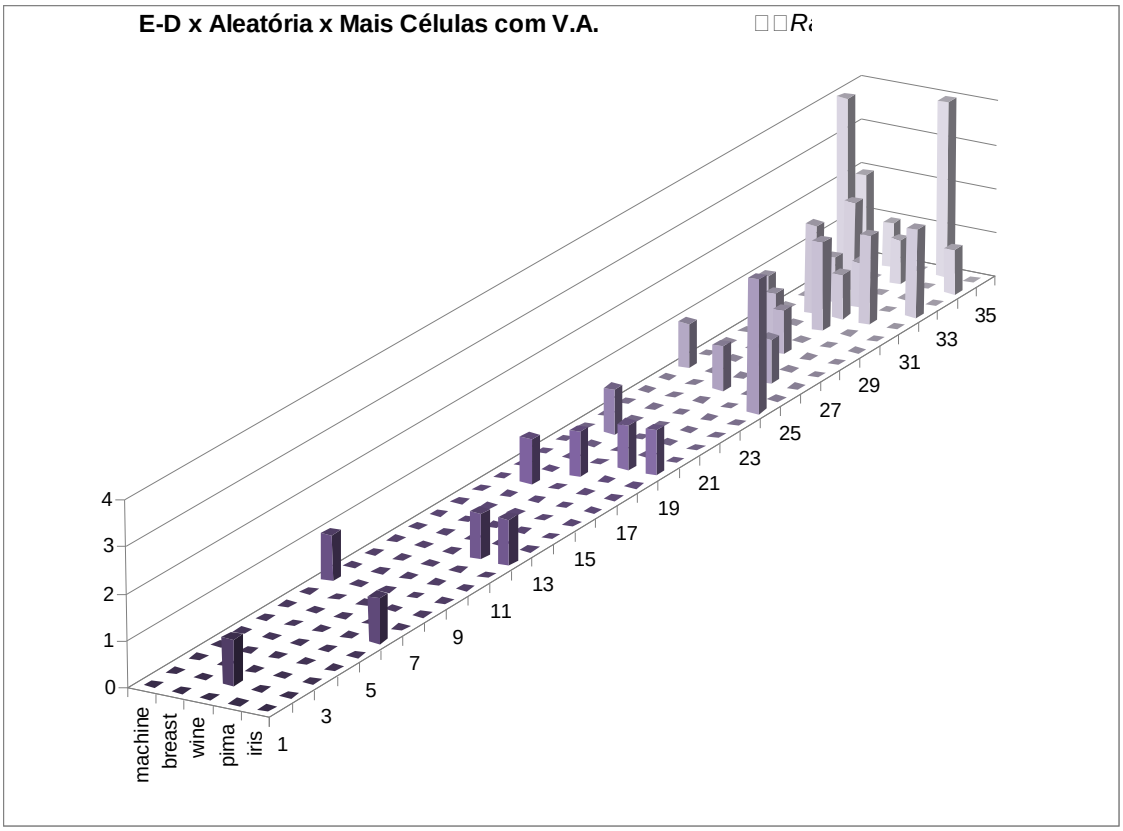


Gráfico 5.47- Ranking da ordenação (E,D) nas Bases
Critério de Ordenação dos Grupos (Externa): E -Aleatória
Critério de Ordenação de Atributos(Interna):D – Mais Células com Valores Ausentes

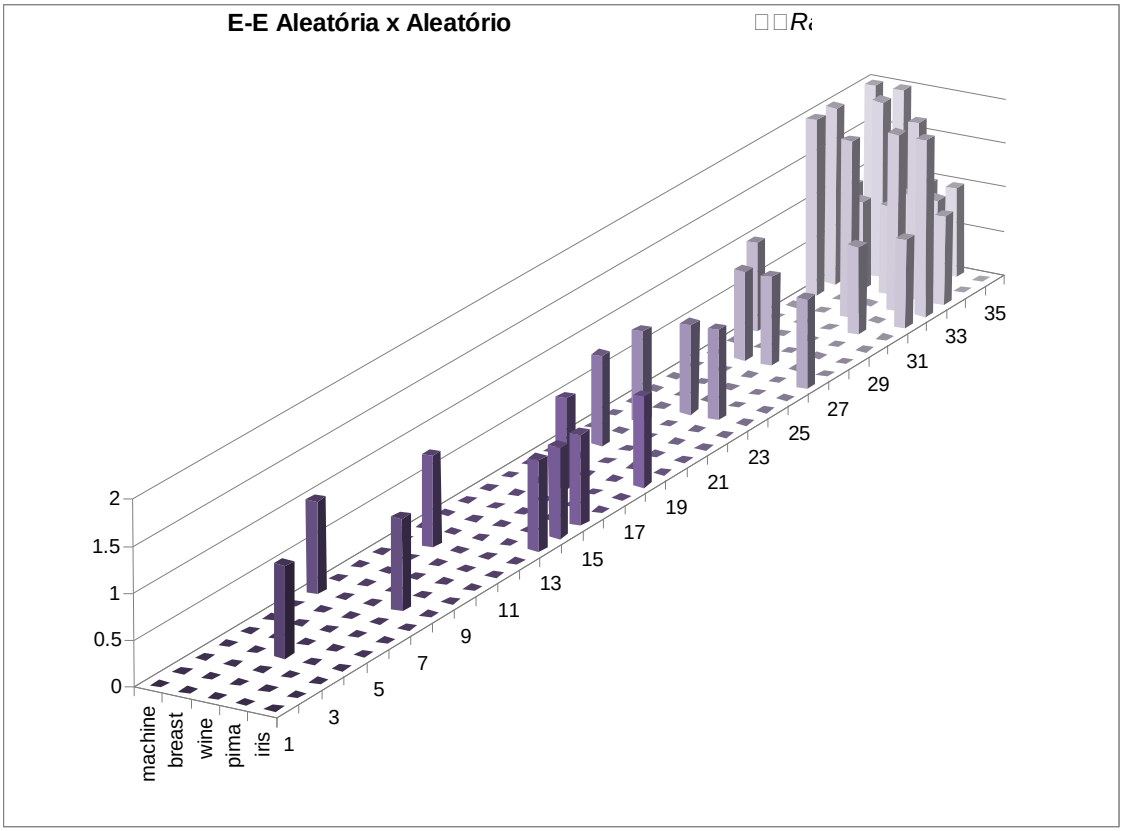


Gráfico 5.48- Ranking da ordenação (E,E) nas Bases
Critério de Ordenação dos Grupos (Externa): E - Aleatória
Critério de Ordenação de Atributos(Interna):E – Aleatório

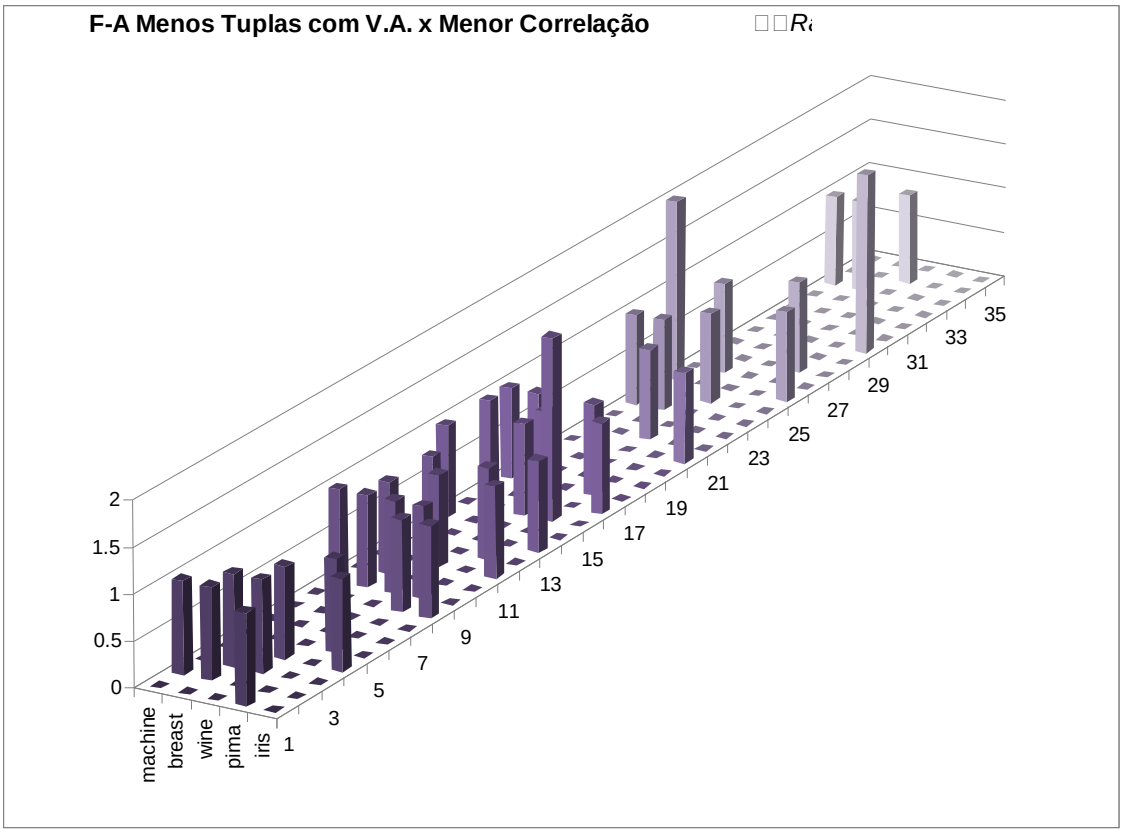


Gráfico 5.49- Ranking da ordenação (F,A) nas Bases
Critério de Ordenação dos Grupos (Externa): F - Menos Tuplas com Valores Ausentes
Critério de Ordenação de Atributos (Interna): A - Menor Correlação.

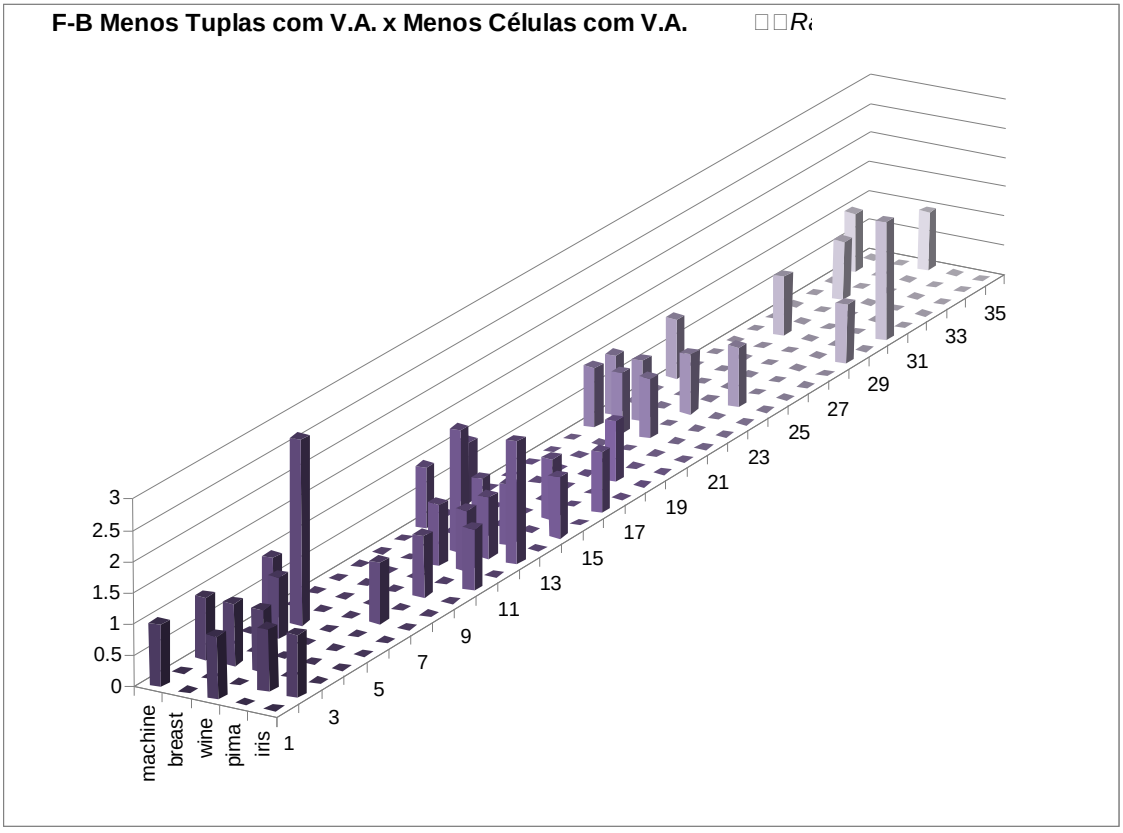


Gráfico 5.50- Ranking da ordenação (F,B) nas Bases
Critério de Ordenação dos Grupos (Externa): F - Menos Tuplas com Valores Ausentes
Critério de Ordenação de Atributos(Interna): B – Menos Células com Valores Ausentes.

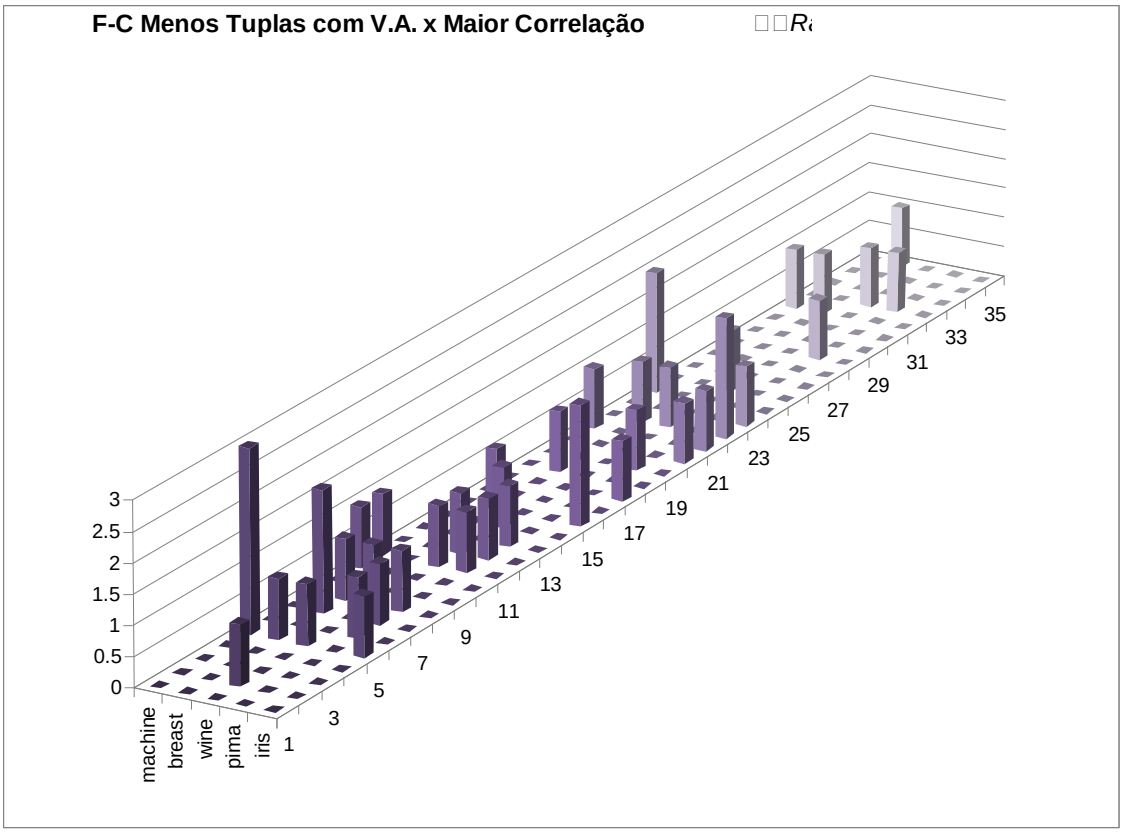


Gráfico 5.51- Ranking da ordenação (F,C) nas Bases
Critério de Ordenação dos Grupos (Externa): F – Menos Tuplas com Valores Ausentes
Critério de Ordenação de Atributos(Interna): C – Maior Correlação.

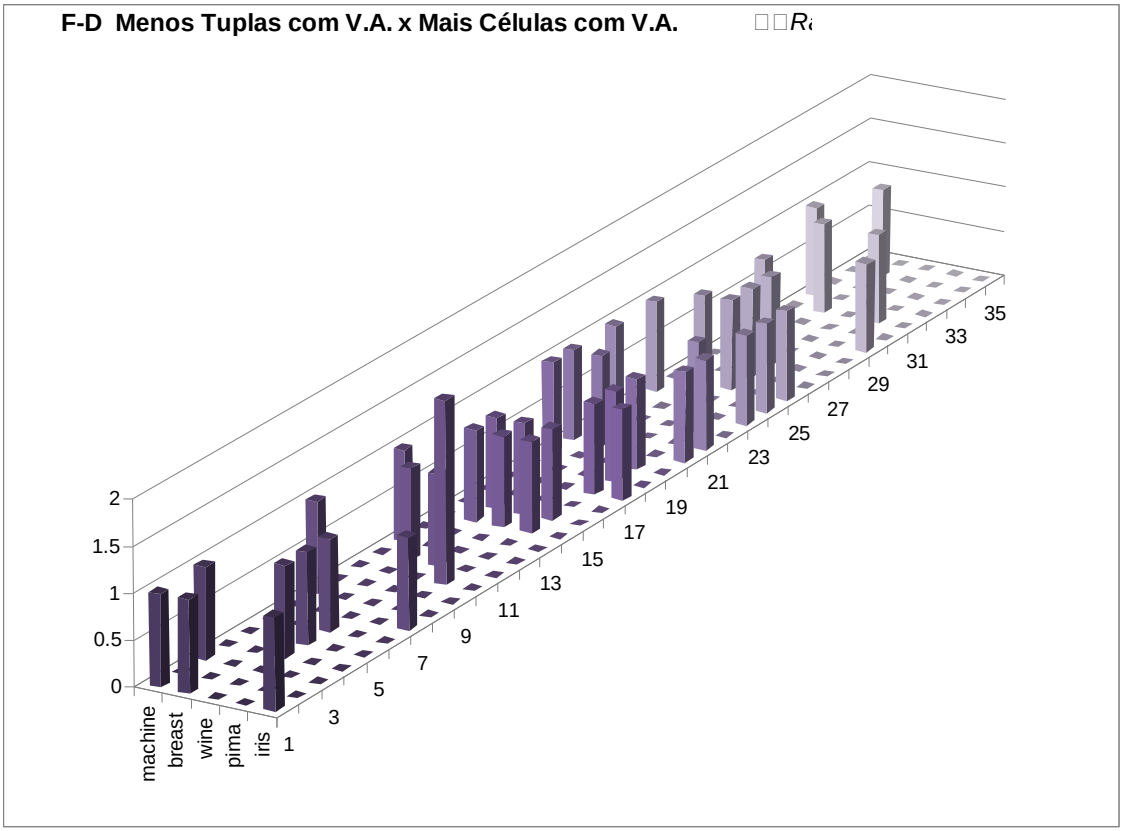


Gráfico 5.52- Ranking da ordenação (F,D) nas Bases
Critério de Ordenação dos Grupos (Externa): F – Menos Tuplas com Valores Ausentes
Critério de Ordenação de Atributos(Interna):D – Mais Células com Valores Ausentes

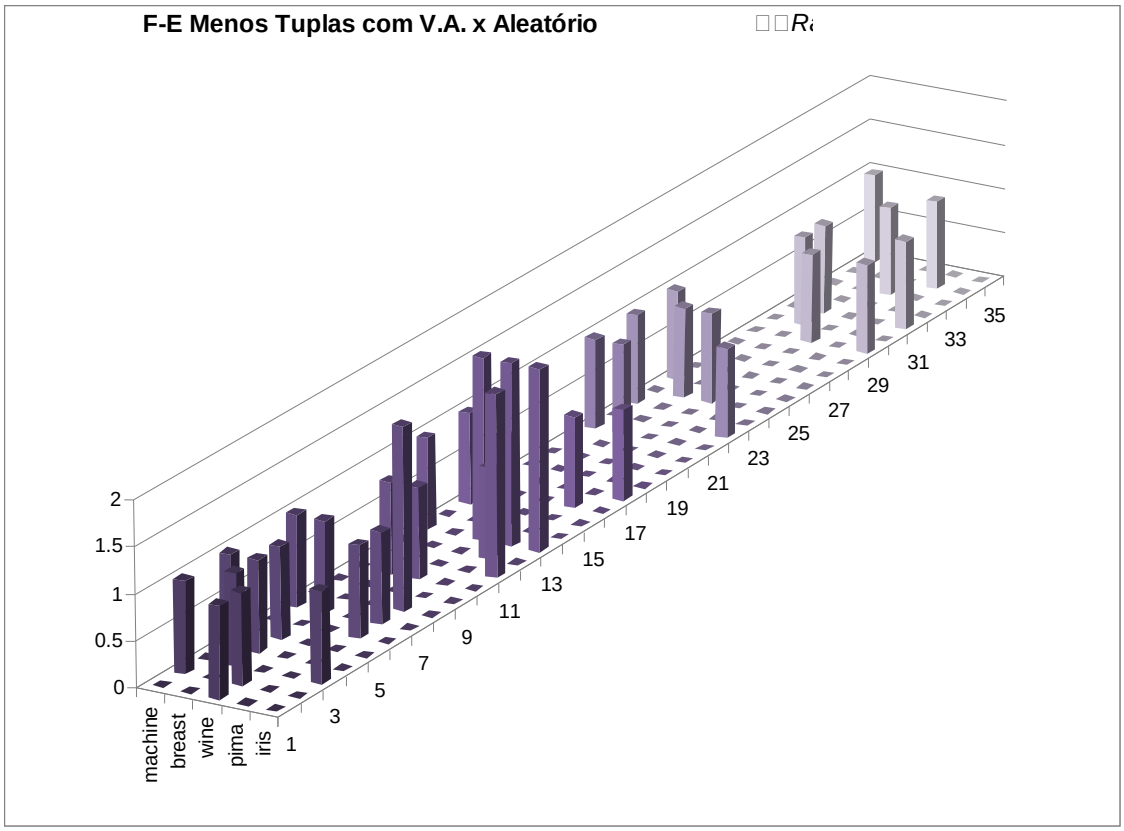


Gráfico 5.53- Ranking da ordenação (F,E) nas Bases
Critério de Ordenação dos Grupos (Externa): F – Menos Tuplas com Valores Ausentes
Critério de Ordenação de Atributos(Interna):E – Aleatório

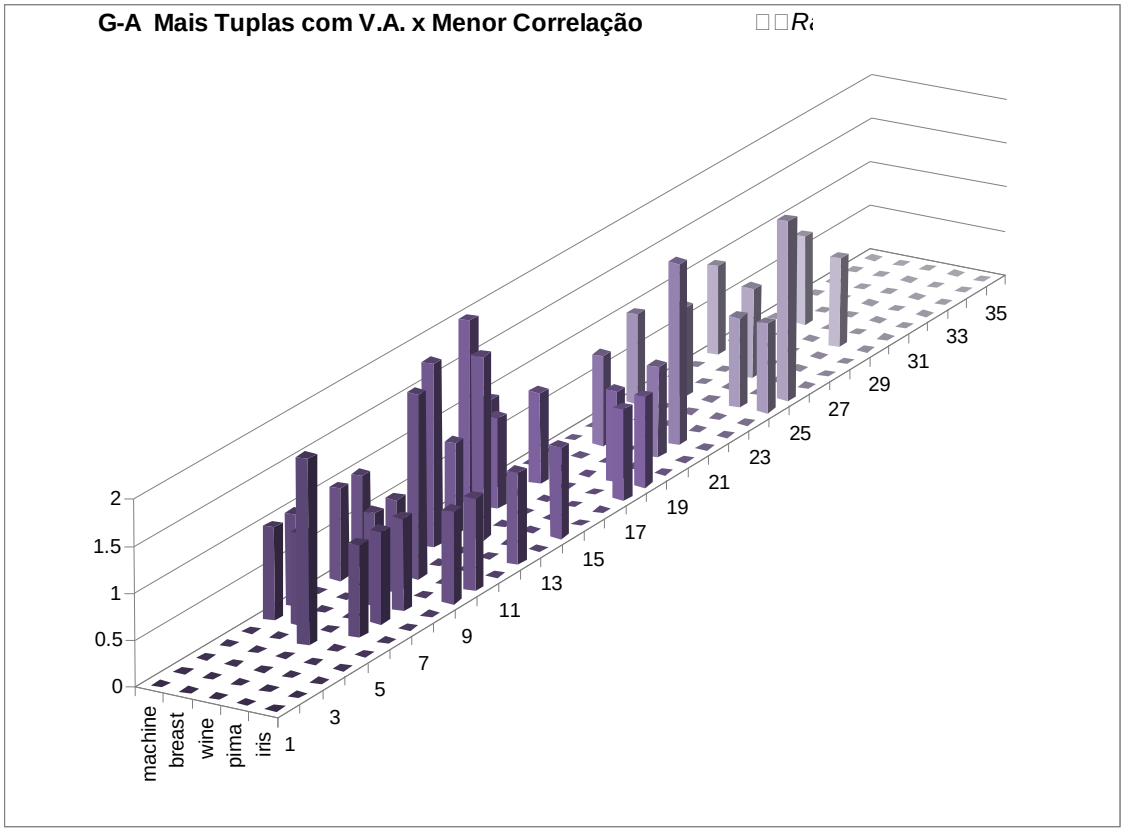


Gráfico 5.54- Ranking da ordenação (G,A) nas Bases
Critério de Ordenação dos Grupos (Externa): G – Mais Tuplas com Valores Ausentes
Critério de Ordenação de Atributos (Interna): A - Menor Correlação.

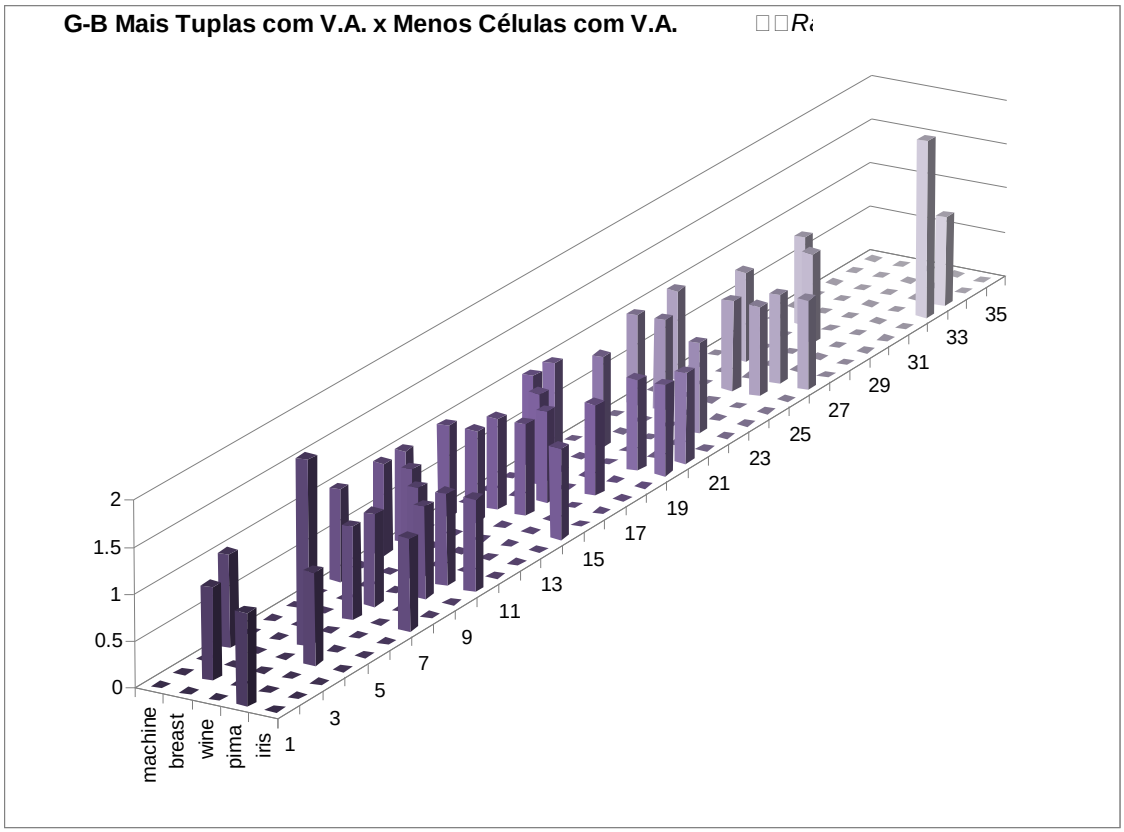


Gráfico 5.55- Ranking da ordenação (G,B) nas Bases
Critério de Ordenação dos Grupos (Externa): G – Mais Tuplas com Valores Ausentes
Critério de Ordenação de Atributos(Interna): B – Menos Células com Valores Ausentes.

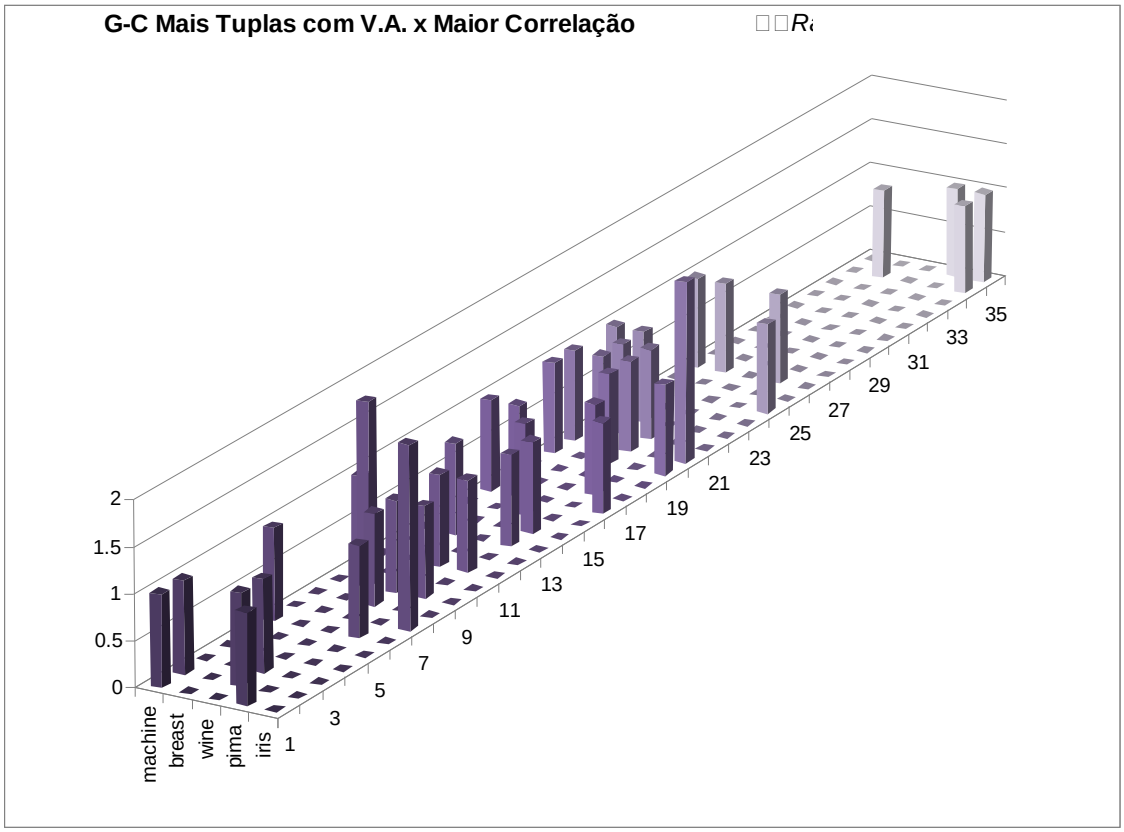


Gráfico 5.56- Ranking da ordenação (G,C) nas Bases
Critério de Ordenação dos Grupos (Externa):G – Mais Tuplas com Valores Ausente
Critério de Ordenação de Atributos(Interna): C – Maior Correlação.

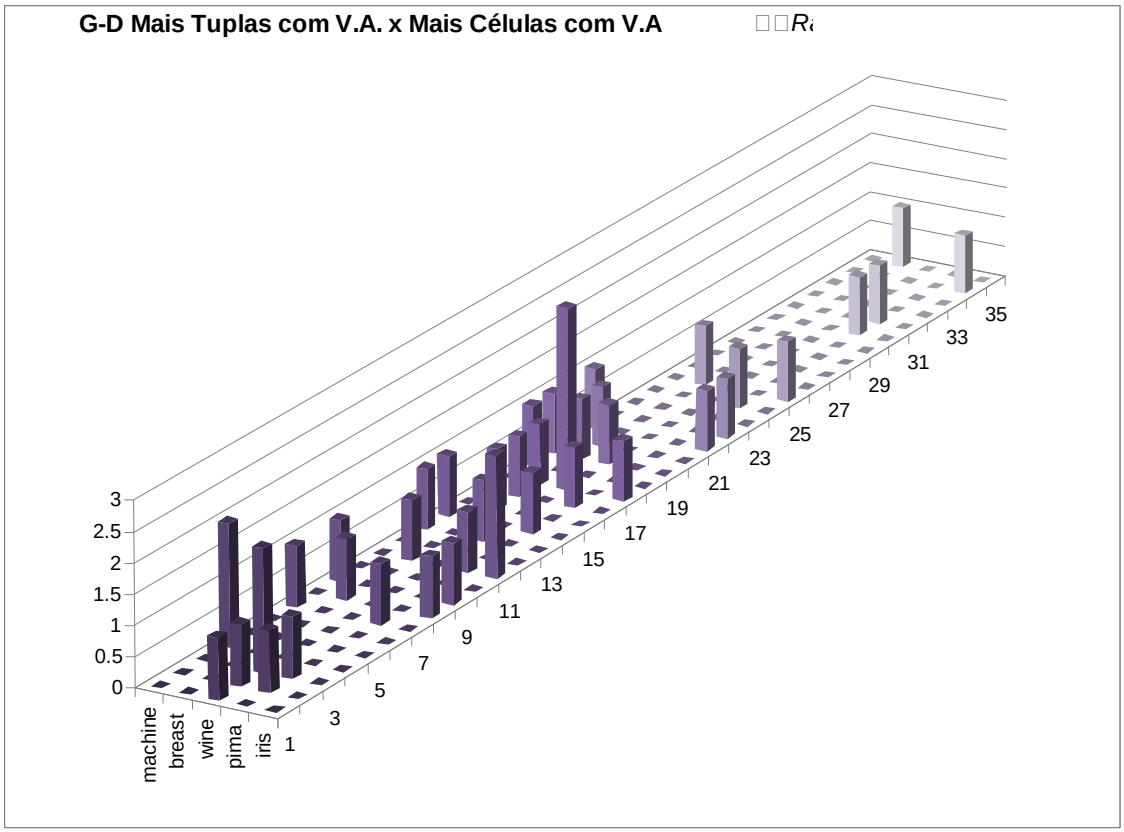


Gráfico 5.57- Ranking da ordenação (G,D) nas Bases
Critério de Ordenação dos Grupos (Externa): G – Mais Tuplas com Valores Ausentes
Critério de Ordenação de Atributos(Interna):D – Mais Células com Valores Ausentes

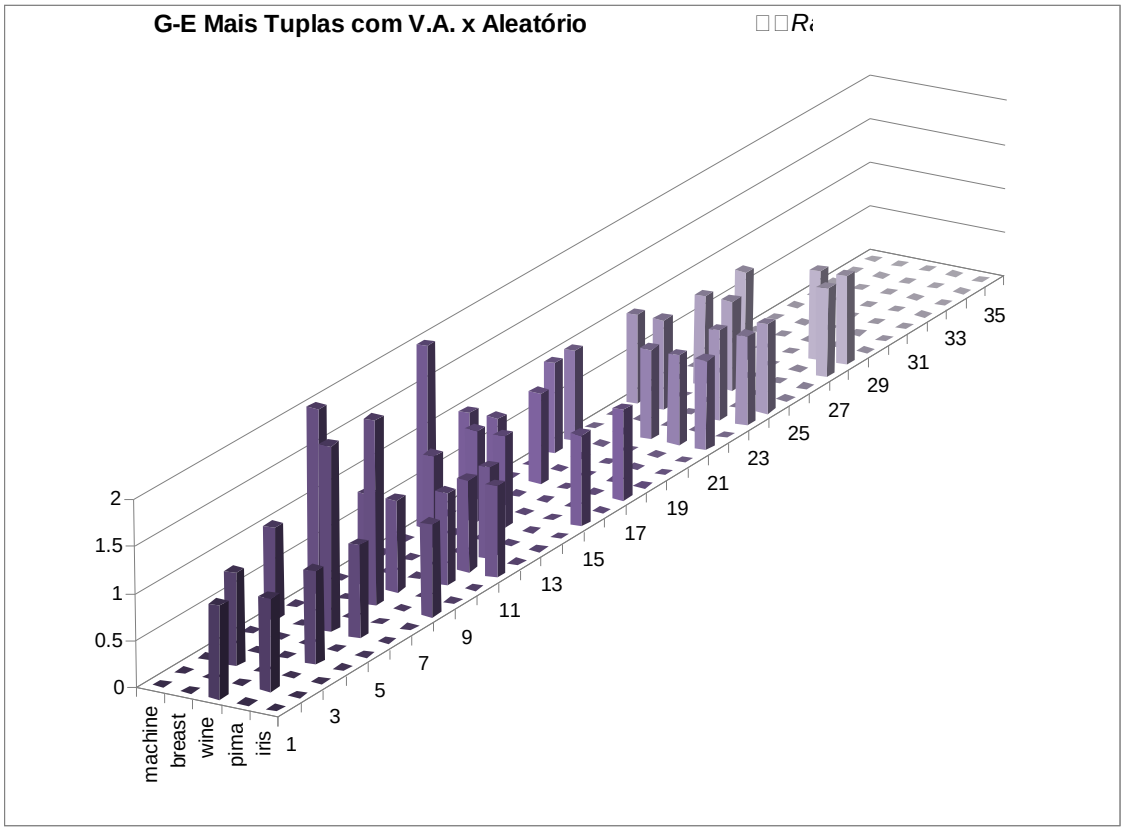


Gráfico 5.58- Ranking da ordenação (G,E) nas Bases
Critério de Ordenação dos Grupos (Externa): G – Mais Tuplas com Valores Ausentes
Critério de Ordenação de Atributos(Interna):E – Aleatório

5.3.8 Análise quanto à classificação das ordenações

Observa-se:

- A ordem (a) *fieldLessMissing* (menos células com valores ausentes) tem tendência mediana inferior
- A ordem (b) *fieldMoreMissing* (mais células com valores ausentes) tem tendência mediana, mas é sensível à ordenação interna que pode torná-la inclusiva.
- A ordem (c) *fieldPerTupleLessMissing* (menos células com valores ausentes) tem tendência frontal, com alta concentração de primeiros lugares, mas sofre interferência da ordem interna.
- A ordem (d) *fieldPerTupleLessMissing* (mais células com valores ausentes) tem tendência mediana e não sofre muita interferência da ordem interna
- A ordem (e) *noSort* (aleatória) tem alta concentração nos últimos lugarest
- A ordem (f) *TupleLessMissing* (menos tuplas com valores ausentes) tem tendência mediana frontal mas sofre interferência da ordem interna.
- A ordem (g) *TupleMoreMissing* (mais tuplas com valores ausentes) tem tendência mediana frontal, e é pouco influenciada pela ordem interna.

As ordens internas (a) *lessCorrelation* (menor correlação) e (d) *moreMissing* mais células com valores ausentes) são as que mais interferem na ordenação da ordem externa.

5.3.9 Classificação do desempenho e estabilidade das ordenações

Partindo dos gráficos acima se pode obter alguns resumos de desempenho. Para resumir o desempenho e a estabilidade de uma configuração frente às demais, observou-se o comportamento de dois modos e refletidos nos gráficos 5.59 a 5.62:

- **Participação nos 15 primeiros lugares**: obtida pela quantidade de vezes que a configuração ficou entre os quinze primeiros lugares, independente da base.
- **Estabilidade de desempenho**: o foco é observar a relação que a configuração está entre os primeiros lugares e entre os últimos lugares. Toda vez que a

configuração acaba nas últimas 15 posições ela é penalizada na mesma proporção que é recompensada quando se posiciona nos primeiros 15 lugares. Por isso, este índice de comparação é o resultado da subtração do número de vezes que a configuração esteve entre os quinze primeiros pelo número de vezes que a configuração esteve entre os últimos 15 lugares.

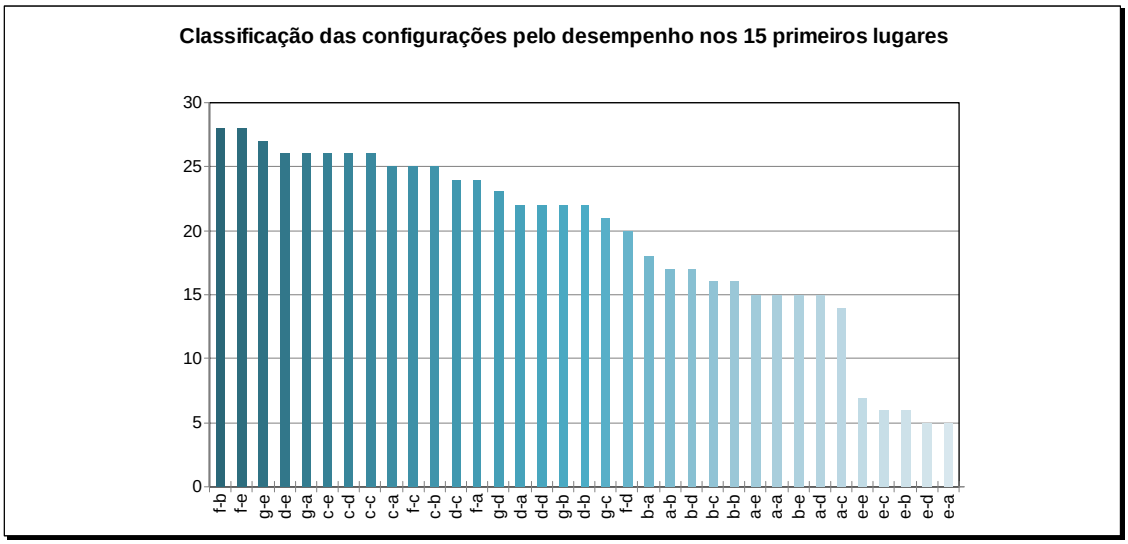


Gráfico 5.59- Ranking das ordenações considerando os 15 primeiros lugares

.

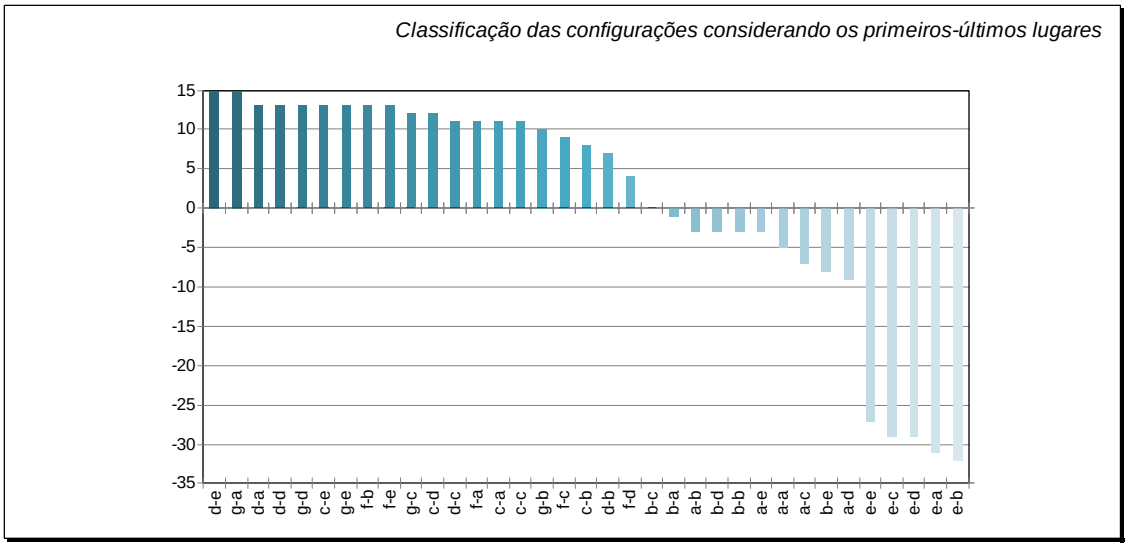


Gráfico 5.60- Ranking das ordenações considerando a estabilidade do desempenho

.

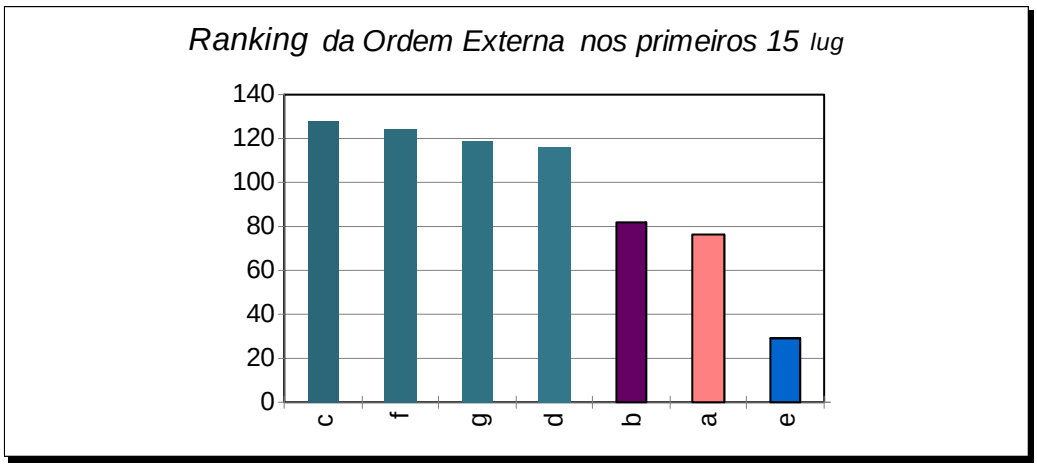


Gráfico 5.21 *Ranking da ordenação dos grupos os 15 primeiros lugares*

.

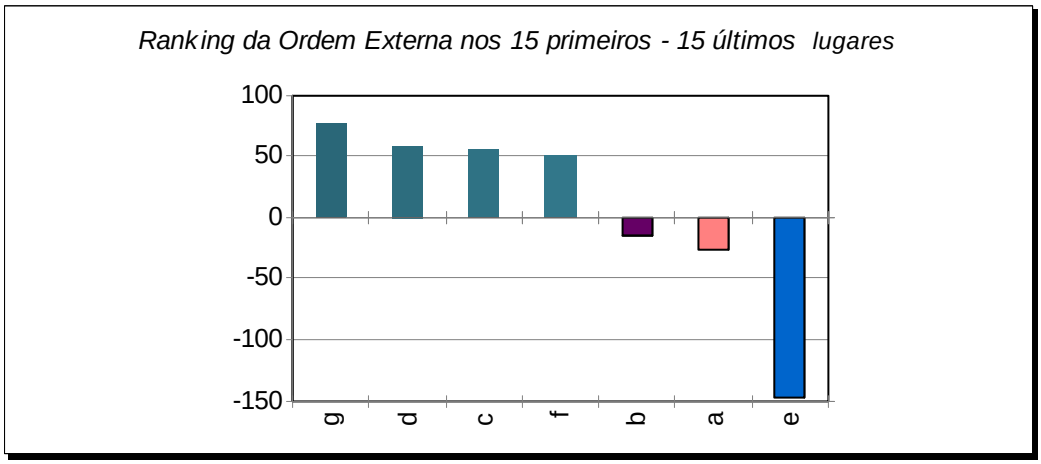


Gráfico 5.60- Ranking da ordenação dos grupos considerando a estabilidade do desempenho

5.3.10 Análise quanto ao desempenho e estabilidade das ordenações

Confirmando os resultados obtidos na análise do comportamento apenas pelo erro mínimo, a ordem externa (c) *FieldperTupleLessMissing* obteve os melhores resultados nos quinze primeiros lugares, porém quando penalizada por ocupar as últimas posições, esta ordem caiu para a terceira posição, demonstrando uma certa instabilidade de comportamento, quando então a ordem (g) *tupleMoreMissing* assume a liderança. Este critério torna a quantidade de registros restaurados o fator mais relevante na qualidade da imputação. O critério tupleMoreMissing já era vencedor para a base *Wine* e uma explicação adicional pode ser o fato de em algumas versões de bases , com os percentuais mais altos de ausência, se ter apenas dois casos completos inicialmente. No entanto, as diferenças entre os três primeiros critérios, (g) *TupleMoreMissing*, (d) *fieldPerTupleMoreMissing* (c) *fieldPerTupleLessMissing*, não é acentuada, sendo que os dois últimos consideram também a morfologia da ausência. Conclui-se então, que o

processo busca um balanceamento entre a complexidade do problema de imputação(o quão difícil é encontrar um bom valor para uma determinada célula) e a quantidade de registros restaurados que podem ser devolvidos à base para auxiliar na imputação dos próximos elementos. Observando os critérios com pior desempenho, deduz-se que a seleção dos grupos pela quantidade de células ausentes no grupo não produz um resultado satisfatório. Pela inegável última posição da ordenação aleatória, pode-se deduzir que a ordem interfere positivamente na qualidade da imputação, não podendo ser desprezada.

Em termos de estabilidade, considerando também a Ordem Interna, a Ordem Externa Vencedora é (d) *fieldPerTupleMoreMissing.*

CAPÍTULO 6

CONSIDERAÇÕES FINAIS

6.1 Resumo do Trabalho

Nesta tese, destacou-se a importância da complementação de dados ausentes em bases de dados no processo de KDD. Dados ausentes podem ser extremamente prejudiciais aos processos que as manipulam. Para amenizar estas conseqüências, por vezes desastrosas, foi proposta uma nova abordagem para imputação multivariada: a Imputação em Cascata que trabalha com tabelas compostas por atributos numéricos e em cujas tuplas pode haver vários atributos com valores ausentes simultaneamente.

Nesta abordagem híbrida, a tarefa de agrupamento precede o processo de imputação. Os casos incompletos são distribuídos em grupos considerando como critério de alocação o conceito de morfologia da ausência neles existentes. A morfologia de ausência, conceito aqui proposto para descrever a distribuição de valores presentes e ausentes nos atributos de um conjunto de casos, apresentou um comportamento muito satisfatório. O objetivo e o critério do agrupamento foram inéditos, pois os trabalhos existentes na literatura, aos quais se teve acesso, utilizam critérios estatísticos, nebulosos ou da teoria de informação, mas não analisam a distribuição espacial da ausência. O foco usual do agrupamento é dividir o conjunto original para que o processo de imputação seja influenciado apenas pelos objetos do conjunto de dados que possuam alguma relação com aqueles incompletos. Foi proposta também a binarização dos casos, para que os valores dos atributos não influenciem o método de agrupamento. Dado o viés em Inteligência Computacional, para agrupar os casos binarizados, utiliza-se uma rede SOM (s*elf-organizing map).* Após o agrupamento, a restauração dos casos foi realizada pela imputação seqüencial com realimentação de valores. Para regressão de cada célula ausente, utilizou-se a média dos valores dos atributos dos casos selecionados pelo algoritmo *k*-vizinhos mais próximos (k-NN).

Nesta abordagem, houve a possibilidade de analisar outras variáveis que não apenas o desempenho do método tais como: topologias da rede, número de vizinhos, critérios de distância, critérios de ordenação entre grupos e intra-grupo. Para isso, foi

construída uma plataforma, *workflow-like*, em Java, que gera, segundo parâmetros definidos, a seqüência de experimentos desejada.

A análise da ordem de imputação dos atributos no processo de complementação de dados não consta como objeto de pesquisa nos trabalhos da área aos quais se teve acesso. Deste modo, algumas ordens, dentre as muitas possíveis, foram propostas e analisadas em relação ao erro introduzido nesta tarefa.

A questão da alteração da correlação pelo uso dos valores previamente estimados na imputação dos primeiros valores, também foi alvo de estudo, pois embora esta seja uma afirmação na área, há poucos trabalhos que analisam e demonstram o bias da correlação após a aplicação dos métodos de complementação dos dados.

Resultados foram gerados em cinco bases do repositório de aprendizado de máquina da Universidade da Califórnia, Irvine. Este repositório possui diversas bases de dados com as mais diferentes características, e serve como *benchmark* para diversos trabalhos na área de descoberta de conhecimento de bases de dados. As bases *Íris Plants, Pima Indians, Wisconsin Breast Câncer, Wine e Hardware Machine* foram escolhidas para os experimentos, pois são formadas por casos originalmente completos, condição necessária para a avaliação. Todos os atributos utilizados, à exceção da classe, são numéricos. A ausência dos dados foi gerada artificialmente, adotando-se o mecanismo de ausência completamente aleatório (MCAR - *Missing Completely At Random*), com diferentes percentuais de ausência. Estes índices variaram de 10% a 30%, com saltos de 10%. Para cada base foram criadas três versões sujas para cada um dos percentuais citados, totalizando 45 bases que resultaram em 252.000 instâncias de teste. Os resultados gerados foram comparados com os gerados pela imputação seqüencial sem o prévio agrupamento, com e sem reutilização de valores. Algumas considerações sobre os parâmetros ajustáveis nos algoritmos também foram tecidas.

6.2 Contribuições da Tese

Esta tese tem como contribuição os seguintes pontos para a área de complementação de dados:

1. Proposta de um método híbrido de imputação de dados ausentes em bases de dados: a Imputação em Cascata, baseada no conceito de morfologia da ausência e com a utilização de tarefas do processo de Descoberta de Conhecimento em

Bases de Dados (etapa de pré-processamento de dados, o agrupamento, para posterior imputação)

2. Os experimentos gerados e persistidos em um banco de dados podem ser minerados para que associações e heurísticas sejam descobertas, de modo a nortear a condução do processo de complementação de valores ausentes.
3. A possibilidade de analisar outras variáveis que não apenas o desempenho do método, tais como: topologias da rede, número de vizinhos, critérios de distância, critérios de ordenação entre grupos e intra-grupo.
4. A avaliação do impacto da ordem no processo de imputação seqüencial.
5. Estudar a utilização da realimentação, nos contextos experimentados, no que diz respeito ao aumento do bias natural da imputação.
6. A verificação através dos experimentos realizados nesta tese, da observação de SOARES (2007), que as estratégias compostas produzem dados a serem imputados de melhor qualidade (com um menor índice de erro);
7. A observação, também baseado no escopo dos testes feitos, de que a distância de Mahalanobis não apresenta o comportamento relatado na literatura.
8. A constatação pelos resultados dos testes que, caso os grupos estejam bem construídos, a ordem de imputação dos atributos (a Ordem Interna) tem pouco impacto na qualidade do processo como um todo.
9. Ao implementar o método de Imputação em Cascata, sobre um *framework workflow-like*, permite que a troca de componentes para suas tarefas possam ser selecionados e testados sem esforço computacional e principalmente, sem alteração da sua estrutura principal .

6.3 Trabalhos Futuros

Extensões possíveis à abordagem proposta, bem como idéias não concretizadas que surgiram ao longo do desenvolvimento desta tese são citadas a seguir.

→ Na fase de pré-processamento:

- Avaliar a correlação dos atributos de forma mais ampla, permitindo que se utilize correlação envolvendo NxM atributos.

→ Na fase de agrupamento:

- Avaliar outros métodos de agrupamento auto-organizáveis, hierárquicos ou mesmo de forma determinística pelo padrão de ausência. Uma vez que a topologia do SOM que apresentou melhor desempenho permite a representação de todas as 2^K possíveis combinações de ausência, pode-se avaliar uma primeira distribuição por padrão de ausência e um reagrupamento a posteriori caso necessário.
- Associar descritores que representem algumas características dos grupos, visando a determinação de métodos distintos de imputação. Deste modo, para grupos complexos, métodos mais sofisticados de imputação podem ser aplicados.
- Incluir uma fase de análise dos agrupamentos gerados pela utilização de índices que permitam avaliar a qualidade do agrupamento principalmente no que diz respeito ao grau de coesão de seus elementos e a densidade dos mesmos, pois os testes indicam uma dependência da qualidade da imputação com estes fatores. Grupos com valores não aceitáveis deveriam ser reagrupados, inclusive por métodos distintos de agrupamentos.
- Utilizar outros métodos de particionamento automático dos grupos, inclusive determinando uma quantidade mínima e máxima de elementos em cada grupo pois quando há grupos com poucos elementos e não há casos completos suficientes a imputação dos valores do grupo pode ficar comprometida.
- Pesquisar e identificar razões para alguns agrupamentos ter comportamento tão fora do padrão

→ Na fase de imputação sequencial:

- Utilizar outras métricas de similaridade, que diferenciem os valores imputados dos originais. Em particular, acredita-se que métricas que diferenciem e ponderem a participação de casos restaurados devam ser testadas
- Usar os protótipos da SOM que gerou o grupo como valor inicial dos valores ausentes, substituindo o uso de casos incompletos

- Determinar o k do k-NN usando a heurística k=$\sqrt{N}$ ou métricas de análise de agrupamento para cada atributo
- Investigar a morfologia de ausência como heurística para a determinação de algoritmos de imputação

→ Na fase de imputação simples são extensões visualizadas:

- Utilização de outros métodos para estimar o valor ausente, inclusive algoritmos genéticos ou os centróides da SOM e que ponderem a participação dos valores imputados e dos originais.
- Distinção do valor imputado do valor original para a regressão do novo valor, incluindo lógica fuzzy e ponderações que considerem a credibilidade dos valores que participam da estimativa do novo valores

→ Para a análise da eficiência da abordagem deve-se:

- Aplicar em bases compostas por atributos categóricos e bases mistas.
- Avaliar seu desempenho em grandes bases.
- Utilizar outros percentuais de ausência
- Utilizar os mecanismos de ausência NMAR e MAR
- Medir a acurácia dos valores imputados pela reclassificação dos casos, em bases que possuam um atributo-classe.
- Analisar a instabilidade de alguns agrupamentos.

→ Na ordenação dos grupos e atributos pode-se:

- Considerar outros critérios de ordem Externas e Internas que considerem correlações múltiplas, entropia, quantidade de ausência nos atributos considerados como principais. o grau de coesão ou compactação do segmento.
- Avaliar a relação da ordem dos segmentos com o desempenho não satisfatório de alguns grupos.

REFERÊNCIAS

AAMODT, A., PLAZA, E., 1994, “Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches.” AI Communications.

ADRIAANS, P., ZANTINGE, D.,1996, “ Data Mining”. Addison-Wesley.

AGRAWAL, R. IMIELINSKI, T., SWAMI, 1993, “A. Mining Association rules between sets of itens in large databases.”, Proc. 1993 Int. Conf. Management of Data (SIGMOD-93), 207-216.

AHA, D.W., BANKERT, R.L., 1995, “A comparative evaluation of sequential feature selection algorithms.”, Proc. 5th Int. Workshop on Artif. Intel. And Statistics, 1-7. Ft. Lauderdale, FL.

AHA, D. W., KIBLER, D., ALBERT, M., 1991, “Instance-based Learning Algorithms”, *Machine Learning*, v. 6, pp. 37-66.

ALEXANDER, S., 2003, “ Introduction to Workflows and Use of Workflows in Grids and Grid Portals.” Indiana University. Disponível em: www.extreme.indiana.edu/swf-survey/IntroductionToWorkflowsInGridsAndPortals_GGF9_2003.ppt

ALLISON, P. D., 2000, “Multiple Imputation for Missing Data: A Cautionary Tale”, *Sociological Methods & Research*, v. 28, pp. 301-309.

ALLISON, P. D., 2001, “Missing Data.”, Sage Publications.

ARABIE, P., LAWRENCE, J., 1996, “ An overview of combinatorial data analysis” , In P. Arabie, L. J. Hubert, and G. De Soete, editors, Clustering and Classification, pages 5-63. World Scientific Puc., New Jersey.

ARIMA, C., OKAMOTO, M., HANAI, T., 2003, “Gene Expression Analysis Using Fuzzy K-Means Clustering”. Genome Informatics, 14:334–335.

AUSTIN, P. C., ESCOBAR, M. D., 2005, “Bayesian modeling of missing data in clinical research”, Computational Statistics & Data Analysis, v. 49, pp. 821-836.

AURÉLIO, M., VELLASCO, M., LOPES, C.H.,1999, “Descoberta de Conhecimento e Mineração de Dados”, ICA – Lab. Inteligência Computacional Aplicada , DEE, PUC–Rio.

BALA, J., HUANG, J., VAFAIE, H., DEJONG, K., WECHSLER, H., 1995, “Hybrid learning using genetic algorithms and decision trees for pattern classification.”, Proc. 14th Int. Joint Conf. AI (IJCAI-95), 719-724.

BATISTA, G. E. A. P. A., MONARD, M. C., 2001, “A Study of K-Nearest Neighbour as a Model-Based Method to Treat Missing Data”. In: Proceedings of the Argentine Symposium on Artificial Intelligence (ASAI’01), pp. 1–9, Buenos Aires, Argentina.

BATISTA, G. E. A. P. A., MONARD, M. C., 2003a, “An Analysis of Four Missing Data Treatment Methods for Supervised Learning”, Applied Artificial Intelligence, v. 17, n. 5 (May-Jun), pp. 519-533.

BAYENS T., 2004, "The State of Workflow", The Server Side. Disponível em: http://www.theserverside.com/tt/articles/article.tss?l=Workflow

BERRY, M. J. LINOFF, G., 1997, “Data mining techniques: For marketing, sales, and customer support.” New York: John Wiley and Sons.

BERRY, M. J.; LINOFF, G., 200, “Mastering data mining.”, New York: Wiley Computer Publishing.

BEZERRA, E., 2006, Agrupamento Semi-supervisionado de Documentos XML, Tese de Doutorado, COPPE-UFRJ.

BEZERRA, E., MATTOSO, M., XEXEO, G., 2006, “Semi-Supervised Clustering of XML Documents: Getting the Most from Structural Information.”, Data Engineering Workshops, Atlanta, GA, USA.

BEZDEK, J.C. , 1998, “Some new indexes of cluster validity”, IEEE Trans. Syst., Man, Cybern. B, vol. 28, pp. 301–315.

BOLL, E. M., ST. CLAIR, D. C., 1995, “Analysis of rule sets generated by the CN2,ID3, and multiple convergence symbolic learning methods”. In: Proceedings of the 1995 ACM 23rd annual conference on Computer science, pp. 48 - 55, Tennessee, USA

BRAZMA, A., VILO, J., 2000. Gene expression data analysis. FEBS Letters, 480(1):17–24.

BREIMAN, L., 1996, Bagging Predictors, Technical Report n. 421, Department of Statistics, University of California, Berkeley, California.

BRIN, S., MOTWANI, R., ULLMAN, J. D, TSUR, S., 1997, "Dynamic itemset counting and implication rules for market basket data". In: Proceedings of the ACM

BROWN, M. L., KROS, J. F., 2003, “The impact of missing data on data mining”. In: Wang, J. (Author), Data mining: opportunities and challenges, 1 ed., chapter VII, pp.174-198, IGI Publishing , Hershey, PA, USA.

CARTWRIGHT, M. H., SHEPPERD, M. J., SONG, Q., 2003, “Dealing with missing software project data”. In: Proceedings of the 9th International Symposium on Software Metrics, pp. 154 – 165, Sep.

CARVALHO, L. A. V., 2001, “Datamining: A Mineração de Dados no Marketing, Medicina, Economia, Engenharia e Administração. “, 2 ed, São Paulo, SP, Editora Érica.

CARVALHO, L. A. V., FISZMAN, A., FERREIRA, N. C., 2001, “A Theoretical Model for Autism”, Journal of Theoretical Medicine, v. 25, pp. 123-140, USA.

CASTRO, F.C.C.; CASTRO, M.C.F., 2001, “Redes Neurais Artificiais”, Porto Alegre: PUCRS, 2001. Apostila para fins didáticos. Disponível em: <http://diana.ee.pucrs.br/~decastro/RNA_hp/RNA.html> . Acesso em: 14 Sep. 2004.

CASTANEDA, R., FERLIN, C.,GOLDSCHMIDTH, R., SOARES, J.A., CARVALHO, L.A., CHOREN, R. , 2008, “Aprimorando Processos de Imputação Multivariada de Dados com Workflows”. Aceito para publicação no SBBD 2008.

CAVALCANTI M.C., TARGINO R., BAIÃO F., RÖSSLE S., BISCH P., PIRES P., CAMPOS M.L., MATTOSO M., 2005, “Managing Structural Genomic Workflows using Web Services”, COPPE-Sistemas, UFRJ

CHAVES, E., NAVARRO, G. BAEZA-YATES, R., MARROQUIN, J.L.,2001, “Searching in metric spaces”. ACM Computing Surveys, 33(3):273–321.

CHEN, J., SHAO, J., 2000, “Nearest Neighbor Imputation for Survey Data”, *Journal of Official Statistics*, v. 16, n. 2, pp. 113-131.

CIOS, K. J.; PEDRYCZ , W.; SWINIARSKI, R. W., 1998, “Data mining: Methods for knowledge discovery. “, Boston: Kluwer.

CLAUDIO, 2003, “Incremental Tree-Based Missing Data Imputation with Lexicographic Ordering.”, Computing Science and Statistics, 35

COSTA, J.A.F., NETTO, M.L.A.., 1999, " Classificação Automática e Análise de Dados por Redes Neurais Auto-Organizáveis.", Tese de Doutorado, Unicamp, SP.

COSTA, J.A.F., NETTO, M.L.A. , 2003. "Segmentação automática de mapas de Kohonen.", Congresso Brasileiro de Automática, Natal, RN, pp. 1607-1613.

COSTA, J.A.F., NETTO, M.L.A., 2003, "Segmentação do SOM Baseada em Particionamento de Grafos.", Anais do VI Congresso Brasileiro de Redes Neurais, São Paulo, pp. 451-456.

CRÉMILLEUX, B., RAGEL, A., BOSSON, J. L., 1999, "An Interactive and Understandable Method to Treat Missing Values: Application to a Medical Data Set". In: Proceedings of the 5th International Conference on Information Systems Analysis and Synthesis (ISAS/SCI 99), pp. 137-144.

DASARATHY, B., 1990, "Nearest Neighbor (NN) norms: NN pattern classification techniques", 1 ed., IEEE Computer Society Press, Los Alamitos.

DAVIES, D. L.; BOULDIN, D. W. A,1979, "Cluster separation measure." IEEE Transactions on Pattern Analysis and Machine Intelligence, v. PAMI-1, p. 224–227.

DEMPSTER, A. P., LAIRD, N. M., RUBIN, D. B., 1977, "Maximum Likelihood from Incomplete Data via the EM Algorithm", Journal of the Royal Statistical Society, Series B (Methodological), v. 39, n. 1, pp. 1-38.

DENG, J., 1989, "Introduction to Grey System", The Journal of Grey System, v. 1, pp. 1-24.

DIDAY, E., SIMON, J.C., 1980, "Clustering Analysis", In Du, K.S.(ed), Digital pattern recognition, Springer-Verlag.

DILLY, R., 1995, "Data mining:An introduction.", Parallel Computer Centre. Queen's University of Belfast.

DI ZIO, M., SCANU M., COPPOLA, L., LUZI, O., PONTI, 2004 A., "Bayesian Networks for Imputation." Journal of the Royal Statistical Society A, 167, Part 2, p. 309-322.

DOUGHERTY, J., KOHAVI, R., SAHAMI, M., 1995, "Supervised and unsupervised discretization of continuos features.", Proc. 12th Int. Conf. Machine Learning, 194-202.

EIN-DOR, FELDMESSER, 2002, ”Computer Hardware Dataset.” CACM 4/87, pp 308-317

ESTIVILL-CASTRO, V., 2002, “Why so many clustering algorithms - a position paper.“, SIGKDD Explorations, 4 (1), 65–75.

ERWIN, E., OBERMAYER, K., SCHULTEN, K.,1992, “Self-organizing maps: ordering, convergence properties and energy functions.”, Biological Cybernetics, Springer.

FAYYAD, U., PIATETSKY-SHAPIRO, G., SMYTH, P., 1996a, “From Data Mining to Knowledge Discovery in Databases”, AI Magazine, American Association for Artificial Intelligence, pp. 37-54.

FAYYAD, U., PIATETSKY-SHAPIRO, G., SMYTH, P., 1996b, “The KDD Process for Extracting Useful Knowledge from Volumes of Data”, Communications of the ACM, v. 39, n. 11, pp. 27-34, Nov.

FAUSETT, L.,1994, Fundamentals of neural networks: architectures, algorithms, and applications. Upper Saddle River, NJ: Prentice Hall, p 459.

FATTORE, M., ARRIGO, P., 2005, “Knowledge discovery and system biology in molecular medicine: An application on neurodegenerative diseases.” In Silico Biology, 5(2):199–208.

FESSANT F., MIDENET S., 2002, “Self-Organising Map for Data Imputation and Correction in Surveys”, Neural Comput & Applic, vol 10, pp 300–310 Springer-Verlag London.

FORD, B. L., 1983, “An Overview of Hot-Deck Procedures”. In: Madow, W. G., Olkin, I. (auth.), Rubin, D. B. (ed.), Incomplete Data in Sample Surveys, 1 ed., vol. 2, Part IV, Chapter 14, pp. 185-207, Academic Press.

FREITAS, A.A., LAVINGTON, S.H., 1998, “Mining Very Large Databases with Parallel Processing”, Kluwer.

FREITAS, A. A., 2002, “Data Mining and Knowledge Discovery with Evolutionary Algorithms”, 1 ed. New York, Springer-Verlag Heidelberg.

FULLER, W. A., KIM, J. K., 2001, “Hot Deck Imputation for the Response Model”, Survey Methodology, v. 31, n. 2, pp. 139-149.

GAALOUL, W., ALAOUI, S., BAINA, K., GODART, C., 2005, ”Mining workflow patterns through event-data analysis”, In SAINT-W ’05: Proceedings of the 2005 Symposium on Applications and the Internet Workshops, pages 226–229, Washington, DC, USA. IEEE Computer Society.

GELMAN, A., RAGHUNATHAN, T., 2001, “Conditionally specified distributions: An introduction. Statistical Science”, 16(3):268–269.

GELMAN, A., HILL, J.,2006, “Data Analysis Using Regression and Multilevel/Hierarchical Models”, Cambridge University Press.

GENEVA, U. N. S., 2000, “Comission and E. C. for Europe Glossary of terms on statistical data editing”, United Nations.

GIRAUDEL, J.-L.; LEK, S., 2001, “A comparison of self-organizing map algorithm and some conventional statistical methods for ecological community ordination”, Ecological Modelling, v.146, p.329-339.

GLEASON, T., STAELIN, R. 1974, “A proposal for handling missing data”, Psychometrika, 40(2):229–252.

GOLDSCHMIDT, R., PASSOS, E., 2005, “Data Mining: Um Guia Prático”, 1 ed, Editora Campus.

GONÇALVES, A.F., PANTOJA, C.S, 2005, “Imputação de Dados no Contexto de Mineração de Dados”, Universidade da Amanzônia, Centro de Ciências Exatas e Tecnologia, Curso de Computação e Informática.

GOWDA, K. C., DIDAY, E., 1992, “Symbolic Clustering Using a New Dissimilarity Measure”, IEEE Transactions on Systems, Man, and Cybernetics, v. 22, pp. 368-378.

GRAHAM, J. W., CUMSILLE, P. E., ELEK-FISK, E., 2002, “Methods for handling missing data”. In: Schinka, J. A., Velicer, W. F. (eds.). Research Methods in Psychology, v. 2 of Handbook of Psychology (I. B. Weiner, Editor-in-Chief), pp. 87-114, New York, John Wiley & Sons.

GRAHAM, J. W., DONALDSON, S. I., 1993, “Evaluating interventions with differential attrition: the importance of nonresponse mechanisms and use of follow-up data”, Journal of Applied Psychology, v. 78, n. 1, pp. 119-128.

GROTH, R., 1998, “Data mining: a hands-on approach for business professionals”. Prentice Hall, New Jersey.

GUHA, S., RASTOGI, R., SHIM, K., 1998, “CURE: An E.cient Clustering Algorithm for Large Databases”. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD-98), pp. 73–84, Seattle, WA, USA.

HÄKKINEN, E., 2001, “Design, Implementation and Evaluation of the Neural Data Analysis Environment”, PhD thesis, Diss. Jyväskylä Studies in Computing. University of Jyväskylä, Finland.

HALKIDI, M., BATISTAKIS, Y., VAZIRGIANNIS, M.,2002, “On Clustering Validation Techniques”, J. Intell. Inf. Syst., 17(2-3):107–145.

HAN, J.; KAMBER, M., 2001, “Data mining: Concepts and techniques”, San Francisco: Morgan Kaufmann.

HAND, D.; MANNILA, H.; SMYTH, P.,2001, “Principles of data mining”, Cambridge: MIT.

HARVEY, M., ARTHUR, C., 2004 , “Fitting Models to Biological Data Using Linear and

Nonlinear Regression”, Oxford University Press, Oxford.

HAYKIN, S., 1994,”Neural Networks: A Comprehensive Foundation”, Macmillan College Publishing Company, New York, NY.

HAYKIN, S., 1999, “Redes Neurais: Princípios e Prática”. 2ed. Porto Alegre, RS, Editora Bookman.

HOLLINGSWORTH, D., 2004, “The Workflow Reference Model: 10 Years On Fujitsu Services”, UK; Technical Committee Chair of WfMC. Disponível em: http://www.wfmc.org/standards/docs/Ref_Model_10_years_on_Hollingsworth.pdf

HOLT, J. D., CHUNG, S. M., 1999, “Efficient Mining of Association Rules in Text Databases”. In: Eight International Conference on Information and Knowledge Management (CIKM'99), Kansas City, USA, pp. 234-242, Nov.

HORTON N.J., LIPSITZ S.R., 2001, “Multiple Imputation in Practice: Comparision of Software Packages for Regression Models with Missing Variables”, *τ*he American Statistician, August 2001, Vol. 55, No. 3

HRUSCHKA, E. R., HRUSCHKA JR., E. R., EBECKEN, N. F. F., 2002, “Missing values prediction with K2”, Intelligent Data Analysis Journal, v. 6, n. 6, pp. 557-566.

HRUSCHKA, E. R., HRUSCHKA JR., E. R., EBECKEN, N. F. F., 2003a, “A Nearest-Neighbor Method as a Data Preparation Tool for a Clustering Genetic Algorithm”. In: Anais do 18o Simpósio Brasileiro de Banco de Dados (SBBD), pp. 319-327, Manaus, AM, Out.

HRUSCHKA, E. R., HRUSCHKA JR., E. R., EBECKEN, N. F. F., 2003b, “Evaluating a Nearest-Neighbor Method to Substitute Continuous Missing Values”. In: The 16th Australian Joint Conference on Artificial Intelligence - AI´03, 2003, Perth. Lecture Notes in Artificial Intelligence (LNAI 2903), v. 2903, pp. 723-734. Heidelberg, Springer-Verlag.

HRUSCHKA, E. R., HRUSCHKA JR., E. R., EBECKEN, N. F. F., 2005, “Missing Values Imputation for a Clustering Genetic Algorithm”. In: First International Conference on Natural Computation (ICNC´05), Changsha. Lecture Notes in Computer Science 3612 (Advances in Natural Computation). Berlin, Springer-Verlag Berlin Heidelberg, v. 3612, pp. 245-254.

HU, M., SALVUCCI, S., COHEN, M. P.,1998, “Evaluation of Some Popular Imputation Algorithms”, Proceedings of the Section on Survey Research Methods, American Statistical Association, 308-313

HUISMAN, M., 2000, “Imputation of Missing Item Responses: Some Simple Techniques”, Quality and Quantity, v. 34, n. 4, pp. 331-351, Nov.

INDURKHYA N., WEISS S. M., 1999, “Estimating Performance Gains for Voted Decision Trees”, In Intelligent Data Analysis (IDA), Disponível em http://www.research.ibm.com/dar/papers/pdf/weiss-98-1_with_cover.pdf).

JAIN, A. K., DUBES, R. C., 1988, “Algorithms for Clustering Data”, 1 ed. New Jersey, Prentice Hall College Division.

JAIN, A. K., MURTY, M. N., FLYNN, P. J., 1999, “Data Clustering: A Review”, ACM Computing Surveys, v. 31, n. 3 (Set), pp. 264-323.

JIANG, D., C. TANG, & A. ZHANG, 2003, ”Cluster analysis for gene expression data: A survey.”, IEEE Transactions on Knowledge and Data Engineering.

JOHNSON, R.A.; WICHERN, D.W., 2002, “ Applied multivariate statistical analysis”, 5ª edição, New Jersey: Prentice-Hall

JONES, M. P., 1996, “Indicator and stratification methods for missing explanatory variables in multiple linear regression”, Journal of the American Statistical Association, v. 91, n. 433, pp. 222-230, Mar.

JÖNSSON, P., WOHLIN, C., 2004, "An Evaluation of k-Nearest Neighbour Imputation Using Likert Data", In: Proceedings of the 10th IEEE International Symposium on Software Metrics (METRICS'04), pp. 108-118, Chicago, USA, Sep.

JÖNSSON, P., WOHLIN, C., 2006, “Benchmarking k-nearest neighbour imputation with homogeneous Likert data”, Empirical Software Engineering, v. 11, pp. 463-489.

JUNNINEN H., NISKA H., TUPPURAINEN K., RUUSKANEN J., KOLEHMAINEN M., 2004, “Methods for imputation of missing values in air quality data sets”, Atmospheric Environment 38 (2004) 2895–2907.

KARYPIS, G., KUMAR, V.,2000, “A comparison of document clustering techniques”. In: Proceedings of the KDD Workshop on Text Mining.

KASKI, S., 1997, “Data Exploration Using Self-Organizing Maps”, Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series, No. 82. Doctoral Thesis, Helsinki University of Technology

KASKI S., NIKKILÄ J., KOHONEN T., 1998, “Methods for interpreting a self-organized map in data analysis”, In Michel Verleysen, editor, Proceedings of ESANN’98, 6th European Symposium on Artificial Neural Networks, Bruges, April 22-24, pages 185-190. D-Facto, Brussels, Belgium.

KIBLER, D., AHA, D., 1988, “Instance-Based Prediction of Real-Valued Attributes”, Proceedings of the CSCSI (Canadian AI) Conference.

KIM, K.Y., 2004, “Reuse of imputed data in microarray analysis increases imputation efficiency”, BMC Bioinformatics, 5, 160.

KOHONEN, T., 1984, “Self-organization and Associative Memory”, Springer-Verlag, Berlin.

KOHONEN, T., KASKI ,S. , LAGUS K., SALOJÄRVI J., HONKELA J., PAATERO V.,

SAARELA A, 2000, “Self organization of a massive document collection”, IEEE Trans. Neural Networks, vol. 11..

KOHONEN, T., 1995, “Self-organizing maps”, Springer, 1995. Third Edition 2001.

KOIKKALAINEN, P.,1995, “Deterministic self-organizing maps”, in Proceedings of International Conference on Artifical Neural Networks (ICANN'95), páginas 63-68.

KRAAIJVELD M.A., MAO J. , JAIN A.K., 1995, “A nonlinear projection method based on Kohonen´s topology preserving maps”, IEEE Transactions on Neural Networks, 6(3):548-559.

LAKSHMINARAYAN, K., HARP, S. A., SAMAD, T., 1999, “Imputation of Missing Data in Industrial Databases”, Applied Intelligence, v. 11, n. 3 (Nov), pp. 259-275.

LEBART, L.; MORINEAU, A.; PIRON, M., 1995, “Statistique exploratories muldimensionnalle”, Paris, Dumond.

LEPKOWSKI, J., RAGHUNATHAN, T., SOLENBERGER, P., VAN HOEWYK, J. A ., 2001, “Multivariate technique for multiply imputing missing values using a sequence of regression models”, Statistics Canada, 27(1):85–95.

LEEUW, EDITH D., 2001, “Reducing Missing Data in Surveys: An Overview of Methods”, Quality & Quantity 35**:** 147–160, Kluwer Academic Publishers. Netherlands.

LITTLE, R., RUBIN, D., 1976, “Statistical analysis with missing data”, Technometrics, 45:364–365.

LITTLE, R.J.A., RUBIN, D.B., 1987, “Statistical Analysis with Missing Data”, John-Wiley and Sons, New York.

LITTLE, R., RUBIN, D., 2003, “Statistical analysis with missing data”, Technometrics, 45:364–365.

LIU, H., HUSSAIN, F., TAN, C. L., DASH, M., 2002, “Discretization: An Enabling Technique“, Data Mining and Knowledge Discovery, v. 6, pp. 393, Kluwer Academic Publishers.

MAGALHÃES, I. B., 2007, “Avaliação de redes bayesianas para imputação em variáveis qualitativas e quantitativas”, Tese de Doutorado, Departamento de Engenharia Mecatrônica e de Sistemas Mecânico, Escola Politécnica-USP.

MAGNANI, M., 2004, “Techniques for Dealing with Missing Data in Knowledge Discovery Tasks”. Obtido em Http://magnanim.web.cs.unibo.it/index.html em 15/01/2007.

MANGIAMELI, P., CHEN, S. K., EWEWST, D., 1996, ”A comparison of SOM neural network and hierarchical clustering methods”, European Journal of Operational Research, 93:402

MARRONE, P., 2007, “Java Object Oriented Neural Engine: The Complete Guide.”, Obtido em http://ufpr.dl.sourceforge.net/sourceforge/joone/JooneCompleteGuide.pdf.

MCQUEEN, J., 1967, “Some methods for classification and analysis of multivariate observations”. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297.

MERZ, C., MURPHY, P., 1998, “UCI repository of machine learning databases”, University of California, Irvine, Department of Information and Computer Sciences. http://www.ics.uci.edu/ mlearn/MLRepository.html.

MILLIGAN, G. W, COOPER, M. C., 1985, “An examination of procedures for determining the number of clusters in a data set,” Psychometrika, vol. 50, no. 2, pp. 159–179, June.

MITCHELL, T. M., 1997, “Machine Learning”, Ed. McGraw-Hill.

MOSCATO, ZUBEN, V., “Tópico 5: Uma Visão Geral de Clusterização de Dados.”, Disciplina Análise de Dados em Bioinformática, DCA/FEEC/Unicamp.

MOTRO, A., 1995, “Management of Uncertainty in Database Systems”, In: Kim, W. (ed), Modern Database Systems: The Object Model, Interoperability, and Beyond, 1 ed, chapter 22, New York, ACM Press.

MYRTVEIT, I., STENSRUD, E., OLSSON, U. H., 2001, “Analyzing Data Sets with Missing Data: An Empirical Evaluation of Imputation Methods and Likelihood-Based Methods”, IEEE Transactions on Software Engineering, v. 27, n. 11, Nov.

NEWGARD, C. D., HAUKOOS J. S.,LEWIS R. J. L., 2006,”Missing Data: What Are You Missing?” Society for Academic Emergency Medicine Annual Meeting, San Francisco, CA.

NIRELLI, L.M., LARSEN, M.D., CROGHAN, I.T., SCHROEDER, D.R., OFFORD. K.P., HURT, R.D., 2003, “Comparison of Methods for Handling Missing Data in a Collegiate Survey of Tobacco Use”, Iowa State University and Mayo Clinic, Department of Statistics, Snedecor Hall, Ames, Iowa

OBA, S., et al., 2003, “A bayesian missing value estimation method for gene expression profile data.”, Bioinformatics, 19, 2088–2096.

ONISKO, A., DRUZDZEL, M. J., WASYLUK, H., 2002, “An experimental comparison of methods for handling incomplete data in learning parameters of Bayesian networks”, In: Proceedings of the Intelligent Information Systems 2002 Symposium, pp. 351-360, Heidelberg.

OUDSHOORN, C. S. VAN BUUREN, and J. VAN RIJCKEVORSEL.,1999, ”Flexible multiple imputation by chained equations”, Netherlands Organization for Applied Scientific, Technical Report PG/VGZ/99.045.

OUYANG, M. *et al*., 2004, “Gaussian mixture clustering and imputation of microarray data”, Bioinformatics, 20, 917-923

PARSONS, S., 1996, “Current approaches to handling imperfect information in data and knowledge bases”, IEEE Transactions on Knowledge and Data Engineering, v. 8, n. 3 (Jun), pp. 353-372.

PEARL, J., 1988, “Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference”, 1 ed., Morgan Kaufmann Publishers.

PEUGH J.L., ENDERS C.K.,2004, “Missing Data in Educational Research: A Review of Reporting Practices and Suggestions for Improvement”, Review of Educational Research Winter, Vol. 74, No. 4, pp. 525–556

PIATETSKY-SHAPIRO, G.,1991, “Knowledge Discovery in real databases: A report on the IJCAI-89 Workshop”, AI Magazine, Vol. 11, No. 5, Jan. 1991, Special issue, 68-70.

P PIELA, S LAAKSONEN, 2001, “Automatic Interaction Detection For Imputation Œ Tests With The Waid Software Package”, Contributed Paper For The Federal Committee On Statistical.

PYLE, D., 1999, “Data Preparation for Data Mining”, 1 ed, San Francisco/CA, Morgan Kaufmann Publishers.

QUACKENBUSH, J., 2001, “Computational analysis of cDNA microarray data”, Nature Reviews, 6(2):418–428

QUINLAN, J. R., 1993, “C4.5: Programs for Machine Learning”, Ed. Morgan Kaufmann, San Francisco.

RAGEL, A., CRÉMILLEUX, B., 1998, “Treatment of Missing Values for Association Rules”, In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 258-270, Melbourne, Australia, Apr.

RAYMOND T., NG, HAN, J.,1994, “Efficient and effective clustering methods for spatial data mining”, Proc. Of VLDB Conference, Santiago, Chile, September

REZENDER, S.O., PUGLIESI J. B., MELANDA E. A., PAULA M. F, 2003. “Mineração de Dados”, S. O. Rezende (ed.), Sistemas Inteligentes: Fundamentos e Aplicações, Volume 1, Barueri, SP: Editora Manole.

RALLO R. FERR J., GIRALT F., 2004, “Multiple imputation of missing data using self organizing map ensembles “. Relatório Técnico do Departamento de Engenharia Química. Escola Técnica Superior de =ngenhari Química (ETSEQ) Universitat Roviari i Virgili – Catalunya – Espanha

RUBIN DB., 1978, “Multiple imputations in sample surveys – a phenomenological Bayesian approach to nonresponse”, Proceedings of the Survey Research Methods Section of the American Statistical Association, pp. 20–34.

RUBIN, D. B. 1987, “Multiple Imputation for Nonresponse in Surveys”, Wiley Series in Probability and Mathematical Statics, John Wiley & Sons.

RUBIN, D. B. 1988, “An Overview of Multiple Imputation”, In: Proceedings of the Section on Survey Research Methods, pp. 79-84, American Statistical Association.

RUBIN DB.,1996, “Multiple imputation after 18+ years.” Journal of the American Statistical Association, 91: 473–90.

RUBIN D.B., SCHENKER, N.,1991, “Multiple imputation in health-care databases: an overview and some applications”, Statistics in Medicine; 10: 585–98.

RUSSELL, S. NORVIG, P., 1995, “Artificial Intelligence: A Modern Approach.” New Jersey: Prentice-Hall.

SALTON, G.,1988, “Automatic text processing: the transformation, analysis, and retrieval of information by computer”, Addison-Wesley Publishing Company, Inc., New York.

SCHAFER, J.L., 1997. “Analysis of Incomplete Multivariate Data.”, Chapman and Hall, London.

SCHAFER, J. L., GRAHAM, J. W., 2002, “Missing Data: Our View of the State of the Art”, Psycological Methods, v. 7, n. 2, pp. 147-177.

SCHAFER, J.L, OLSEN, M. K.,1998, “Multiple imputation for multivariate missing-data problems: A data analyst’s perspective.” Multivariate Behavioral Research, 33(4):545–71.

SCHAPIRE, R. E., FREUND, Y., BARTLETT, P., LEE, W. S., 1998, “Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods”, The Annals of Statistics, v. 26, n. 5, pp. 1651-1686.

SCHEFFER, J., 2002, “Dealing with Missing Data”, Research Letters in the Information and Mathematical Sciences, v. 3, n. 1 (Apr), pp. 153-160.

SEHGAL, M.S.B., GONDAL, I., DOOLEY, L.S., 2005 , “Collateral missing value imputation: a new robust missing value estimation algorithm for microarray data”. Bioinformatics, 21, 2417–2423

SHANNON, C. E.,1948, “A mathematical theory of communication”. Bell System Technical Journal, 27:379 – 423, Jul.

SILVA, M.A.S., MONTEIRO, A.M.V., MEDEIROS, J.S., 2004, “Semi-Automatic Geospatial Data Clustering by Self- Organizing Maps.”, Anais do Simpósio Brasileiro de Redes Neurais, São Luiz, MA, outubro de 2004.

SKOWRON, A., RAUSZER, C., 2001, “The discemibility matrices and functions”, in, information systems, In Intelligent Decision Support: Handbook of Applications and Advances of Prevention Intervention Research, Washington. NIDA Research Monograph, the Rough Sets Theory (Edited by Slowinski, R.), Kluwer Academic Publishers,

SLONIM, D., ,2002, “From patterns to pathways: gene expression data analysis comes of age.”, Nature Genetics, 32:502–508.

SOARES, J., 2007, “ Pré-Processamento em Mineração De Dados: Um Estudo Comparativo em Complementção. “, Teses de Doutoramento, COPPE/UFRJ.

SOIBELMAN, L.; KIM, H., 2002, “Data preparation process for construction knowledge generation through knowledge discovery in databases.”, Journal of Computing in Civil Engineering, v.16, n.1, p.39-48, June.

SONG, Q., SHEPPERD, M., CARTWRIGHT, M., 2005, “A Short Note on Safest Default Missingness Mechanism Assumptions”, Empirical Software Engineering, v. 10, n. 2, pp. 235-243, Apr.

SOUTO, M.,2007, “Validação de Agrupamentos.” DIMAP/UFRN.

SRIKANT, R., AGRAWAL, R., 1997, “Mining generalized association rules”, Future Generation Computer Systems, v. 13, n. 2-3, pp. 161-180.

STRIKE, K., EL EMAM, K., MADHAVJI, N., 2001, “Software Cost Estimation with Incomplete Data”, *IEEE Transactions on Software Engineering*, v. 27, pp. 890-908.

SWINIARSKI, R.W., SKOWRON, A., 2003, “Rough set methods in feature selection and recognition.”, Pattern Recognition Letters, 24:833–849.

TEKNOMO, K., 2007a, “K-Means Clustering Tutorials”, Obtido em http://people.revoledu.com/kardi/tutorial/kMean em 27/03/2008.

TEKNOMO, K., 2007b,”Mean and Average”, Obtido em http://people.revoledu.com/kardi/tutorial/BasicMath/Average em 27/03/2008.

THEODORIDIS, S., KOUTROUMBAS, K.,1999, “Pattern Recognition”, Academic Press, NY.

TROYANSKAYA, O., CANTOR, M., SHERLOCK, G., BROWN, P., HASTIE, T., TIBSHIRANI, R., BOTSTEIN, D., ALTMAN, R., 2001, "Missing value estimation methods for DNA microarrays", Bioinformatics, v. 17, n. 0, pp. 1-6.

TSENG, S., WANG, K., LEE, C., 2003, “A Preprocessing Method to Deal with Missing Values by Integrating Clustering and Regression Techniques”, Applied Artificial Intelligence, v. 17, n. 5 (May-Jun), pp. 535-544.

TUFTE, R.E., 1983, “The Visual Display of Quantitative Information”, Graphics Press, Cheshire, Conn.

TWALA, B., CARTWRIGHT, M., SHEPPERD, M., 2005, “Comparison of Various Methods for Handling Incomplete Data in Software Engineering Databases”, International Symposium on Empirical Software Engineering, pp. 105-114, Nov.

ULTSCH, A, SIEMON H.P.,1990, “Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis”, Proc. Intern. Neural Networks, Kluwer Academic Press, Paris, pp. 305-308.

VACH, W., 1995, “Logistic Regression with Missing Values in the Covariates”, Technometrics, v. 37, n. 4, pp. 460-461, Nov.

VAN DER AALST, W., WEIJTERS, A., MARUSTER, L.,2003, ”Workflow mining: Discovering process models from event logs.”

VAN RIJSBERGEN, C.J., 1979, ”Information Retrieval”, Butterworths.

VERBOVEN S., BRANDEN V. K., GOOS P., 2007, “Sequential imputation for missing values”, *Computational Biology and Chemistry 31*, pp 320–327.

WAYMAN, J. C., 2003, “Multiple Imputation For Missing Data: What Is It And How Can I Use It?”. In: *Proceedings of the Annual Meeting of the American Educational Research Association*, Chicago, IL, Apr.

WFMC, Workflow Management Coalition Terminology & Glossary, The Workflow Management Coalition Specification, Winchester, UK. Fevereiro de 2009. Disponível em: http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf

WONG D. S. V., WONG F. K. e WOOD G. R., 2007 “A multi-stage approach to clustering and imputation of gene expression profiles,” BioInformatics, vol 23 no. 8, pages 998–1005, 2007

XU, Y. 1997, “Contextual tonal variations”, in Mandarin. Journal of Phonetics 25, 61-83.

ZAKI, M. J., PARTHASARATHY, S., OGIHARA, M., LI, W., 1997, “New Algorithms for Fast Discovery of Association Rules” In: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, pp. 283-286.

ZHANG, S., ZHANG, C., YANG, Q., 2003, “Data Preparation for Data Mining”, Applied Artificial Intelligence, v. 17, n. 5-6 (May-Jun), pp. 375-381.

ZHEN-PING L., BAVARIAN, B., 1993, “Multiple job scheduling with artificial neural networks.” Computers and Electrical Engineering archive. Volume 19, Issue 2, pp 87-101, Special issue on artificial intelligence in engineering design and manufacturing.

ZHU X., ZHANG S., ZHANG J., ZHANG C., 2006, “Cost-Sensitive Imputing Missing Values with Ordering”, American Association for Artificial Intelligence.

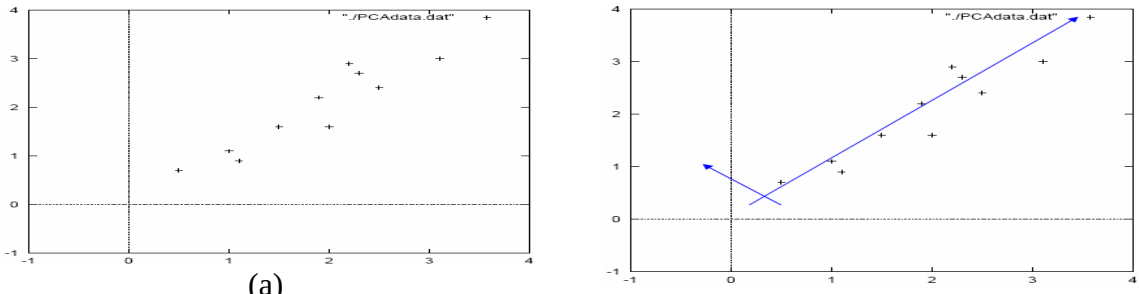
APÊNDICE I .

CONCEITOS BÁSICOS

Os conceitos descritos nesta seção são utilizados por diversas técnicas e medidas da etapa de pré-processamento. Considere que $X = (X_1, X_2, \ldots, X_n)$ e $Y = (Y_1, Y_2, \ldots, Y_n)$ são vetores do espaço de dimensionalidade *n*, representados na base canônica.

1. **Média Amostral**: é uma medida de tendência central que representa o valor médio de uma distribuição onde todas as ocorrências têm a mesma importância, ou seja, têm o mesmo peso relativo. Calculado pela soma de todos componentes da amostra, dividida pelo número de observações da mesma:

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$



(a) (b)

Figura I.1: O gráfico (a) mostra um conjunto de pontos em um gráfico que usa a base canônica de duas dimensões. Se obtivermos uma outra base, como na figura (b), poderemos observar que o novo sistema de coordenadas x'y' mostra uma variação bem mais significativa no novo eixo das abscissas. Fonte: SOARES (2007)

2. **Média Ponderada**: é também uma medida central onde as ocorrências podem não ter a mesma importância, ou seja, têm uma importância relativa diferente que deve ser considerada. Sendo assim, cada valor é multiplicado por um peso que expressa esta importância. É calculado por:

$$\bar{x}_p = \frac{\sum_{i=1}^{n} (p_i * x_i)}{\sum_{i=1}^{n} p_i}$$

3. **Desvio Padrão Amostral**: medida que indica o quão espalhados estão os dados em relação à média, quantificando o grau de dispersão (ou concentração) dos quantifica à dispersão dos eventos em relação ao ponto central. É calculado por:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}}$$

4. **Variância Amostral**: é também uma medida que expressa o desvio dos elementos da amostra em relação ao centro desta amostra, ou seja, a dispersão dos dados da amostra em relação à própria amostra. É calculado pela soma dos quadrados dos desvios, dividido pelo número de elementos.

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}$$

É bom salientar que o denominador da variância e o do desvio padrão, que indica a quantidade de elementos da amostra, é *n-1*, e não *n*. Esta diferença tem origem na teoria estatística, e basicamente reporta à diferença entre amostra e população, pois como a amostra é um subconjunto observado de valores da população, e que não reflete todas as suas características, torna-se necessário sua representação na fórmula. Logo, a redução de *n para* n-1 simboliza que nem todos os dados da população foram considerados no cálculo da medida. Se tivéssemos todos os valores da população registrados, aí sim o denominador poderia ser substituído por *n*.

5. **Covariância**: indica se existe alguma relação na variação dos dados de duas amostras de mesma dimensão:

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

A variância é um caso particular da covariância, quando observa-se a variação da amostra em relação a ela mesma:

$$var(X) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})}{(n-1)}$$

Quando as variáveis são independentes, a sua covariância é zero, já que o valor esperado do produto das variáveis é igual ao valor esperado de cada uma delas:

$$E(X \cdot Y) = E(X) \cdot E(Y)$$

6. **Matriz de Covariância**: Matriz simétrica, que indica todas as combinações de covariâncias entre duas medidas de uma amostra:

$$C^{n \times n} = (c_{i,j},\ c_{i,j} = cov(Dim_i, Dim_j))$$

Por exemplo, em uma amostra com três tipos de medidas (*x*, *y* e *z*), a matriz de covariância seria:

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

Conforme já citamos, a matriz é simétrica: *cov(x,y) = cov(y,x)* pois *y* varia em função de *x* do mesmo modo que *x* varia em função de *y*. Além disso, por o cálculo da covariância envolver cálculo de quadrados, seu valor nunca é negativo ($cov(x,y) \geq 0$).

7. **Normalização :** as variáveis podem ser normalizadas segundo a amplitude de seus valores, mais aplicável no caso de unidades diferentes ou dispersões muito heterogêneas, ou segundo a distribuição dos mesmos, aplicável para a remoção de distorções de *outliers*, ou obtenção de simetria. Goldschmidt et al (2005) e HAN e KAMBER (2005) apresentam algumas das normalizações mais utilizadas em aplicações práticas:

   a. **Normalização Linear**: também conhecida como normalização por interpolação linear, mapeias os valores do atributo no intervalo [0.0, 1.0]. Considera os valores máximos e mínimos de cada atributo no ajuste de escala, mantendo a distância relativa entre os dados normalizados e os originais:

$$A' = \frac{(A - A_{\min})}{(A_{\max} - A_{\min})}$$

b. **Normalização Min-Max** mapeia os valores do atributo no intervalo [novo_min$_A$, novo_max$_A$]. É uma variação do método anterior, que permite a determinação dos limites do intervalo dos dados normalizados. Considera os valores máximos e mínimos de cada atributo no ajuste de escala, mantendo a distância relativa entre os dados normalizados e os originais na nova escala:

$$A' = \frac{(A - A_{\min})}{(A_{\max} - A_{\min})} * (novo\_\max_A - novo\_\min_A) + novo\_\min_A$$

c. **Normalização pela Soma**: divide o valor do atributo pelo somatório de todos os valores deste atributo.

$$A' = \frac{A}{\sum A}$$

d. **Normalização por Score-Z (normalização por Desvio Padrão ou Média Zero)**: é um dos métodos mais populares e foi o escolhido nesta tese. É útil no caso em que não se conhece os valores máximos e mínimos do atributo ou não se pode garantir a existência dos mesmos. Ajusta os valores considerando a média dos valores do atributo e o grau de dispersão deste em relação à média, ou seja, desloca o eixo central para a média dos valores do atributo, e os normaliza em função do seu desvio-padrão:

$$A' = \frac{(A - \overline{A})}{\sigma}$$

e. **Normalização pelo Máximo**: é semelhante à normalização linear. Utiliza como fator de normalização o maior valor existente para o atributo em questão.

$$A' = \frac{A}{A_{\max}}$$

f. **Normalização por Escala Decimal**: desloca o ponto decimal dos valores do atributo a ser normalizado. A quantidade de casas decimais depende do maior valor absoluto de *A*.

$$A' = \frac{A}{10^j}$$

43Aqui, *j* é o menor inteiro tal que o maior valor absoluto normalizado seja inferior a 1, ou seja, $\max|A'| < 1$.

8. **Medidas de Similaridade para Atributos Numéricos:** Em geral, as medidas para determinar a semelhança entre objetos onde o domínio dos valores dos atributos é contínuo, interpretam o objeto como um vetor n-dimensional, onde n corresponde à quantidade de atributos. Neste contexto, medidas geométricas de distâncias podem ser utilizadas. Há uma grande variedade de métricas com esse propósito na literatura. A escolha da métrica aplicável é subjetiva e dependente do problema.

   A escolha mais freqüente é a **distância Euclidiana**:

$d(a,b) = \sqrt[2]{\sum_{i=1}^{n}(a_i - b_i)^2} = \|a-b\|_2$, considerando a e b dois vetores de dados.

que é um caso particular, para o valor de p=2, da distância Minkowski:

$$d_{Mi}(a,b) = (\sum_{i=1}^{n}|a_i - b_i|^p)^{\frac{1}{p}} = \|a-b\|_p$$

Outra métrica derivada da distância Minkowski é a **distância Manhattan**, (para o valor de p = 1) também conhecida como distância City-Block :

$$d(a,b) = \sum_{i=1}^{n}|a_i - b_i| = \|a-b\|_1$$

Estas distâncias, apesar de populares, são sensíveis à variação dos dados. Quando a ordem de grandeza de um atributo é muito maior que as demais, o cálculo da distância pode tornar-se desproporcional. Uma possível solução para este problema é a normalização dos valores dos atributos. No entanto, a normalização dos dados também não é adequada em algumas aplicações da tarefa de agrupamento, particularmente quando a dimensionalidade dos dados é grande. SALTON (1988) discute as deficiências das medidas baseadas em distâncias.

Outro fator que pode prejudicar as métricas da família Minkowski é a existência de correlação linear acentuada entre os atributos dos objetos. Para compensar a variância e a correlação entre as variáveis, podemos usar a **distância Malahanobis**:

$$d_{Ma}(a,b) = \sqrt{(a-b)^T Cov^{-1}(a-b)} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\sigma_{ij}(a_i - b_i)(a_i - b_i)}$$

onde Cov^{-1} é a matriz de covariância entre os vetores *a* e *b*. Quando a matriz de covariância é a matriz identidade I a distância de Mahalanobis é equivalente à distância Euclidiana. Quando *Cov* é uma matriz diagonal (ou seja, uma matriz com todos os elementos da diagonal principal diferentes de zero, e os demais elementos iguais a zero), a distância de Mahalanobis pode ser vista como uma generalização da distância Euclidiana. Essa generalização permite considerar correlações entre as características dos objetos.

Para melhor entendimento, considere o problema de estimar a probabilidade de um ponto x no espaço euclidiano n-dimensional pertencer a um conjunto do qual se tem uma amostra (um conjunto de pontos pertinentes a ele). Inicialmente calcula-se a média ou centro da massa de pontos desta amostra. Intuitivamente, quanto mais próximo o ponto x estiver do centro da massa, mais provável ele pertencer ao conjunto. No entanto, é necessário, também, conhecer a largura deste conjunto. Uma abordagem simplista de tal cálculo é estimando o desvio padrão das distâncias dos pontos da amostra em relação ao centro da mesma. Assim, se a distância entre o ponto x e o centro da massa é menor que o desvio padrão, então pode-se concluir que é altamente provável que o ponto pertença ao conjunto. Quanto mais longe ele estiver, então mais provável ele não pertencer ao conjunto.

Uma deficiência desta abordagem é assumir que os pontos da amostra estão distribuídos em uma esfera ao redor do centro. No entanto se a distribuição tiver uma forma geométrica elipsoidal, por exemplo, então a probabilidade do ponto x pertencer ao conjunto depende não só do centro da massa mas também da direção. Nas direções onde o eixo da elipse é mais estreito, o ponto x deve estar mais perto enquanto nas direções onde o eixo é mais longo este ponto pode estar mais longe do centro. Portanto, mais formalmente, a elipse que melhor representa a distribuição de probabilidade do conjunto pode ser estimada construindo a matriz de covariância dos exemplos. A distância de Malahanobis é a distância do ponto do centro da massa dividido pela largura da elipse na direção do ponto.

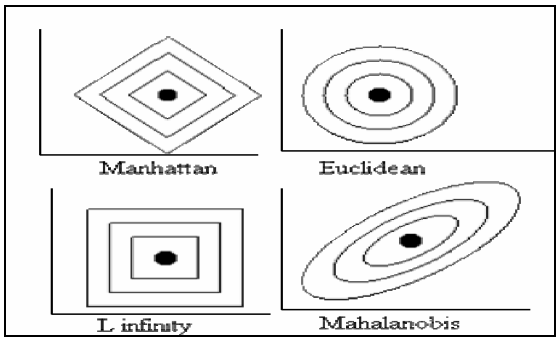


Figura I.2: Interpretação geométrica de algumas medidas
Fonte: Salton, 1988

O coeficiente de correlação linear de Pearson (DIDAY e SIMON, 1980) que considera a associação linear entre 2 variáveis é também bastante popular. É uma medida robusta em relação à rotação e escala, variando de -1 a 1 (inversamente correlacionadas à perfeitamente correlacionadas). Dados dois vetores x_i e x_j esta medida é definida como:

$$d_{PCC}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{k=1}^{p}(x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)/p}{\sigma_i \sigma_j}$$

onde $\bar{x}_i$ é a média e σ_i é o desvio padrão dos valores do vetor x_i e p é o número de atributos do vetor. Na Bioinformática, utiliza-se uma variação desta medida como uma medida de *dissimilaridade* ou distância:

$$d_p = \frac{1 - d_{PCC}\rho(\mathrm{x}, \mathrm{y})}{2}$$

Há uma grande variedade de distâncias entre classes ou grupos/ padrões que podem ser agrupadas em famílias de acordo com o critério que utilizam. Algumas, entre as principais, são (Theodoridis et al,1999):

- Distâncias entre centróides: determinam um representante e medem a distância entre eles;
- Distâncias baseadas em matrizes de espalhamento: utilizam medidas baseadas na análise de discriminantes;
- Distâncias geométricas: como a Família de Minkowski. Utilizam critério espacial, a geometria formada pelos pontos;

- Distâncias baseadas em correlação e variância: consideram a forma geométrica e a correlação e variância entre os padrões, como as distâncias da família Mahalanobis;
- Distâncias baseadas em funções de probabilidade: como a distância de Bhattacharyya e divergência. Calculam as funções de densidade de probabilidade dos conjuntos e usam como critério a diferença na forma destas funções;
- Distâncias nebulosas: aplicam os conceitos da teoria Fuzzy;
- Distâncias na análise da entropia: usam o grau de informação intrínseco(entropia) para medir a distância entre grupos/classes

Desta diversificação um fato importante deve ser salientado: os agrupamentos resultantes destas medidas podem ser completamente diferentes. Portanto fica a pergunta. “Como escolher?”

APÊNDICE II

TAREFAS DE MINERAÇÃO DE DADOS

Na etapa de Mineração de Dados os algoritmos responsáveis por “garimpar” o conhecimento são executados e por isso esta etapa é usualmente considerada o coração do processo. No entanto, como já dito, se os dados não apresentarem a qualidade necessária, se estiverem desbalanceados, com ruídos, incompletos, com erros ou inconsistentes, caso os algoritmos de mineração consigam atuar e extrair padrões estes podem ser falsos ou inválidos.

Esta etapa caracteriza-se pela seleção e aplicação do(s) algoritmo(s) de mineração adequado(s) à tarefa de mineração que atende ao objetivo do processo KDD. As tarefas são definidas como classes de problemas e os algoritmos são as técnicas para solucionar tais problemas. Cada tarefa apresenta várias técnicas, e algumas técnicas podem ser utilizadas para solucionar tarefas diferentes. As tarefas de KDD podem ser primárias ou compostas. Uma tarefa primária de KDD é aquela que não pode ser desmembrada em outras tarefas de KDD. Por outro lado, uma tarefa de KDD composta pode ser desmembrada em duas ou mais tarefas primárias de KDD (GOLDSCHIMDT, PASSOS., 2005). Como já descrito, as tarefas básicas, são classificadas em duas categorias: preditivas e descritivas. As tarefas preditivas subdividem-se em classificação e regressão. Já as descritivas subdividem-se em regras de associação, clusterização e sumarização conforme ilustrado na figura abaixo: (REZENDE, 2003).

Figura II.1. Taxonomia das Tarefas de Mineração de Dados
Fonte: (REZENDE, 2003)

1. Classificação:

É uma das tarefas mais usuais em KDD com aplicabilidade nas mais diferentes áreas do conhecimento. O objetivo principal é encontrar uma função que mapeie cada registro de um conjunto de dados em uma de n possíveis classes Ci de um determinado problema. Se esta função for encontrada, pode-se determinar a classe de novas tuplas, ou seja, de registros não conhecidos previamente.

Portanto, em cada registro há, além dos atributos que descrevem as características do objeto (conjunto de atributos previsores), um atributo que representa a classe da observação (atributo meta) e a tarefa é descobrir um relacionamento entre os atributos previsores e o atributo meta, usando registros cuja classe é conhecida com intuito de empregar o relacionamento descoberto para prever a classe (o valor do atributo meta) de um registro com classe desconhecida. Este processo é denominado ***aprendizado*** (RUSSEL e NORVIG, 1995). A função que realiza este mapeamento é conhecida como classificador.

Todo algoritmo que possa ser utilizado na execução do processo de aprendizado é chamado ***algoritmo de aprendizado***. Não existe um algoritmo de classificação que se sobressaia em relação a todos os outros, independente do problema, conforme atesta o teorema ***NFL***–*No Free Lunch Theorem* (WOLPERT, 1996). Sendo assim, a cada nova aplicação que envolva a tarefa de classificação deve-se experimentar os algoritmos disponíveis para identificar os que obtêm melhor desempenho.

As regras de classificação são extraídas a partir de uma base de treinamento. Em geral esta tarefa divide a base de dados em duas. Uma é usada para o treinamento do algoritmo de aprendizado, ou seja, para extrair as regras de classificação e a outra é usada como teste, para determinar a precisão do classificador gerado.

2. Associação

A tarefa de mineração de associações também conhecida por descoberta de associações, tem como objetivo encontrar padrões válidos de relacionamento entre os objetos do domínio que ocorram com uma determinada freqüência nas bases dados. Os padrões minerados são expressos na forma de regras e possuem índices associados que representam sua relevância (fator de suporte) e validade (fator de confiança). Em geral,

somente as regras de associação que possuem valores para este dois fatores maiores que limiares especificados pelo usuário são extraídas, ou seja, estes fatores são utilizados como medidas da seleção.

Seja I = {i_1, i_2, ..., i_n} o conjunto de objetos (itens) do domínio da aplicação, uma regra de associação R definida sobre I é uma implicação da forma $X \rightarrow Y$, onde $X \neq \emptyset$, $Y \neq \emptyset$, $X \subset I$, $Y \subset I$, *e* $X \cap Y = \emptyset$. *X* é denominado **antecedente** e Y é o **conseqüente**. A condição $X \cap Y = \emptyset$ assegura que regras triviais não sejam geradas.

O fator de *suporte,* que indica a relevância de uma regra $X \rightarrow Y$ representa o percentual de transações que satisfazem tanto ao antecedente quanto ao conseqüente da regra, ou seja, a freqüência com que uma regra ocorre na tabela. Seja a regra $r = X \rightarrow Y$, *nX* e *nY* o número de vezes que os valores *X* e *Y* aparecem em seus respectivos atributos, e *D* o número de registros da tabela. O suporte de *r* é definido como:

$$suporte\ (r) = \frac{nX \wedge nY}{D}$$

A *confiança* reflete a validade de uma regra e representa o percentual de transações que incluem os itens X e Y em relação a todas que incluem os itens de X. É uma medida que procura expressar o quanto a ocorrência do conseqüente pode ser assegurada pelas transações que incluem o antecedente Ela é expressa por:

$$confiança\ (r) = \frac{nX \wedge nY}{nX}$$

A descoberta de regras de associação é geralmente executada em duas fases. Na primeira fase, o método identifica todos os conjuntos de itens freqüentes, Z, ou seja, cujo fator de suporte é superior ao valor mínimo (SupMin), estipulado pelo usuário. A seguir, para cada conjunto freqüente Z, identifica os possíveis subconjuntos X e Y, gerando todas as regras candidatas (X→Y) e testando-as em relação ao fator de confiança. As regras produzidas, chamadas de regras de interesse, são as que superam o valor mínimo estipulado pelo usuário para o fator de confiança.

Há muitos algoritmos desenvolvidos especificamente para aplicação na tarefa de descoberta de associações. O mais famoso é o *Apriori*, de AGRAWAL, IMIELINSKI e SWAMI (1993). Porém, existem outras opções, tais como DHP–*Direct Hashing and*

Pruning (HOLT, CHUNG, 1999), DIC–*Dynamic Itemset Counting* (BRIN *et al*, 1997), *Eclat* e *MaxEclat* (ZAKI *et al*, 1997) e *EstMerge* (SRIKANT, AGRAWAL, 1997).

3. Descoberta de Seqüências:

Tarefa valiosa no mercado varejista e na medicina, por exemplo. Identifica padrões levando em consideração o aspecto temporal. Nesta tarefa, busca-se por padrões que existam levando em consideração itens freqüentes em várias transações ocorridas ao longo de um período. Assim, procura capturar nos dados de entrada uma relação de ordem cronológica para identificar os padrões seqüenciais com um suporte mínimo pré-definido. São úteis para descobrir tendências nos dados, tais como: "O número de pessoas que realizam exames preventivos está crescendo na classe social A", por exemplo. É uma extensão da tarefa de descoberta de associações comentada anteriormente.

4. Sumarização:

Consiste em descrever de um modo simples, compreensível e compacto um conjunto de dados, apresentando as principais características dos dados contidos neste conjunto. É usual na criação automática de relatórios e para análise exploratória (FAYYAD, 1996a). Pode ser utilizada em uma fase inicial do processo de KDD onde se deseje ter um melhor conhecimento da base. Entre os métodos que estão aptos a realizar esta tarefa estão a Lógica Indutiva e Algoritmos Genéticos (GOLDSCHMIDT, PASSOS, 2005).

5. Regressão:

Tem como objetivo encontrar uma função que mapeie os registros de um banco de dados em valores reais. Difere da tarefa de classificação, pois é aplicada a atributos numéricos. Alguns exemplos desta tarefa são: predição da soma da biomassa presente em uma floresta; estimativa da probabilidade de um paciente sobreviver, dado o resultado de um conjunto de diagnósticos de exames; predição do risco de determinados investimentos, definição do limite do cartão de crédito para cada cliente em um banco;

dentre outros (GOLDSCHMIDT, PASSOS, 2005). Métodos da área de Estatística e Redes Neurais são usualmente utilizados para implementar esta tarefa.

6. Previsão de Séries Temporais:

Uma série temporal é um conjunto de observações de um fenômeno ordenadas no tempo. A sua análise é o processo de identificação das características, padrões e propriedades importantes, utilizados para descrever em termos gerais o seu fenômeno gerador. Dentre os diversos objetivos da análise de séries temporais, o maior deles é a geração de modelos voltados à previsão de valores futuros (SOARES 2007).

7. Detecção de Desvios:

O objetivo desta tarefa é identificar os registros da base que apresentem características diferentes das usuais, ou seja, apresentem comportamentos não considerados normais no contexto. Tais registros são denominados "*outliers*". As administradoras de cartão de crédito, por exemplo, costumam realizam esta tarefa para monitorar o uso do cartão de forma diferente da usual, identificando desvios nos padrões normais de compra. A Estatística fornece recursos para a implementação desta tarefa.

APÊNDICE III

MÉTODOS DE AGRUPAMENTO

Há uma vasta quantidade de algoritmos de agrupamento propostos na literatura. Este fato deve-se principalmente a: 1) inexistência de uma técnica única capaz de extrair toda a variedade de estruturas ou relações presentes em conjuntos multivariados; 2) o conceito de similaridade normalmente pressupõe que os objetos da coleção estão posicionados no espaço de acordo com alguma configuração geométrica, por exemplo, hiperesféricas, hiperelipsoidais, dentre outras, o que não corresponde à realidade. Deste modo, um algoritmo pode apresentar um comportamento superior a outro para um dado contexto e pior em um contexto distinto. BEZERRA (2006) descreve alguns possíveis critérios para classificação destes algoritmos:

1. *Estratégia de agrupamento*: de que forma o algoritmo de agrupamento interpreta os objetos com o objetivo de formar grupos a partir desses objetos?
2. *Natureza da pertinência de objetos em grupos*: objetos podem pertencer a mais de um grupo simultaneamente?
3. *Estrutura do agrupamento*: como um algoritmo de agrupamento processa a coleção de objetos *X* para identificar os grupos ? Qual é a estrutura dos grupos gerados por ele?

1. Estratégia de Agrupamento

Quanto à estratégia utilizada, pode-se classificar algoritmos de agrupamento em *geradores* ou *discriminativos*. Métodos geradores (ou baseados em modelos) consideram que os objetos são agrupados de acordo com uma mistura de distribuições de probabilidades, onde cada componente da mistura corresponde a um dos grupos. Métodos *discriminativos* (ou baseados em similaridade) calculam a distância ou similaridade entre todos os possíveis pares de objetos e agrupam os pares similares entre si (BEZERRA, 2006).

A estratégia de um algoritmo de agrupamento leva em consideração o seguinte pressuposto: os elementos de um grupo respeitam algum tipo de distribuição probabilística, ou uma mistura de distribuições. Um exemplo desta classe de algoritmos é o **K-means**, que assume como premissa uma distribuição normal dos objetos da

coleção de entrada. Outro exemplo é o **algoritmo EM** (*Expectation-Maximization*) (DEMPSTER, LAIRD, RUBIN, 1977), bastante usado também na tarefa de complementação de dados. Esta técnica tenta estimar os coeficientes da função densidade de distribuição probabilística, e pode também ser usada para agrupamento de dados. O algoritmo *K-means* é considerado um caso particular do algoritmo EM, para distribuição normal dos dados.(SOARES, 2007).

2. Natureza da Pertinência de Objetos em Grupos

Neste critério observa-se como o algoritmo define o grau de pertinência de um objeto em *X* em um ou mais dos *K* grupos. De acordo com esse critério, há os seguintes tipos de agrupamento (BEZERRA, 2006):

- **Agrupamento sobreposto** (*fuzzy clustering*). Esta família de algoritmos de agrupamento utiliza técnicas da Teoria de Conjuntos Difusos (*Fuzzy Set Theory*) para agrupar os objetos. Nestes algoritmos é possível um objeto pertencer a mais de um grupo simultaneamente. Um exemplo de trabalho que propõe um algoritmo de agrupamento sobreposto é o *Fuzzy k-Means* (ARIMA,2003).
- **Agrupamento estrito** (*hard clustering*). Esta família de algoritmos considera que cada objeto deve pertencer somente a um dos grupos, ou seja $G_i \cap G_j = \varnothing$, $i \neq j$. A maioria dos algoritmos existentes cria grupos sem sobreposição.

3. Estrutura do Agrupamento

Quanto à estrutura gerada, os algoritmos de agrupamento podem ser divididos em: *partitivos e hierárquicos*. As características de cada tipo encontram-se descritas nas duas próximas seções.

a) Métodos Hierárquicos

Em algoritmos de agrupamento hierárquico, a similaridade entre os objetos é organizada em uma estrutura hierárquica, ou seja, o relacionamento dos objetos (ou grupos) são representados por uma árvore, usualmente conhecida como **dendrograma**. Objetos similares ficam em ramos próximos da árvore. O comprimento dos ramos reflete o grau de similaridade e registra a seqüência de uniões e divisões do processo de agrupamento. A árvore retornada por estes métodos pode ser facilmente transformada

em partições, bastando “cortar” o dendrograma em um certo nível. A figura 2.13 demonstra este conceito.

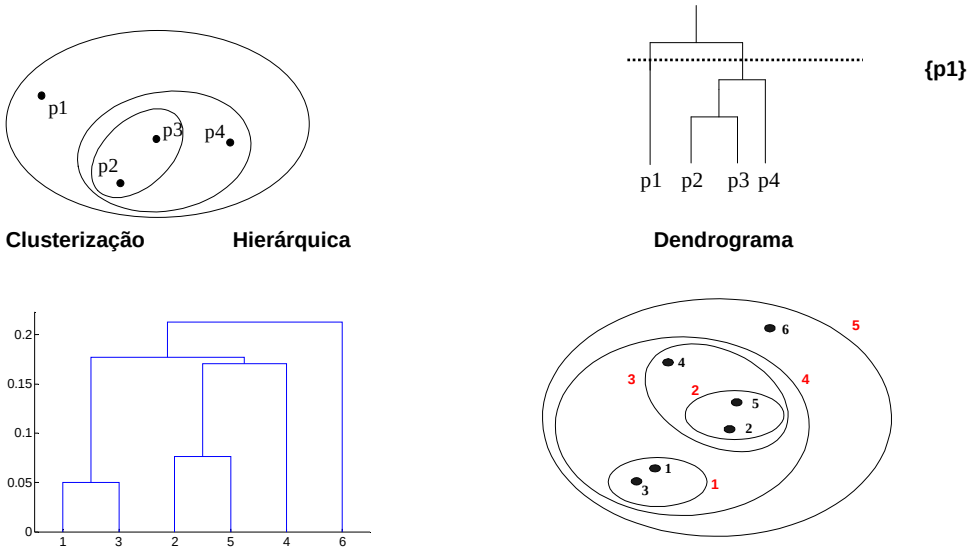
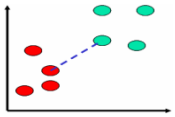


Figura III.1: Exemplos de Dendogramas
Fonte: Adaptado de (Tan,Steinbach,Kumar, 2004)

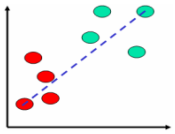
Considerando o processo de construção da árvore, os algoritmos hierárquicos podem ser subdivididos em dois tipos:

- *divisivos (top-down*): partem de um grupo único (conjunto inicial de objetos) e, a cada passo, dividem os grupos gerados na iteração anterior em grupos menores, até que *n* grupos sejam formados (geralmente unitários).
- *aglomerativos (bottom-up*): partem de *n* grupos (por exemplo, cada objeto da coleção é considerado um grupo) e, iterativamente, unem os grupos menores em grupos cada vez maiores, até que um único grupo que contém todos os objetos seja formado. Há diversos algoritmos aglomerativos na literatura e são divididos de acordo com o modo que calculam a proximidade entre os grupos a unir (JAIN, MURTY e FLYNN,1999). Os principais grupos são “*single-linkage*”, “*complete-linkage*” e “*average-linkage*”.

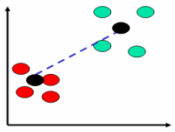
- *single-linkage*: É uma abordagem otimista onde a distância entre dois grupos é determinada pela menor distância entre os objetos de cada grupo;



- *complete_linkage:* è uma abordagem pessimista onde a distância entre dois grupos é determinada pela maior distância entre os objetos de cada grupo



- *average_linkage:* a distância entre dois grupos é determinada pela distância entre os centróides do grupo, sendo que os centróides são as médias das distâncias dos objetos de cada grupo



Um dos algoritmos hierárquicos propostos na literatura é o CURE (GUHA,1998). O CURE utiliza um número constante de pontos para representar cada grupo. Os representantes escolhidos são os que estão mais espalhados possível dentro do grupo. Uma vez selecionados, esses representantes são movidos por um fator α $(0 \leq \alpha \leq 1)$ em direção ao centróide do grupo. Iterativamente, os grupos são unidos em função de suas similaridades. A similaridade entre dois grupos é medida pelo método de ligação simples, entre os representantes desses dois grupos. A escolha correta do parâmetro α, permite ao CURE amenizar os problemas encontrados nos métodos partitivos (preferência por grupos esféricos e de tamanho uniforme e suscetibilidade a ruído).

O método conhecido como WARD, também é bastante usado, principalmente nos trabalhos vinculados à Ecologia. Neste método, o conceito que norteia a formação dos grupos é a variância mínima. Inicialmente, há um grupo para cada amostra da base de dados. Nesta fase, a variância em todos os grupos é nula, pois cada amostra é o próprio vetor médio do grupo. A cada etapa do processo de agrupamento, para cada possibilidade de aglomeração, é calculada a variância interna do novo grupo escolhendo-se a menor delas. As possibilidades de aglutinação entre os grupos são

verificadas, e é escolhido o agrupamento que causa o menor aumento no erro interno do grupo.

Independente da abordagem, os diferentes métodos hierárquicos existentes seguem o seguinte algoritmo:

Entrada: Conjunto de objetos a agrupar
Saída: Dendrograma : árvore com a representação hierárquica dos grupos Algoritmo:

Calcular a matriz de proximidade entre todos os objetos do conjunto inicial;

Torne cada objeto um grupo unitário;

Repita

Unir os dois grupos considerados mais próximos pela abordagem;

Atualizar a matriz de proximidade com o novo grupo;

Até restar apenas um grupo

Os algoritmos hierárquicos tradicionais apresentam limitações e entre elas pode-se salientar: alta complexidade computacional (O(n2log n), onde n é o número de objetos) e o não retrocesso na formação dos grupos, ou seja, iterações posteriores robustas a ruídos. (MANGIAMELI et al., 1996). não podem desfazer um grupo formado, mesmo que este não seja o mais adequado Em conseqüência desta característica determinística, os métodos hierárquicos não são robustos a ruídos. (MANGIAMELI et al., 1996).

b) Métodos Partitivos

Em algoritmos partitivos, os objetos são divididos em k grupos mutuamente exclusivos, sendo k usualmente um parâmetro de entrada do algoritmo. O método escolhido cria k partições e avalia a pertinência dos objetos a cada grupo de acordo com algum critério ou função de energia. Uma característica desta classe de algoritmos é considerar a existência de um objeto representativo do grupo, conhecido como centróide. O centróide μ_i é um vetor de mesma dimensionalidade dos demais vetores associados aos objetos de um grupo e representa um ponto central do grupo podendo ser um vetor associado a um objeto ou gerado a partir da média dos vetores associados aos

objetos do grupo. Pressupondo a existência deste ponto central em cada grupo, o objetivo do algoritmo partitivo é otimizar (maximizar/minimizar) uma função-objetivo baseada na distância dos demais objetos em relação aos centróides dos grupos. Ao amparar-se no conceito de "centro" de um grupo os métodos partitivos, em geral, pressupõem que o conjunto de objetos a agrupar possuem uma distribuição de probabilidades gaussiana. Os dois maiores representantes da família de métodos partitivos, com o k pré-determinado, são o K-Means e o K-Medoids . No K-Means o centro de um grupo (o centróide) é a média dos objetos do grupo, enquanto no K-Medoids o centro de um grupo é o objeto (da coleção de entrada) que se encontra mais próximo do centro.

→ *K-Means Clássico*

O algoritmo partitivo dos ***K*-Centróides**, mais conhecido como *K-Means* (MCQUEEN, 1967), é um dos mais antigos e importantes algoritmos de agrupamento desta classe disponíveis na literatura.

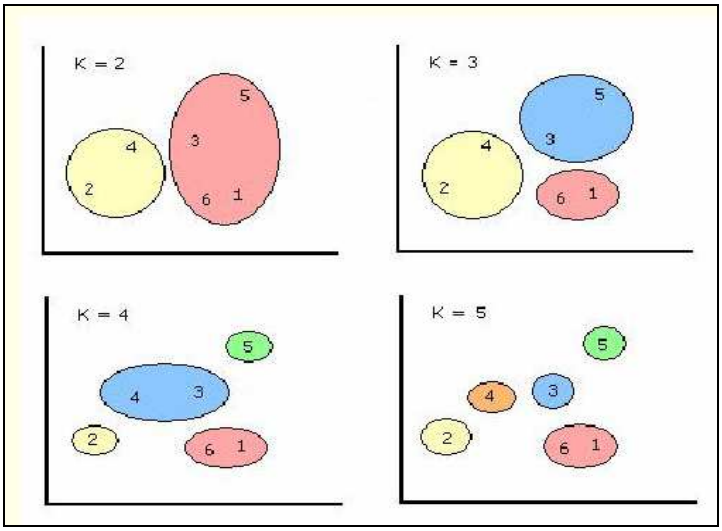


Figura III.2: Exemplo do K-Means, para k=2 a 5
Fonte: Adaptado de Chipman et al. (2003)

O K-Means é um algoritmo guloso no qual o objetivo principal é definir k centros, conhecidos como centróides, (um para cada grupo) através da otimização local de uma função objetivo. O centróide de um grupo é definido como o vetor médio de todos os vetores correspondentes aos objetos associados a este grupo. Deste modo, a tarefa do algoritmo K-Means é minimizar uma função-objetivo correspondente à distância total entre os objetos e os centróides dos grupos aos quais esses objetos foram associados.

Nesse sentido, o algoritmo busca um mínimo local para o problema de minimizar a função de soma dos erros médios quadrados, cuja fórmula é apresentada a seguir. O processo de otimização (local) do *K*-Means é iterativamente realizado até que algum critério de convergência se verifique.

Intuitivamente, o algoritmo K-Means funciona da seguinte maneira (QUACKENBUSH, 2001). Após distribuir aleatoriamente os objetos (padrões) da coleção de objetos X a cada um dos K grupos (pré-determinado), os centróides, contendo a média dos vetores associados aos objetos pertencentes a cada grupo, são calculados. Esses centróides representam os grupos e são utilizados para calcular as distâncias entre os objetos dos grupos. Os padrões são iterativamente deslocados entre grupos, de acordo com as distâncias intra- e inter- grupos. Padrões deslocam-se para um novo grupo caso estejam mais próximos do centróide desse novo grupo do que de seu grupo atual. Após o deslocamento, os centróides são recalculados. O processo continua até ser alcançado um critério de convergência (em geral a estabilidade dos padrões em relação à sua alocação aos grupos).

Entrada: Número K de grupos
conjunto C de objetos

Saída: K grupos com N objetos da coleção original *C*

Algoritmo:

Escolher aleatoriamente K objetos como centróides, um para cada grupo

Repita

Associar cada objeto da coleção a um centróide: $\sum_{x_i \in G} \|x_i - \mu_i\|^2$

Calcular o centróide para cada novo grupo: $\frac{1}{|G_i|} \times \sum_{x \in G_i} x$

Até estabilização dos objetos ou atingir um número máximo de iterações

---

A determinação correta do valor de k é uma questão importante no K-Means. Usualmente, executa-se o algoritmo diversas vezes, testando diferentes valores de k para selecionar aquele que gerou um agrupamento de melhor qualidade considerando algum critério de avaliação (vide seção 2.4.4). Uma heurística apresentada na literatura é que o *k* varia de 2 a $\sqrt{N}$, onde N é total de elementos no conjunto de dados.

Os principais problemas encontrados por métodos de agrupamento partitivo são: (1) o número de grupos a serem identificados deve ser conhecido a priori; (2) sensível à escolha dos primeiros k centróides; (3) a dificuldade em identificar grupos com grandes variações de tamanho. Grupos grandes (relativamente aos demais) tendem a ser divididos; (4) a presença de ruído (*outliers*) nos dados de entrada pode alterar significativamente o cálculo do centro de um grupo. Esses métodos comportam-se melhor quando o coleção de objetos a agrupar tende a formar grupos que apresentam uma estrutura esférica, sendo que grupos com outras geometrias podem não ser encontrados (JAIN et al., 1999), (BEZDEK,1998). Esse aspecto dos algoritmos partitivos tem como origem o fato de eles utilizarem um único objeto como representativo de cada grupo (normalmente esse objeto corresponde ao centróide do grupo, ou ao objeto mais próximo do centro). Apesar desses problemas, métodos de agrupamento partitivo são bastante populares e utilizados em aplicações práticas.

→ *Variantes do K-Means*

A seguir encontram-se relacionados alguns algoritmos cuja formulação foi idealizada a partir do algoritmo K-Means:

a) K-Medoids - É o precursor de uma série de algoritmos. Como já citado, a principal diferença entre o K-Means e o K-Medoids é que no último os centróides são dados da base, enquanto que no primeiro, os dados são médias dos pontos, que podem não coincidir com pontos reais, tornando o processo sensível a *outliers*. Outro ponto importante é que os resultados deste algoritmo não dependem da ordem em que os dados são apreciados (RAYMOND, 1994).

b) DBSCAN – A idéia básica é que, para cada ponto de um cluster, a vizinhança a um certo raio deve conter ao menos um número mínimo de pontos, ou seja, a densidade da vizinhança precisa atingir um limite mínimo especificado (XU et al., 1997a).

Outras discussões sobre métodos de agrupamentos podem ser encontradas em (JAIN88), (ARAB96), (RAYM94) e (AURÉLIO 1999).

Técnicas Baseadas em Redes Auto-Organizáveis

Técnicas baseadas em redes neuronais auto-organizáveis, em geral utilizando algoritmos baseados no modelo Kohonen (HAYKIN, 1994), também são capazes de segmentar grupos de dados e costumam ser eficazes para os casos onde há grupos com distribuições mal-comportadas.

A essência da auto-organização está na descoberta de formas (estrutura) e organizações (ordem) existentes em um sistema sem que haja imposição ou interferência impostas pelo meio externo. Portanto, é um fenômeno interno ao sistema, resultante da interação de seus componentes e não dependente da natureza física destes componentes (MOSCATO e VON ZUBEN, 2004). Sob esta ótica, a tarefa de agrupamento de uma coleção de objetos pode ser definida como um processo no qual o número de grupos é desconhecido, mas sabe-se que eles apresentam propriedades distintas que se refletem em sua localização nas diferentes regiões de um espaço vetorial multidimensional. Sendo assim, inicialmente os objetos devem se auto-organizar nas diferentes regiões (através de um critério de similaridade e otimizando uma função-objetivo) para, então, ser aplicada alguma técnica de discriminação que agrupe os representantes de acordo com suas posições relativas.

As redes SOM (do inglês Self-Organizing Maps) são redes neuronais artificiais que possuem esta capacidade de auto-organização. Utilizam o paradigma de aprendizagem competitiva não-supervisionada, o treinamento é iterativo e são organizadas em duas camadas. A primeira camada, denominada camada de entrada, é composta por um vetor p-dimensional. Cada elemento deste vetor de entrada, **x**, representa um atributo dos objetos da coleção de dados. A segunda, denominada camada de saída, é um reticulado (mapa) de nós, geralmente bidimensional, totalmente conectado a todo os componentes do vetor de entrada por meio de conexões ponderadas (KOHONEN,1987). Portanto, a cada neurônio **j** há um vetor p-dimensional de pesos, $\mathbf{w_j} = [w_{j1}, w_{j2,...,} w_{jp}]$. Este vetor de pesos é conhecido como protótipo, ou vetor de código, (*prototype vector ou codebook vector).* Em cada passo do treinamento, um padrão de entrada **x** é selecionado e sua distância a cada um dos protótipos é calculada. O vencedor (BMU – *best-matching unit*) é o nó do mapa cujo vetor de pesos (protótipo) está mais próximo de **x.** A seguir os protótipos são atualizados. O BMU e seus vizinhos topológicos são aproximados a este padrão de entrada. A relação de vizinhança pode ser

expressa de diferentes formas sendo bastante popular utilizar a função gaussiana. (vide apêndice IV)

As redes SOM são consideradas robustas na manipulação de dados com ruído e apropriadas para a análise exploratória quando se desconhece antecipadamente a distribuição dos dados de entrada. No entanto, é sensível ao ajuste de parâmetros iniciais, tais como: número de ciclos, estrutura topológica da grade, função de vizinhança, taxa de aprendizado, raio de vizinhança, forma de cooperação, entre outros. Esta grande variedade de parâmetros, se não corretamente ajustados, pode levar as redes a mínimos locais, prejudicando o processo de agrupamento. (MANGIAMELI et al., 1996),(KOHONEN, 1997)

4. Avaliação da Qualidade de um Agrupamento

A variedade de métricas possíveis para agrupar os dados (vide os propostos em BEZDEK (1998) e MILLIGAN et al, (1985)) introduz alguns questionamentos, tais como: *De que modo pode-se avaliar a "qualidade" dos grupos resultantes? Quais critérios devem ser analisados para comparar diferentes partições/grupos ou algoritmos de agrupamentos?* A área de pesquisa conhecida como avaliação de agrupamento é responsável por estudar meios de quantificar a *qualidade* (ou *validade*) de uma configuração de agrupamento (BEZERRA, MATOSO, XEXÉO, 2006).

Os principais aspectos que podem ser validados nos agrupamentos e citados em SOUTO (2007) são: (a) Determinar a tendência de agrupamento (clustering tendency) do conjunto de dados; (b) Comparar os resultados de uma análise de agrupamento com resultados externos conhecidos; (c) Avaliar o quão bem os resultados de uma análise de agrupamento ajustam-se aos dados sem usar informações externas, ou seja, utilizando as próprias instâncias do conjunto de treinamento; e (d) Comparar diversos algoritmos de agrupamento ou determinar o valor mais apropriado de algum parâmetro do algoritmo como, por exemplo, a determinação da quantidade de grupos.

Os índices de avaliação como são chamadas as medidas usadas para avaliação da qualidade de um agrupamento, são divididos, na literatura, em três grupos (BEZERRA, 2006):

- Índices Internos: usados para medir a qualidade de um agrupamento com base apenas nos dados originais. Entre eles estão: *Índice Davies-Bouldin*, *Índice Dunn*, *Silhouettes*, *Índice CDbw*, entre outros.

- Índices Externos: usados para avaliar o agrupamento gerado de acordo com uma estrutura pré-determinada e imposta ao conjunto de dados. Entre eles encontram-se *Índice Rand ajustado* (*adjusted Rand*) e *índice de Jaccard*.
- Índices Relativos: utilizados para a comparação de agrupamentos distintos e determinação de qual é o melhor de acordo com algum critério (aspecto). As medidas relativas de avaliação baseiam-se tanto em medidas internas como em medidas externas. Aqui, o objetivo é determinar qual partição entre as existentes (ou geradas) melhor se ajusta aos dados. Para isso, tanto os índices internos como os externos podem ser utilizados. A aplicação mais usual dos índices internos como índices relativos é na determinação do número *k* de sub-conjuntos na qual o espaço deve ser particionado para que melhor reflita a estrutura dos padrões de entrada. Em geral, o algoritmo é executado para valores diferentes do parâmetro *k*. Em seguida, para cada valor de *k*, calcula-se os valores do índice e o melhor número de grupos é dado pelo mínimo ou o máximo dessa função, dependendo de como o índice foi definido. A forma mais comum de utilização dos índices externos como índices relativos é para calcular a de similaridade entre duas partições/grupos. Os algoritmos de agrupamento buscam otimizar a função objetivo associada. A escolha desta função é um problema de otimização de "bom agrupamento". Para uma aplicação, um "bom grupo" pode ser definido como um grupo compacto. Neste caso, a distância intra-grupo deve ser minimizada, e os índices internos podem ser utilizados. Para outra aplicação, um "bom grupo" pode ser aquele no qual um elemento e seu vizinho mais próximo compartilham o mesmo rótulo. Estes rótulos devem pré-existir nos dados o que leva à utilização de índices externos. É intuitivo, também, que os algoritmos para a primeira classe de função objetivo produzem partições completamente distintas dos que implementam a segunda.

APÊNDICE IV

MAPAS AUTO-ORGANIZÁVEIS (SOM)

1.Considerações Gerais

Os mapas auto-organizáveis (SOM – Self Organizing Maps) pertencem a uma classe de redes neuronais artificiais conhecidas como redes competitivas. Neste tipo de rede, os neurônios (unidades básicas de processamento da rede) recebem o mesmo estímulo de entrada e competem entre si para identificar quem é o vencedor.

As redes SOM, introduzidas por Kohonen, são fortemente inspiradas na seguinte característica do córtex cerebral: funções distintas são realizadas em regiões distintas, ou seja, há um "mapa" topológico presente no córtex cerebral. Outra similaridade está no fato do aprendizado ser competitivo e não supervisionado (Kohonen, 2001). São bastante difundidas e aplicadas em diferentes áreas de conhecimento, como Engenharia, Medicina, Geoprocessamento, Bioinformática, Ecologia, entre outras e indicadas para resolver problemas não-lineares de alta dimensionalidade tais como extração de características, classificação automática de dados, classificação e reconhecimento de imagens e padrões acústicos, controle adaptativo de robôs, equalização, modulação e transmissão de sinais, mineração de dados, descoberta de conhecimento, compressão de dados, análise de dados multivariados, imputação de valores ausentes, entre outros (CASTRO; CASTRO, 2001).

Pela diversidade das áreas de aplicação, observa-se que este modelo tem sido amplamente utilizado quando se deseja encontrar estruturas inesperadas no problema modelado e que não são representáveis pelos métodos estatísticos tradicionais (GIRAUDEL, LEK, 2001).

Em resumo, a rede SOM é uma rede competitiva capaz de mapeamentos que preservam a topologia entre os espaços de entrada e de saída e refletem os padrões significativos ou característicos dos dados de entrada.

No aprendizado não supervisionado, isto é, "aprender sem professor", não se almeja a solução ótima, mas uma solução factível e que apresente bons resultados. Neste paradigma, apenas com a apresentação dos padrões de entrada, sucessivas modificações locais e interativas dos pesos sinápticos são realizadas até que se desenvolva uma

configuração final que represente as relações existentes nos dados de entrada. Isto é possível, pois segundo a observação de TURING (1952 apud HAYKIN, 2001, p. 430): "Uma ordem global pode surgir de interações locais". A auto-organização ocorre pela fusão de muitas interações locais originalmente aleatórias entre neurônios vizinhos de onde emerge um estado de ordem global, na forma de padrões espaciais, com um comportamento coerente com o domínio de entrada. Assim sendo, no modelo SOM, o equilíbrio/estabilidade da rede surge de buscas locais durante a competição, quando um determinado grupo de neurônios sempre vence e outros não (HAYKIN, 2001).

A arquitetura de uma rede neuronal do tipo SOM é extremamente simples, consistindo apenas de duas camadas, conforme exemplificado na figura IV.1.

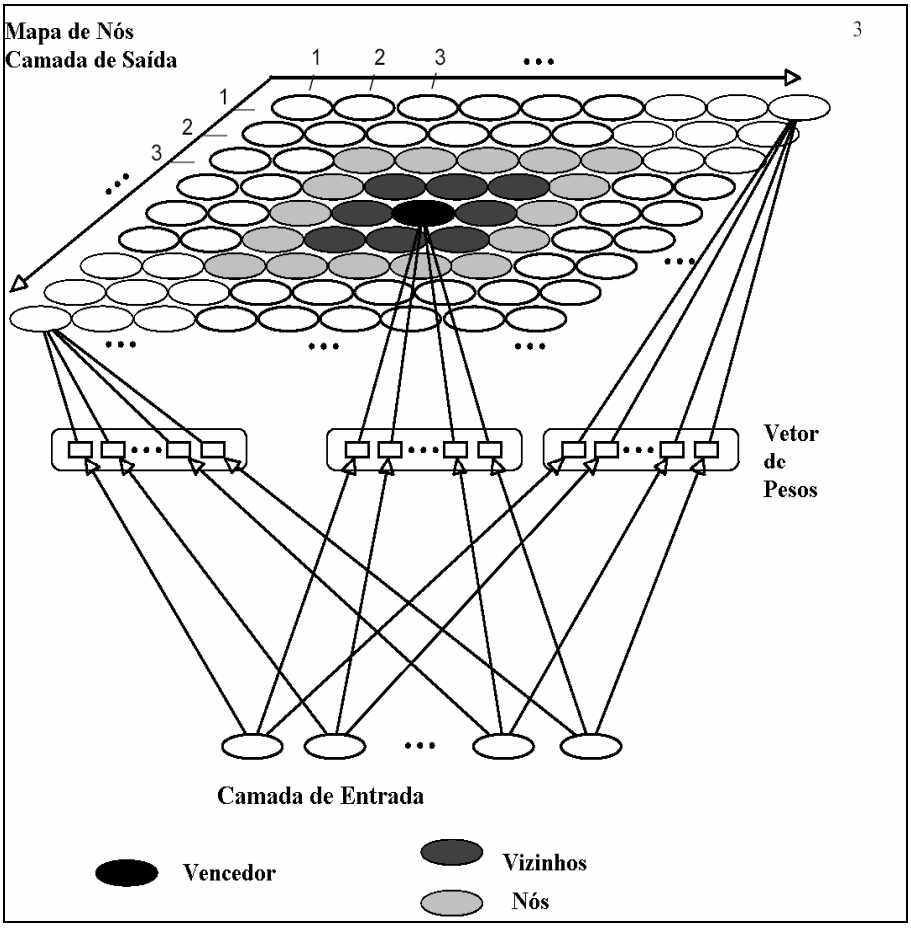


Figura IV.1: Exemplo de uma rede SOM
Adaptado de Ultsch et al ,1992

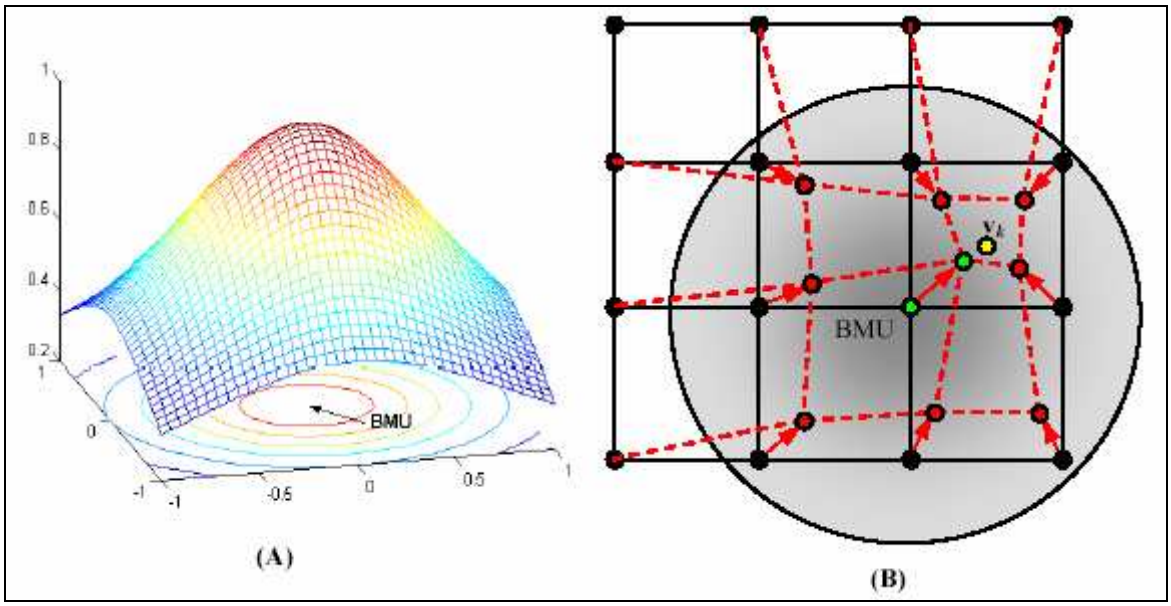


Figura IV.II: Adaptação dos pesos de um SOM após a apresentação de um padrão em $\Re^2$. Em (A) uma representação da função de vizinhança sobre um mapa bidimensional. A projeção desta adaptação mapa pode ser vista em (B).
Fonte: Adaptado de ZUCHINI, 2003

A primeira camada **I** (camada de entrada) é composta por um conjunto ordenado de ***p*** neurônios artificiais, um para cada atributo do padrão de entrada (ou de uma tupla da base de dados ou uma linha da tabela de dados **X** com p atributos). Cada neurônio de entrada está conectado a todos os neurônios da camada seguinte por meio de conexões ponderadas. A segunda camada, também conhecida como camada de Kohonen, ou camada de saída, é composta por um conjunto de neurônios, geralmente organizados na forma de um vetor ou de uma matriz e representa o mapa onde os padrões de entrada serão projetados. Todos os ***p*** componentes do vetor de entrada alimentam cada um dos *m* neurônios do mapa. Portanto, a cada neurônio **j** há associado um vetor de pesos $\mathbf{w_j} = [w_{j1},w_{j2},...,w_{jp}]$, também no espaço p-dimensional. Este vetor de pesos é conhecido como vetor de código (*codebook vector*). Os neurônios desta camada estão interconectados por uma relação de vizinhança que descreve a estrutura do mapa e competem entre si para serem ativados, de forma que apenas um seja considerado "vencedor". Quando o vencedor é determinado, seguindo o princípio de Hebb, ele e sua vizinhança são "premiados" com um reforço. O mapa pode possuir mais de duas dimensões, mas são pouco usuais e de difícil compreensão.

A relação de vizinhança e de agrupamento está ligada ao conceito de regiões do Diagrama de Voronoi. Uma definição ilustrativa do diagrama de Voronoi, apresentada

na disciplina de Geometria Computacional, do IME-USP, é a seguinte: *"considere um mapa de uma cidade onde estão marcados os locais dos vários postos de correio espalhados pela cidade. Sabe-se que existem casas por toda a parte na cidade. Como determinar qual a área de cobertura de cada um dos postos de correio, ou qual o conjunto de casas que será atendido por cada posto de correio da cidade? Ao descobrir os conjuntos de casas que estão mais próximos de cada posto de correio obtém-se a solução do problema. Mais formalmente, o problema pode ser definido como: Dado um conjunto S de n pontos no plano deseja-se determinar para cada ponto p de S qual é a região V(p) dos pontos do plano que estão mais próximos de p do que de qualquer outro ponto em S. Neste caso, o conjunto S seria o conjunto dos postos de correio, o plano seria a cidade e as regiões V(p) seria o conjunto de casas da cidade que seriam atendidos pelo posto p. As regiões determinadas por cada ponto formam uma partição do plano chamada de Diagrama de Voronoi."* Portanto, o diagrama de Voronoi de uma coleção de objetos é a segmentação do espaço em células, onde cada célula consiste de todos os pontos mais próximos de um objeto particular do que qualquer outro.

Como já salientado, os neurônios da camada de entrada estão completamente conectados aos da camada de saída. Desta forma, para cada padrão de entrada apresentado à rede, há uma região de atividade no mapa. Para garantir que este processo de auto-organização ocorra de forma apropriada, todos os neurônios da camada de saída devem ser expostos a uma quantidade suficiente de diferentes padrões de entrada, pois a localização e natureza de uma determinada região variam de um padrão de entrada para outro. Deste modo, o treinamento da rede exige várias iterações sobre os padrões de entrada. Nesta fase de treinamento, após receberem o estímulo do padrão de entrada, os neurônios competem entre si para se tornarem ativos e vencer a competição. São considerados vencedores os neurônios mais similares ao padrão de entrada de acordo com alguma medida de distância, normalmente a distância euclidiana:

$$\| x - w_c \| = \arg\min_i \{ \| x - w_i \| \}$$

onde ||.|| em geral representa a distância Euclidiana

O vencedor da iteração **t**, também conhecido como BMU (B*est Match Unit),* tem seus pesos adaptados, tornando-se mais representativo deste padrão de entrada de acordo com a seguinte equação:

$$w_i(t+1) = w_i(t) + \eta(t) h_{ji}(t)[x(t) - w_i(t)]$$

onde t indica a iteração do processo de treinamento, x(t) é o padrão de entrada, $\eta(t)$ é a taxa de aprendizado e $h_{ji}(t)$ é a relação de vizinhança.

Os neurônios situados na vizinhança do vencedor também podem ter seus pesos atualizados de acordo com a equação acima. Há duas estratégias básicas para ajuste de pesos de acordo com a relação de vizinhança:

- *winner-takes-all*: apenas um neurônio se torna ativo e tem seus pesos adaptados, ou seja, o mais similar ao padrão de entrada "leva tudo". Este tipo de competição pode ser implementado estabelecendo conexões laterais inibitórias entre os neurônios da camada de saída.
- *vizinhos topológicos*: o BMU e os nós localizados em sua vizinhança têm seus pesos ajustados. A vizinhança é estabelecida por uma função responsável por controlar o nível de atuação dos neurônios em torno do neurônio vencedor do processo competitivo. Seguindo o modelo neurobiológico, o nível de atuação dos vizinhos diminui na medida em que o mesmo se distancia do BMU. A forma e raio desta vizinhança são parâmetros da rede e uma relação bastante usual é a gaussiana.

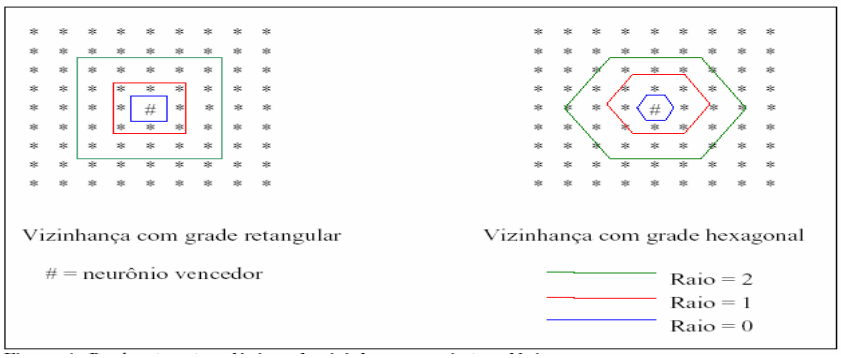


Fig. IV.3: Exemplo de parâmetros topológicos de vizinhança e raio topológico
Fonte: (FRANCISCO, 2004)

Há diversas variantes do modelo SOM, mas, em termos gerais, o algoritmo de treinamento de uma rede SOM pode ser descrito como:

Entrada: conjunto X de objetos,

Parâmetros da rede: taxa de aprendizado, raio de vizinhança, tipo de topologia, etc.

Saída: K grupos com N objetos da coleção original X

Algoritmo:

Inicializar pesos e parâmetros. Os pesos podem ser inicializados randomicamente, utilizando os elementos do próprio conjunto de dados ou então linearmente

Repita

Para cada padrão x Є X´ selecionado aleatoriamente

Definir o nó vencedor (competição)

Definir a vizinhança do nó vencedor (colaboração)

Atualizar os pesos deste nó e de seus vizinhos (Adaptação)

Ajustar taxa de aprendizado e raio de vizinhança

Até o mapa não mudar ou atingir um número máximo de iterações

As propriedades envolvendo a convergência do algoritmo de treinamento estão demonstradas em ZHEN-PING et al. (1993).

Os três processos do treinamento do SOM são detalhados nas seções seguintes.

→ Competição

Seja *p* a dimensão do espaço de entrada. Seja **x** o um padrão de entrada (vetor p-dimensional) selecionado aleatoriamente do espaço de entrada, denotado por:

$$\underline{x} = [x_1, x_2, ..., x_p]^T$$

Seja $\underline{\mathbf{w}_j}$ o vetor p-dimensional de pesos sinápticos do neurônio **j**, também conhecido como vetor de código, denotado por:

$$\mathbf{w_j} = [w_{j1,}\ w_{j2,...,}\ w_{jp].}^{\ T}$$

onde j = 1,2,3,...,*m e m* é o número total de neurônios da camada de saída, usualmente organizados em uma matriz.

Sabendo que para determinar o vetor de pesos sinápticos $\mathbf{w_j}$ mais próximo do padrão de entrada $\underline{\mathbf{x}}$ são utilizadas medidas de distâncias (em geral a distância Euclidiana) e considerando "estar mais próximo", o neurônio com menor distância (critério de mínima distância), o processo de competição é dado por:

$$\mathbf{i}(\underline{x}) = \text{argmin} \, \| \underline{x} - \underline{w}_j \|, \text{ para } j = 1,2,3,...,m$$

onde $\mathbf{i}(\underline{x})$ é o índice do neurônio mais próximo de $\underline{\mathbf{x}}$

Portanto, como é possível observar, a norma euclidiana entre o padrão de ativação de entrada e os vetores de pesos associados aos neurônios da camada de saída é responsável pelo mapeamento do padrão de entrada no espaço discreto de neurônios de saída, ou seja, no neurônio i do mapa cujo vetor de pesos associado é o de menor distância ao padrão de entrada. Esta busca pelo protótipo de menor distância resume o processo de competição. Dependendo da aplicação, pode-se devolver o índice no mapa do neurônio vencedor **i** (e utilizar, por exemplo, para classificação ou agrupamento) ou o vetor de pesos do vencedor (e utilizar, por exemplo, este protótipo para imputação ou como centróide de agrupamentos).

→ Cooperação

Inspirado na neurobiologia, o neurônio vencedor interage cooperativamente com seus vizinhos, de forma graduada. Com a vizinhança imediata, a interação é bem maior do que com os vizinhos distantes. Desta forma, o neurônio vencedor localiza-se no centro de uma vizinhança topológica cooperativa que decai com a distância lateral.

Seja:

- h_{ji} a vizinhança topológica. O centro é neurônio vencedor i, circundado por pelo conjunto de neurônios excitados cooperativos. Um neurônio deste conjunto é denotado por j.
- d_{ij} a distância lateral entre o neurônio vencedor i e seu vizinho excitado j.

A vizinhança topológica h_{ji} é uma função unimodal (com um único mínimo ou máximo) da distância lateral d_{ij}, tal que satisfaça duas exigências distintas (CASTRO; CASTRO, 2001):

- A vizinhança topológica h_{ji} é simétrica ao redor do ponto máximo definido por $d_{ij}=0$, ou seja, no neurônio vencedor i.
- A amplitude da vizinhança topológica h_{ji} decresce monotonicamente com o aumento da distância lateral d_{ij}, decaindo a zero na medida em que $d_{ij} \to \infty$ Esta é a condição necessária para convergência.

As funções da figura 2.18 satisfazem o critério de vizinhança topológica e podem ser utilizadas. A função gaussiana, item (b) da figura, é a mais usual. A função gaussiana é invariante à translação e a rede SOM com este tipo de função de vizinhança converge mais rapidamente (LO et al., 1991; ERWIN et al.,1992; LO et al., 1993).

Por estas propriedades, a função gaussiana: $h_{ji}(t) = e^{-d_{ji}^2 / 2\sigma_t^2}$ também foi escolhida neste trabalho, conforme será explicitado na abordagem proposta. Na equação acima, *t* representa o número de iterações, σ representa o raio da vizinhança topológica e o grau de participação dos neurônios vizinhos do BMU no processo de aprendizagem adaptativa. A função σ_t deve ser monotonicamente decrescente em função do tempo para que o raio da vizinhança decaia durante a aprendizagem, permitindo uma sintonia "mais fina" após algum tempo de treinamento. Uma das funções mais utilizadas para que o raio decaia ao longo do tempo de treinamento, ou seja, que modela a dependência de σ em relação ao tempo *t* é a exponencial descrita por:

$$\sigma(t) = \sigma_0 e^{-n/T_1}; t = 0,1,2,\ldots$$

onde, σ_0 é o valor do raio σ na inicialização do algoritmo e T_1 é uma constante.

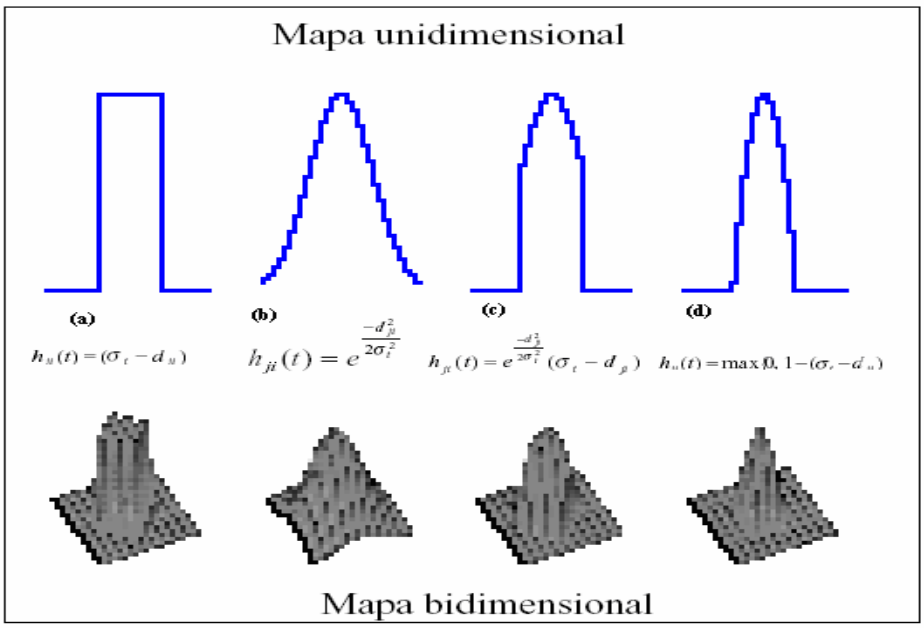


Figura IV.III: Diferentes funções de vizinhança
Fonte: VESANTO et al., 2000b, p.10)

→ Adaptação

A atualização dos pesos sinápticos é a última etapa na auto-organização da rede SOM. Nesta fase, o vetor de pesos, w_j de cada neurônio *j* considerado ativo nesta iteração (ou seja o vencedor e sua vizinhança topológica) é modificado para ficar mais próximo do padrão de entrada.

Como já dito, o aprendizado na rede SOM é inspirado no córtex cerebral e deriva do postulado de Hebb que observou que quando um impulso passa pela célula nervosa, "facilita" a passagem dos próximos impulsos. Portanto, em concordância com este princípio, quando dois neurônios conectados são ativados simultaneamente, o peso da conexão entre eles deve ser aumentado, ou seja, a cada apresentação deste padrão, a saída deve ficar "mais ativa". A maioria das regras de aprendizagem deriva da regra de Hebb (FAUSETT,1994).

Para não levar os pesos sinápticos à saturação, o ajuste de pesos da rede SOM inclui um termo de esquecimento, e é dado por:

$$\Delta \underline{w}_j = \underbrace{\alpha\, y_j\, \underline{x}}_{\text{Termo Hebbiano}} - \underbrace{g(y_j)\underline{w}_j}_{\text{Termo de Esquecimento}}$$

onde α é a taxa de aprendizagem do algoritmo.

O primeiro e o segundo termo da função são respectivamente o termo Hebbiano e o termo de esquecimento. O termo de esquecimento, $g(y_j)$ deve ser uma função escalar positiva da resposta y_j. Portanto, escolhe-se para $g(y_j)$ uma função linear: $g(y_j) = \alpha y_j$.

A variação do peso pode ser, também, denotada por:

$$\Delta \underline{w}_j = w_{j(novo)} - w_{j(velho)}$$

Assim, $w_{j(novo)} = w_{j(velho)} + \eta\, y_j (\underline{x} - w_{j(velho)})$.

Como $y_j = h_{j,i(\underline{x})}$ e realizando as substituições adequadas, tem-se

$$\Delta \underline{w}_j = \eta\, h_{ji(\underline{x})} \left(\underline{x} - \underline{w}_j\right).$$

Portanto, o vetor de pesos do neurônio j no tempo t+1, $w_j(t+1) = w_j(t) + \Delta w_j$ e dado por:

$$w_{ji}(t+1) = w_{ji}(t) + \alpha(t)\, h(t) \left[x_{ik}(t) - w_{ij}(t)\right]$$

A equação é responsável pelo movimento do vetor de pesos sinápticos w_i do neurônio vencedor i na direção do vetor de entrada x. Após diversas passagens no conjunto de treinamento, os vetores de pesos sinápticos tendem a seguir a distribuição existente nos vetores de entrada. A ordenação topológica do mapa reflete as características no espaço de entrada, pois os neurônios que são adjacentes na grade tenderão a ter vetores de pesos sinápticos similares (HAYKIN, 2001).

Para compreender a regra de aprendizado competitivo pode-se analisa-la vetorialmente. O neurônio vencedor, $\mathbf{u_i}$, é aquele com maior nível de ativação. Como o nível de ativação de um neurônio i é o produto do vetor de entrada **x** por seu vetor de pesos $\mathbf{w_i}$. ($\mathbf{u_i} = \mathbf{w_i} \cdot \mathbf{x}$) que pode ser escrito como:

$$u_i = \|w_i\| \|x\| \cos\theta,$$

onde θ é o ângulo entre os vetores $\mathbf{w_i}$ e **x.**

Para vetores de entrada normalizados, um valor grande de $\mathbf{u_i}$ indica que **x** está na vizinhança de $\mathbf{w_i}$ e um valor pequeno que os vetores são quase perpendiculares entre si,

conforme ilustra a figura IV.4 para um padrão bidimensional e três neurônios e vetores de peso normalizados.

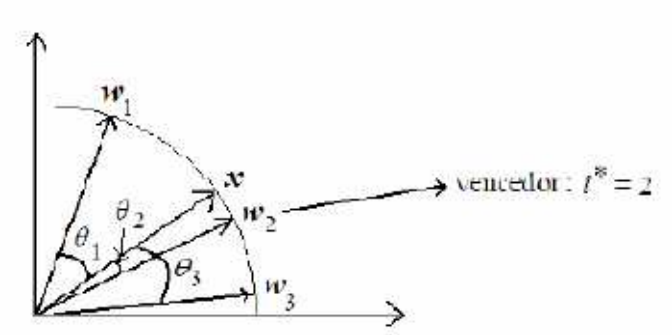


Figura IV.4 Processo de escolha do neurônio vencedor
Fonte: (SASSI, 2006)

O neurônio vencedor *i* é o que tiver o valor máximo de $\mathbf{w}_i \cdot \mathbf{x}$. que é equivalente a minimizar a distância euclidiana entre **x** e $\mathbf{w}_i$ (só para entradas normalizadas). Após a determinação do neurônio vencedor, seu vetor de pesos é atualizado pela regra: $\Delta w_{i^*} = \eta(x - w_{i^*})$. que significa que o vetor de pesos do neurônio *i* deve ser proporcionalmente (por η) alterado na direção de $\boldsymbol{x}$- $\mathbf{w}_i$, ou seja que este seja puxado na direção do vetor **x**, conforme mostra figura IV.5.

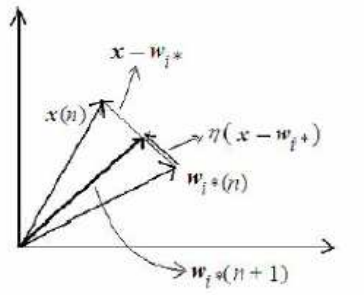


Figura IV.5 Alteração do vetor de pesos do neurônio vencedor
Fonte: (SASSI, 2006)

→ Variantes do SOM

O algoritmo básico da rede SOM possui inúmeras variantes, tanto na forma de treinamento como no posicionamento dos neurônios no mapa de saída. Os fatores comuns a todas estas variantes é a existência do conjunto de neurônios do mapa de saída, que pode ser vista como uma coleção de vetores protótipos e uma relação de vizinhança entre eles. Os protótipos são iterativamente ajustados para se adequarem aos padrões de entrada e as relações de vizinhança são usadas para que se tornem similares entre si, representando a ordenação topológica. Alguns exemplos de variantes são:

- **Batch map:** ou processamento em lote. Nesta variante, o algoritmo de aprendizado é, também, iterativo. No entanto, a taxa de aprendizado α não é usada e a atualização dos vetores de pesos é realizada após uma época, ou seja, após a apresentação de todos os padrões de entrada do conjunto de treinamento. Esta característica, atualizar os pesos após uma época, que batizou o algoritmo e libera o treinamento da apresentação aleatória dos padrões. A cada passo, os padrões de entrada são divididos de acordo com as regiões de Voronoi, seguindo o critério de proximidade entre os padrões e os vetores de peso. Após a apresentação completa do conjunto de treinamento, os vetores de pesos *w* são atualizados de acordo com a equação abaixo (VESANTO, 2000):

$$w_i(t+1) = \frac{\sum_j^m h_{ji}(t)\, s_j(t)}{\sum_j^m n_{V_j} h_{ji}(t)}$$

onde h_{ji} é a função de vizinhança, n_{Vi} é o número de amostras do conjunto de Voronoi do neurônio i e s_i representa o somatório dos padrões da região de Voronoi V_i, ou seja:

$$s_i(t) = \sum_j^{n_{V_i}} x_j$$

Segundo Costa (1999) e Vesanto (2000), a ausência do parâmetro de aprendizado α e a independência do resultado quanto à ordem de apresentação são fatores importantes para o bom desempenho desta variante. O algoritmo de treinamento em lote de uma rede SOM pode ser descrito como:

Entrada: conjunto X de objetos,

Parâmetros da rede: taxa de aprendizado, raio de vizinhança, tipo de topologia, etc.

Saída: K grupos com N objetos da coleção original X

Algoritmo:

Inicializar pesos linearmente e parâmetros.

Repita

Para cada padrão x Є X´

Definir o nó vencedor (competição)

Definir a vizinhança do nó vencedor (colaboração)

Calcule a contribuição parcial do padrão de entrada ao BMU e seus vizinhos

°

Atualizar os pesos dos nós e de seus vizinhos

Ajustar taxa de aprendizado e raio de vizinhança

Até o mapa não mudar ou atingir um nº máximo de iterações (época)

Pelas razões encontradas na literatura e descritas acima, o processamento em lote foi o modo utilizado neste trabalho.

- ***Tree-Structured SOM***: é formada por um conjunto de camadas, onde cada uma é uma quantização completa dos padrões de entrada, ou seja, é composta por diversas SOM organizadas em forma de árvore. A principal diferença entre as camadas é a quantidade de protótipos, que cresce exponencialmente em direção às folhas. Por exemplo, se o nível raiz (primeiro nível, L=*0*) tem 1 protótipo, o segundo nível, *L=1,* tem 4 neurônios no caso bi-dimensional ou dois neurônios no uni-dimensional e assim sucessivamente, conforme mostrado na figura IV.6 Portanto, cada neurônio de um nível tem seus próprios descendentes no nível abaixo, ou seja, tem sue próprio subgrupo de dados. Os

subgrupos formam um grupo cujo centróide é o vetor de pesos w_b.do *bmu b*. O centróide do nível 0 representa a média de todos os dados.

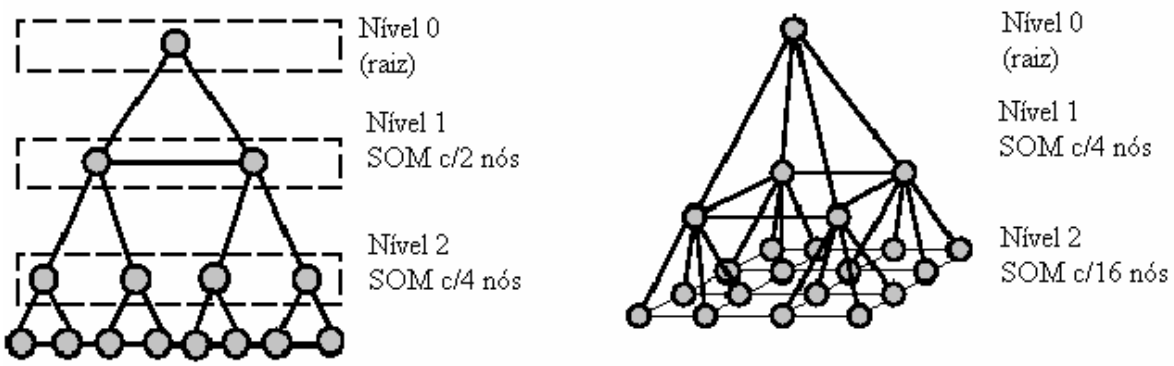


FIG. IV.6 - TS- Som uni e bi-dimensional
Adaptado de (HÄKKINEN, 2001)

Os níveis superiores são utilizados no treinamento dos níveis inferiores. Por exemplo, em vez de comparar o padrão de entrada com todos os vetores protótipos da camada 3, ele é primeiro comparado com os protótipos da camada 1. A seguir é comparado com os descendentes do(s) vencedor(es) da primeira camada e seus vizinhos e assim sucessivamente. O treinamento usualmente é realizado em lote. Durante cada época, os *BMUs* são procurados para todos os vetores de dados via pesquisa na árvore e os novos centróides w_b^{new} são computados usando a regra:

$$\mathbf{w}_b(t+1) = \frac{1}{N_b + \sum_{i \in N_c(b)} \alpha N_i} \left(N_b \mathbf{w}_b^{new} + \sum_{i \in N_c(b)} \alpha N_i \mathbf{w}_i^{new} \right)$$

onde $N_c(b)$ são os vizinhos de *b e* N_i é o numero de registros do grupo *i*.

Uma das vantagens desta arquitetura é a redução significativa do número de cálculo de distâncias, especialmente nas camadas inferiores. A inclusão sucessiva das camadas também é uma boa característica, pois permite um mapeamento gradualmente mais detalhado do mapa de entrada (KOIKKALAINEN, 1995, VESANTO 2001).

- ***Neural gas***: Nesta variante os vizinhos são adaptativamente definidos durante o treinamento. A vizinhança é definida pelo "*ranking*" resultante da ordenação dos protótipos em relação à distância de seus pesos ao padrão de entrada apresentado. (MARTINEZ et al, 1993,VESANTO, 2000).

- ***Growing Cell Structures*: A**qui, o mapa de protótipos é adaptado não só quanto à vizinhança como também quanto à quantidade de nós. De acordo com um critério de erro o mapa cresce, ganhando um novo nó, ou decresce, perdendo um nó. Quando um novo nó é introduzido no mapa, a vizinhança é redefinida (FRITZKE,1994), (VESANTO, 2000).

A rede SOM tem sido utilizada em todas as etapas da mineração de dados, até mesmo para a classificação, embora nesta tarefa não tenha o melhor desempenho, uma vez que pelo próprio paradigma não-supervisionado, a classe disponível associada ao padrão de entrada não é utilizada.(KOHONEN,1995). Na mineração de dados, o analista de dados precisa interpretar os resultados de forma eficiente e com isso o pós-processamento da rede SOM torna-se uma tarefa importante (VESANTO, 2000). A seguir é mostrado como utilizar redes SOM em algumas destas etapas:

- **Redução dos Dados:** A rede SOM pode ser utilizada tanto para a seleção de dados como para reduzir a dimensionalidade dos padrões de entrada. Em vez de usar todos os dados do conjunto original, após o treinamento da rede, pode-se capturar apenas um percentual das amostras no conjunto de Voronoi de cada unidade. (VESANTO 2000).
- **Discretização:** Em função da camada de saída ser discreta, a SOM encaixa-se naturalmente para definir os compartimentos ("*binning*"). Como mais protótipos estão posicionados nas áreas onde há maior quantidade de dados, o "*binning*" pode ser visto como a equalização de um histograma multidimensional. (VESANTO,2000)
- **Transformação de valores simbólicos em numéricos:** pela incapacidade da rede SOM usar variáveis simbólicas durante o treinamento, após o treinamento pode-se analisar suas posições no mapa. O BMU da cada entrada, utilizando apenas as variáveis numéricas é rotulado com os valores simbólicos e a distribuição dos valores no mapa é analisada. Se valores simbólicos distintos são mapeados em diferentes áreas do mapa e as coordenadas no mapa podem substituir os valores simbólicos correspondentes.
- **Substituição de valores ausentes:** O algoritmo de treinamento da rede SOM é robusto em relação a valores ausentes. Na determinação do *BMU*, somente os valores preenchidos são usados para o cálculo das distâncias. Os valores a serem imputados, então, podem ser obtidos dos protótipos dos *BMUs*. Este

processo corresponde ao uso de estimativas condicionais (ou média) como substituição dos valores ausentes, onde a estimativa é baseada nos valores preenchidos (SAMAD et al, 1991). O valor que substituirá o ausente pode ser calculado de diversas formas, mas em geral é utilizado o protótipo do *BMU ou* a média dos protótipos do *BMU* e seus vizinhos. Encontra-se na literatura alguns trabalhos relacionados entre eles podemos citar (HÄKKINEN,2001), (FESSANT F., MIDENET S., 2002)(WONG,WONG e WOOD, 2007)(PIELA, P. , LAAKSONEN S.,2001)

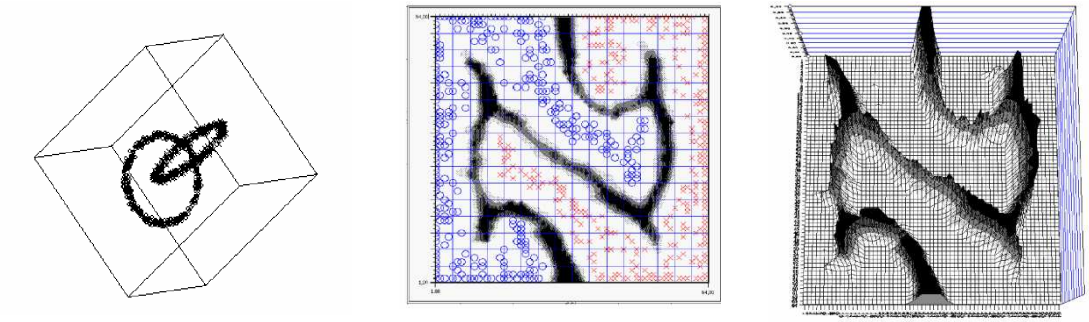
2. Visualização da Rede SOM

O objetivo da visualização de redes SOM é apresentar uma grande quantidade de informação, o mais detalhada possível, para se ter uma idéia qualitativa das propriedades dos dados, ou seja, *para que um conhecimento possa emergir a partir dos dados* (TUFTE, 1983).

Há diversas técnicas para visualizar o mapa resultante do treinamento de redes SOM. Baseado na meta desejada, as técnicas de visualização podem ser divididas em três grandes classes (ULTSCH,2002): visualização das formas e estruturas dos grupos, visualização dos componentes e visualização dos dados no mapa:

- **Visualização das Formas e Estruturas dos Grupos**: Normalmente a visualização inicia tentando responder às perguntas: “Existem grupos? Caso existam, como estão inter-relacionados? Existem dependências significativas entre as variáveis?“. Portanto, o objetivo é fornecer uma “explicação visual” da forma do mapa em relação ao espaço de entrada. Dentre estes métodos de visualização, podemos destacar a U-matrix, proposta por A. Ultsch (1993), que permite que as relações topológicas existentes no mapa sejam detectadas visualmente. Para isto, constrói uma imagem f(x,y) onde as coordenadas dos pixels equivalem às coordenadas dos neurônios no *grid* do mapa e a intensidade de cada pixel na imagem corresponde à distância entre os pesos sinápticos de neurônios adjacentes. Interpretando uma imagem como uma função tridimensional em que o valor do pixel na coordenada (x; y) é representado por um ponto na coordenada z, tem-se uma superfície em 3D cuja topografia revela a configuração dos neurônios obtida pelo treinamento. Vales, neste relevo topográfico, são regiões de neurônios similares, enquanto que

montanhas refletem a dissimilaridade entre neurônios vizinhos e podem ser interpretados como fronteiras (bordas) de agrupamentos de neurônios. A figura IV.7 ilustra a visualização dos clusters gerados por uma rede SOM para a base de dados *Chain Link* em 2-D e 3-D.



Base de Dados "Chain link" **U-matrix 2-D** **U-matrix 3-D**

Figura IV.7. Visualização de uma Rede SOM aplicada à base "Chain Link"
Fonte: (ULTSCH, 1992)

A **Matriz de Distâncias** é uma técnica bastante comum de visualização. As distâncias entre unidades vizinhas são armazenadas em uma matriz. Esta matriz pode armazenar todas as distâncias entre os nós do mapa e seus vizinhos, como o faz a U-Matrix (ULTSCH,1990) ou apenas um valor para cada nó do mapa. Como exemplo, a média das distâncias entre os vizinhos (KRAAIJVELD,1995) encontra-se ilustrada na figura IV.8 itens (a) e (b). Como os nós vizinhos no mapa estão usualmente tão próximos quanto os dados no espaço de entrada, estas matrizes estão relacionadas ao agrupamento "*single-linkage*". As matrizes de distância mostram a densidade de protótipos nas diferentes partes do mapa. Os **mapas coloridos** são técnicas relacionadas à matriz de distâncias. Aqui, unidades similares no mapa são coloridas com a mesma cor, conforme ilustrado na figura IV.8 item (c). O método introduzido por Kaski (1997) leva em consideração, se possível, as matizes das cores para refletir as distâncias no espaço original de dados. São úteis não só para visualização como na criação de um código de cores considerando as estruturas dos clusters. Esta codificação pode ser utilizada para unir diferentes visualizações. Outro modo de visualizar é mostrar o número de "acertos" de cada nó e considerar como bordas para o diagrama de Voronói onde há muito poucos hits ou mesmo nenhum (VESANTO e ALHONIEMI,2000).

- **Componentes do mapa:** Após a idéia holística da forma do conjunto de dados, pode ser desejado ater-se a detalhes e responder a perguntas, tais como: “Que tipos de valores as variáveis possuem? Quais valores ou combinação de valores são típicos nos diferentes grupos? Existem dependências significativas entre as variáveis?”. As técnicas de visualização desta classe têm como objetivo responder tais perguntas. Em geral, mostram os planos dos componentes, conforme ilustra a figura IV.9. Cada plano pode ser visto como uma fatia do mapa, correspondente a um componente do vetor em todos os nós do mapa, e expressam o espalhamento dos valores deste componente (podem ser interpretados como histogramas mas com a diferença que o mesmo valor pode estar presente em diferentes clusters se forem características a diversos clusters distintos). Sobrepostos com a visualização dos agrupamentos, os valores das variáveis em cada grupo podem ser identificados. Correlações podem ser detectadas observando estes planos, e comparando-os entre si. A correlação é observada por padrões similares em posições idênticas nos planos dos componentes, embora esta técnica não tenha sido projetada para tal tarefa (VESANTO e AHOLA apud VESANTO, 2000). Uma propriedade interessante de agrupamentos é tentar descobrir “o que faz um grupo”. Métodos para mostrar a contribuição das variáveis originais à estrutura do grupo foram desenvolvidos por Kaski et al. (1998)
- **Dados no mapa: A**qui o objetivo é descobrir a localização de um determinado exemplo (ou amostra) no mapa ou resposta do mapa para uma determinada amostra. Aplicável quando a tarefa de mineração desejada é classificação, reconhecimento de padrão ou descoberta de conhecimento

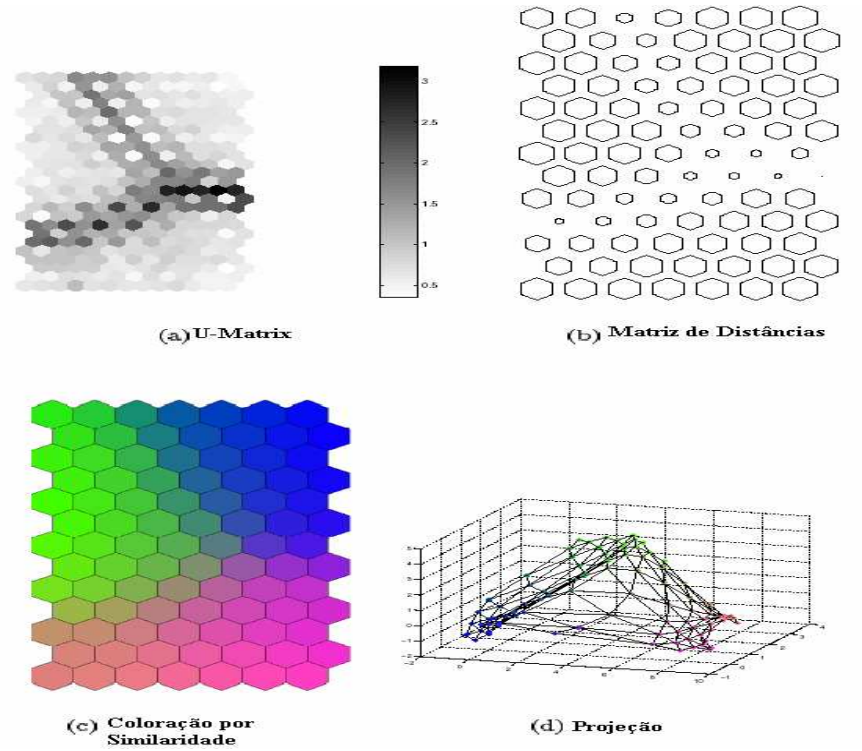


Figura IV.8: Visualização de Agrupamentos: (a) U-matrix com escala de cinza indicando os valores, (b) matriz de distãncias, usando tamanho de hexágonos para mostrar os valores, (c) coloração por similaridade, (d) a rede do mapa em um espaço tri-dimensional. As cores de (c) e (d) são equivalentes.
Fonte: (VESANTO, 2000)

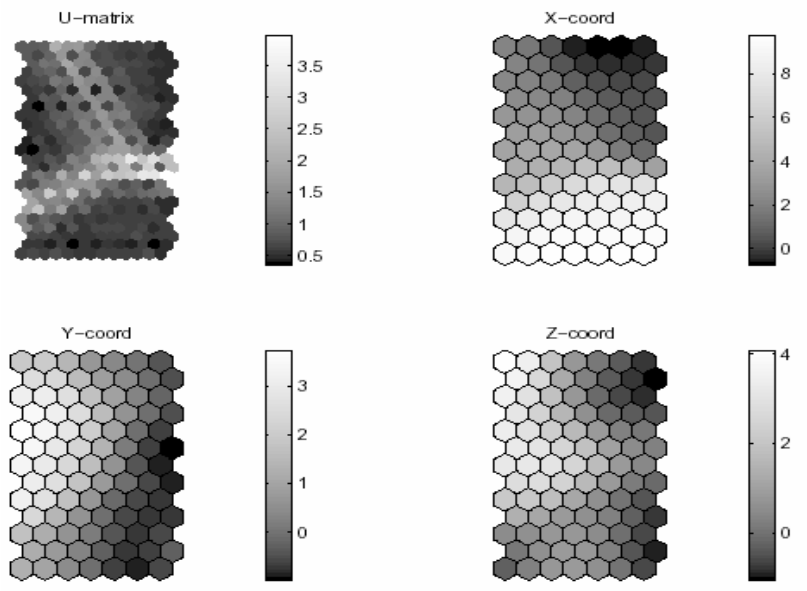


Figura IV.9: A U-Matrix e seus três planos componentes.
Fonte: (VESANTO, 2000)

- **Dados no mapa: A**qui o objetivo é descobrir a localização de um determinado exemplo (ou amostra) no mapa ou resposta do mapa para uma determinada amostra. Aplicável quando a tarefa de mineração desejada é classificação, reconhecimento de padrão ou descoberta de conhecimento.

→ Segmentação do Mapa

Como a rede SOM neste trabalho é uma das etapas para a imputação de valores ausentes na base de dados, não é desejado visualizar o mapa gerado, mas segmentá-lo automaticamente para obter os agrupamentos equivalentes nos dados de entrada. Para esta tarefa de segmentação, analisou-se os métodos Costa (COSTA, 1999), Vesanto (VESANTO, 2000) e Costa-Netto (COSTA,NETTO, 2003). Embora o método proposto por Costa-Netto seja aplicável apenas em mapas bidimensionais, esta restrição não afeta a decisão de projeto tomada, pois esta foi a topologia escolhida. Portanto, em função de ser um método conceitualmente simples, baseado apenas nas informações contidas na rede após o treinamento, completamente automático e aplicável a mapas com diferentes topologias optou-se pelo algoritmo Costa-Netto (COSTA, NETTO, 2003). Este algoritmo foi aplicado com sucesso em alguns trabalhos, dentre os quais podem ser citados Silva et al (2004) e Gonçalves et al.,(2005).

A proposta de Vesanto e Alhoniemi (2000) determina os agrupamentos em dois passos. A rede SOM é treinada e, a seguir, aplica-se o K-*Means* ou um método hierárquico aglomerativo nos protótipos para a descoberta dos agrupamentos. O índice Davies-Bouldin (explicado na seção 2.4.4.1) é utilizado como critério para a fusão ou separação dos grupos. Portanto, a SOM é um redutor do tamanho do conjunto de dados a ser analisado. Um inconveniente deste método é a necessidade da intervenção do usuário e, portanto, não ser totalmente automático como desejado.

Para um mapa treinado o algoritmo Costa-Netto pode ser definido como:

Entrada: Mapa de nós da Rede SOM,

Saída: K grupos com N objetos da coleção original X

Algoritmo:

1. Obter as distâncias entre os pesos de neurônios adjacentes *i* e *j*, d(w_i,w_j); e a atividade de cada neurônio *i*, *H(i).*
2. Para cada par de neurônios adjacentes *i e j*, a aresta será considerada inconsistente caso:
    1. a distância entre os pesos excede em 2 a distância média dos outros neurônios adjacentes a i ou a j;
    2. os dois neurônios adjacentes *i* e *j* possuem atividade *(H)* abaixo de 50% do mínimo permitido (H_{min}), ou um dos neurônios for inativo (*H(i) = 0*); $H_{min} = \omega H_{med}$, sendo que $0.1 \leq \omega \leq 0.6$ e $H_{med} = n/m$.
    3. a distância entre os centróides dos conjuntos de dados associados aos neurônios *i* e *j* exceder em 2 vezes a distância entre os pesos *d(wi, wj)*.
3. Remoção das arestas inconsistentes. Para cada aresta*(i,j)* considerada inconsistente, a matriz de Adjacência **A,** que representa o grafo, armazena uma conexão nula na pos (i,j) , caso contrário, a pos (i,j) de **A** armazena 1.
4. Atribuir um código distinto para cada conjunto de neurônios conectados.
5. Remover grupos conectados pequenos (com menos de 3 neurônios).

O que acontece com a aplicação do algoritmo é uma poda dos neurônios conectados adjacentes. Ou seja, ao final têm-se vários grupos de neurônios conectados representando um agrupamento específico, conforme ilustrado na figura IV.10.

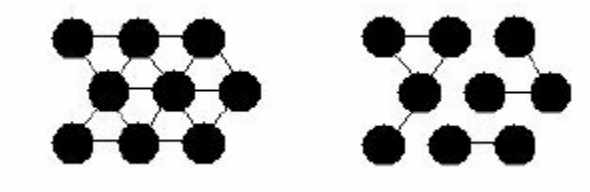


FIG. IV.10: Eliminação das conexões pelo algoritmo Costa-Netto
Fonte: (COSTA e NETTO, 2003)

O algoritmo proposto por Costa e Netto (2003) não depende da dimensão do mapa nem da U-Matrix o que o torna mais genérico que a proposta de segmentação baseada na U-matrix (COSTA,1999).

O algoritmo, como pode ser observado, utiliza alguns limiares empíricos que foram definidos por experimentações. Monta os grupos usando somente informações contidas no mapa treinado, como a distância entre os neurônios, erro de quantização e nível de atividade. No entanto, não garante que todos os vetores de entrada serão rotulados. Por exemplo, dados atípicos podem não ser rotulados devido à restrição do passo 5. Este problema foi solucionado usando-se o critério do vizinho mais próximo para rotulação de todos os neurônios “especializados” do mapa.