Seleção de atributos via agrupamento

Thiago Ferreira Covões

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Seleção de atributos via agrupamento ¹

Thiago Ferreira Covões

Orientador: *Prof. Dr. Eduardo Raul Hruschka*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos necessários à obtenção do título de Mestre em Ciências da Computação e Matemática Computacional.

USP - São Carlos
Janeiro/2010

¹Trabalho Realizado com Suporte Financeiro da FAPESP, Processo nº 2009/03580-2

À minha família

Agradecimentos

Ao meu orientador professor Eduardo Raul Hruschka, pessoa que tenho o prazer de trabalhar já há cerca de cinco anos. Pessoa não apenas responsável pela minha iniciação no mundo da pesquisa, mas também por diversos ensinamentos tanto profissionais quanto pessoais. Dono de um vasto conhecimento, grande senso de humor e imensa humildade, é, não apenas um excelente profissional, mas também, um grande amigo.

À minha família, meus pais José e Rosana, minha irmã Thais e meu sobrinho Matheus, por todo o apoio durante todos os meus anos de vida, e por estarem sempre positivamente presentes.

À minha namorada, Fernanda, pela sua paciência e grande apoio durante toda essa etapa da minha vida.

Gostaria também de agradecer aos amigos do BIOCOM, em especial a Danilo Horta, Débora Medeiros, Jonathan Andrade, Lucas Vendramin, Luiz Coletta, Márcio Basgalupp, Murilo Naldi, Pablo Andretta e Ricardo Cerri, pelos momentos de discussão sobre assuntos relacionados a este trabalho, pelos momentos de companheirismo e confraternização e por estarem sempre dispostos a ajudar. Agradeço também aos colegas do LABIC.

Agradeço ao professor Estevam Rafael Hruschka Júnior pelo auxílio na realização do estudo de caso com os dados do projeto Read the Web. Agradeço também ao professor Leandro Nunes de Castro e Helder Knidel pela oportunidade de atuar no projeto CIAC.

A todos os funcionários do ICMC da USP, pela competência e dedicação.

À FAPESP pelo apoio financeiro para a realização deste trabalho.

Resumo

O avanço tecnológico teve como consequência a geração e o armazenamento de quantidades abundantes de dados. Para conseguir extrair o máximo de informação possível dos dados tornou-se necessária a formulação de novas ferramentas de análise de dados. Foi então introduzido o Processo de Descoberta de Conhecimento em Bancos de Dados, que tem como objetivo a identificação de padrões válidos, novos, potencialmente úteis e compreensíveis em grandes bancos de dados. Nesse processo, a etapa responsável por encontrar padrões nos dados é denominada de Mineração de Dados. A acurácia e eficiência de algoritmos de mineração de dados dependem diretamente da quantidade e da qualidade dos dados que serão analisados. Nesse sentido, atributos redundantes e/ou não-informativos podem tornar o processo de mineração de dados ineficiente. Métodos de Seleção de Atributos podem remover tais atributos. Nesse trabalho é proposto um algoritmo para seleção de atributos e algumas de suas variantes. Tais algoritmos procuram identificar redundância por meio do agrupamento de atributos. A identificação de atributos redundantes pode auxiliar não apenas no processo de identificação de padrões, mas também pode favorecer a compreensibilidade do modelo obtido. O algoritmo proposto e suas variantes são comparados com dois algoritmos do mesmo gênero descritos na literatura. Tais algoritmos foram avaliados em problemas típicos de mineração de dados: classificação e agrupamento de dados. Os resultados das avaliações mostram que o algoritmo proposto, e suas variantes, fornecem bons resultados tanto do ponto de vista de acurácia como de eficiência computacional, sem a necessidade de definição de parâmetros críticos pelo usuário.

Abstract

The technological progress has lead to the generation and storage of abundant amounts of data. The extraction of information from such data has required the formulation of new data analysis tools. In this context, the Knowledge Discovery from Databases process was introduced. It is focused on the identification of valid, new, potentially useful, and comprehensible patterns in large databases. In this process, the task of finding patterns in data is usually called Data Mining. The efficacy and efficiency of data mining algorithms are directly influenced by the amount and quality of the data being analyzed. Redundant and/or uninformative features may make the data mining process inefficient. In this context, feature selection methods that can remove such features are frequently used. This work proposes a feature selection algorithm and some of its variants that are capable of identifying redundant features through clustering. The identification of redundant features can favor not only the pattern recognition process but also the comprehensibility of the obtained model. The proposed method and its variants are compared with two feature selection algorithms based on feature clustering. These algorithms were evaluated in two well known data mining problems: classification and clustering. The results obtained show that the proposed algorithm obtained good accuracy and computational efficiency results, additionally not requiring the definition of critical parameters by the user.

Esta dissertação foi preparada com o formatador de textos $\text{\LaTeX}$. O estilo utilizado no documento foi desenvolvido por Ronaldo Cristiano Prati. A bibliografia foi gerada automaticamente pelo $\text{\BibTeX}$, utilizando o estilo *Chicago* com modificações para o português.

Algumas palavras utilizadas neste trabalho não foram traduzidas da língua inglesa para a portuguesa por serem amplamente conhecidas e difundidas na comunidade acadêmica.

Sumário

Lista de Abreviaturas	xvii
Lista de Figuras	xix
Lista de Tabelas	xxi
Lista de Algoritmos	xxv
1 Introdução	1
1.1 Considerações Iniciais	1
1.2 Agrupamento	3
1.3 Classificação	4
1.4 Maldição da Dimensionalidade	5
1.5 Objetivos	6
1.6 Organização do Trabalho	7
2 Seleção de Atributos	9
2.1 Considerações Iniciais	9
2.2 Busca por Subconjuntos de Atributos	10
2.3 Abordagens para Avaliação de Subconjuntos de Atributos	11
2.4 Métodos de Seleção de Atributos	12
3 Filtros Baseados em Agrupamento de Atributos	15
3.1 Considerações Iniciais	15
3.2 Fundamentação Teórica	15
3.3 Trabalhos Relacionados	19
3.3.1 Filtro proposto por Mitra, Murthy e Pal (MMP)	19
3.3.2 Filtro proposto por Au et al. (ACA)	21
3.3.3 Filtro Silhueta Simplificada (SSF)	23

3.4	Modificações para o Filtro Silhueta Simplificada (SSF)	26
3.4.1	Medidas de correlação	27
3.4.2	Critério de seleção de atributos	28
3.5	Considerações Finais	30
4	Resultados Experimentais em Problemas de Classificação	31
4.1	Considerações Iniciais	31
4.2	Metodologia de Avaliação	31
4.3	Algoritmos e Parâmetros Utilizados	34
4.4	Bases de Dados	35
4.5	Resultados e Discussão	37
4.5.1	Comparações entre SSF e MMP	37
4.5.2	Comparações entre SSF e ACA	43
4.5.3	Comparações entre Variantes Supervisionadas do SSF	46
4.6	Estudos de Caso	48
4.6.1	Classificador Inteligente de Amostras de Café	48
4.6.2	<i>Read the Web</i>	52
5	Resultados Experimentais em Problemas de Agrupamento de Dados	55
5.1	Considerações Iniciais	55
5.2	Metodologia de Avaliação	55
5.3	Algoritmos e Parâmetros Utilizados	57
5.4	Bases de Dados	58
5.5	Resultados e Discussão	61
5.5.1	Comparações entre SSF e MMP	62
5.5.2	Comparações entre SSF e ACA	66
6	Conclusões e Trabalhos Futuros	69
6.1	Principais Contribuições	69
6.2	Conclusões	70
6.3	Trabalhos Futuros	71
	Referências Bibliográficas	73
A	Tabelas dos Resultados dos Experimentos de Classificação	85
B	Tabelas dos Resultados dos Experimentos de Agrupamento de Dados	95

Lista de Abreviaturas

ACA Algoritmo de Agrupamento de Atributos

ARI *Adjusted Rand Index*

CART *Classification and Regression Trees*

CBE *Consistency Based Evaluation*

CFS *Correlation based Feature Selection*

CJ Coeficiente de Jaccard

FCBF *Fast Correlation Based Filter*

KDD *Knowledge Discovery in Databases*

KNN *k-Nearest Neighbors*

KS *k-medoids* considerando supervisão

IMCN Informação Mútua Condicionada Normalizada

ICMI Índice de Compressão Máxima de Informação

RL Regressão Logística

MD Mineração de Dados

MMP Algoritmo de Mitra, Murthy e Pal

MRI Medida de Redundância de Interdependência

PKID *Proportional k-Interval Discretization*

NB *Naïve Bayes*

SA Seleção de Atributos

SS Silhueta Simplificada

SSF Filtro Silhueta Simplificada

SU *Symmetrical Uncertainty*

SU_S *Symmetrical Uncertainty* considerando supervisão

SVM *Support Vector Machine*

WEKA *Waikato Environment for Knowledge Knowledge Analysis*

WRP *Wrapper*

Lista de Figuras

1.1	Exemplos de relações entre atributos e número de grupos em problemas de agrupamento de dados.	6
2.1	Subconjuntos de atributos possíveis considerando $M=3$	10
2.2	Abordagens fundamentais para avaliação de subconjuntos de atributos.	12
3.1	Grupos de atributos: medóides representados por losangos; atributos de fronteira representados por triângulos.	29
4.1	Uma iteração de validação cruzada com a seleção de atributos integrada.	32
4.2	Avaliação de seleção de atributos multicritério.	33
4.3	Resultados (erro de classificação - validação cruzada) obtidos nas bases Bio1, ..., Bio5 e Yeast utilizando o algoritmo Algoritmo de Mitra, Murthy e Pal (MMP) para cada valor do parâmetro k	39
4.4	Resultados (erro de classificação - validação cruzada) obtidos nas bases de dados Iono, Pima, Wisc, Spam, Colon e Leu utilizando o algoritmo MMP para cada valor do parâmetro k	40
4.5	Visualização dos grupos de atributos obtidos pelo SSF-R-2 na base de dados do Projeto “Read the Web”.	54
5.1	Base de dados 10_250.	59
5.2	Base de dados 12_200.	60
5.3	Base de dados 20_250.	60
5.4	Base de dados 1000_1000.	61
5.5	Índices externos (<i>Adjusted Rand Index</i> (ARI) e Coeficiente de Jaccard (CJ)) obtidos nas bases de dados artificiais utilizando o algoritmo MMP para cada valor do parâmetro k	63
5.6	Índices externos (ARI e CJ) obtidos nas bases de Bio1,...,Bio5 e Yeast utilizando o algoritmo MMP para cada valor do parâmetro k	64

Lista de Tabelas

1.1	Base de dados representada no formato atributo-valor.	2
3.1	Exemplo do XOR Correlacionado.	17
4.1	Valores críticos (q_α) para o teste de Nemenyi.	34
4.2	Acrônimos utilizados nas tabelas de resultados.	36
4.3	Características das bases de dados utilizadas nos experimentos.	37
4.4	Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o número de atributos selecionados entre SSF- λ , SSF- ρ e MMP (Melhor caso). .	38
4.5	Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o erro obtido pelo classificador <i>Naïve Bayes</i> (NB) utilizando os subconjuntos de atributos selecionados pelos algoritmos SSF- λ , SSF- ρ e MMP (Melhor caso e caso médio).	41
4.6	Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o erro obtido pelo classificador <i>k-Nearest Neighbors</i> (KNN) utilizando os subconjuntos de atributos selecionados pelos algoritmos SSF- λ , SSF- ρ e MMP (Melhor caso e caso médio).	41
4.7	Avaliação multicritério (Seção 4.2) considerando o erro obtido pelo classificador NB utilizando os atributos selecionados pelos algoritmos SSF- λ , SSF- ρ e MMP (Melhor caso).	41
4.8	Avaliação multicritério considerando o erro obtido pelo classificador KNN utilizando os atributos selecionados pelos algoritmos SSF- λ , SSF- ρ e MMP (Melhor caso).	42
4.9	Erros de classificação — média (desvio padrão) — obtidos pelo SSF- ρ -2 e MMP selecionando (aproximadamente) o mesmo número de atributos.	43
4.10	Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o número de atributos selecionados entre SSF-R, SSF- <i>Symmetrical Uncertainty</i> (SU) e Algoritmo de Agrupamento de Atributos (ACA).	43

4.11	Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o erro obtido pelo classificador NB utilizando os subconjuntos de atributos selecionados pelos algoritmos SSF-R, SSF-SU e ACA.	43
4.12	Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o erro obtido pelo classificador KNN utilizando os subconjuntos de atributos selecionados pelos algoritmos SSF-R, SSF-SU e ACA.	44
4.13	Avaliação multicritério (Seção 4.2) considerando o erro obtido pelo classificador NB utilizando os atributos selecionados pelos algoritmos SSF-R, SSF-SU e ACA.	44
4.14	Avaliação multicritério (Seção 4.2) considerando o erro obtido pelo classificador KNN utilizando os atributos selecionados pelos algoritmos SSF-R, SSF-SU e ACA.	45
4.15	Erros de classificação — média (desvio padrão) — obtidos pelo SSF-R-2 e ACA-2 utilizando valor de k definido pelo SSF-R-2.	45
4.16	Erros de classificação — média (desvio padrão) — obtidos pelo SSF-R-2 e ACA-2 utilizando valor de k definido pelo ACA-2.	46
4.17	Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o número de atributos selecionados entre as variantes supervisionadas do SSF e o SSF-SU.	46
4.18	Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando a taxa de erro obtida pelo NB entre as variantes supervisionadas do SSF e o SSF-SU.	47
4.19	Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando a taxa de erro obtida pelo KNN entre as variantes supervisionadas do SSF e o SSF-SU.	47
4.20	Avaliação multicritério (Seção 4.2) considerando o erro obtido pelo classificador NB utilizando os atributos selecionados pelas variantes supervisionadas do SSF e o SSF-SU.	48
4.21	Avaliação multicritério (Seção 4.2) considerando o erro obtido pelo classificador KNN utilizando os atributos selecionados pelas variantes supervisionadas do SSF e o SSF-SU.	48
4.22	Erros de classificação — média (desvio padrão) — obtidos na base de dados do Projeto CIAC considerando a classificação em 2 classes.	50
4.23	Erros de classificação — média (desvio padrão) — obtidos na base de dados do Projeto CIAC considerando a classificação em 13 classes.	51
4.24	Avaliação multicritério (Seção 4.2) dos resultados na base de dados do Projeto CIAC.	51
4.25	Erros de classificação — média (desvio padrão) — obtidos na base de dados do Projeto “ <i>Read the Web</i> ”.	53
4.26	Avaliação multicritério (Seção 4.2) dos resultados na base de dados do Projeto “ <i>Read the Web</i> ”.	53

5.1	Tabela de contingência para duas partições.	56
5.2	Sumários das bases de dados de agrupamento utilizadas.	61
5.3	Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando os valores de ARI obtido pelos algoritmos SSF- λ , SSF- ρ e MMP.	65
5.4	Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando os valores de CJ obtido pelos algoritmos SSF- λ , SSF- ρ e MMP.	65
5.5	Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o número de atributos selecionados pelos algoritmos SSF- λ , SSF- ρ e MMP.	65
5.6	Índices externos (ARI e CJ) obtidos pelo SSF- ρ -1 e MMP selecionando (aproximadamente) o mesmo número de atributos.	65
5.7	Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando os valores de ARI obtidos pelos algoritmos ACA, SSF-R e SSF-SU.	66
5.8	Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando os valores de CJ obtidos pelos algoritmos ACA, SSF-R e SSF-SU.	66
5.9	Vitórias/Empates/Derrotas para algoritmos da 1ª coluna considerando o número de atributos selecionados pelos algoritmos ACA, SSF-R e SSF-SU.	66
5.10	Índices externos (ARI e CJ) obtidos pelo SSF-SU-2 e ACA-2 utilizando o valor de k definido pelo SSF-SU-2.	67
5.11	Índices externos (ARI e CJ) obtidos pelo SSF-SU-2 e ACA-2 utilizando o valor de k definido pelo ACA-2.	67
A.1	Resultados obtidos — Bio1.	86
A.2	Resultados obtidos — Bio2.	86
A.3	Resultados obtidos — Bio3.	87
A.4	Resultados obtidos — Bio4.	87
A.5	Resultados obtidos — Bio5.	88
A.6	Resultados obtidos — Yeast.	88
A.7	Resultados obtidos — Ionosphere.	89
A.8	Resultados obtidos — Pima.	89
A.9	Resultados obtidos — Wisconsin.	90
A.10	Resultados obtidos — Spambase.	90
A.11	Resultados obtidos — Colon Cancer.	91
A.12	Resultados obtidos — Leukemia.	91
A.13	Tempos computacionais (em segundos) dos experimentos de classificação. Computador: Opteron 2GHz com 8Gb de RAM.	92
A.14	Avaliação multicritério — erro obtido pelo classificador NB.	93
A.15	Avaliação multicritério — erro obtido pelo classificador KNN.	93
B.1	Sumário do número de atributos selecionados nos problemas de agrupamento.	95
B.2	Sumário dos valores de ARI obtidos.	96

B.3	Sumário dos valores de CJ obtidos.	96
B.4	Tempos computacionais (em segundos) dos experimentos de agrupamento de dados. Computador: Opteron 2GHz com 8Gb de RAM.	96

Lista de Algoritmos

1	Algoritmo de Mitra, Murthy e Pal (MMP)	20
2	Algoritmo <i>k-modes</i>	22
3	Filtro Silhueta Simplificada (SSF)	25

Introdução

1.1 Considerações Iniciais

Com a evolução da tecnologia nas últimas décadas, a velocidade de processamento e capacidade de armazenamento aumentaram de maneira significativa. Como efeito colateral, têm sido gerados, coletados e processados dados em quantidades abundantes. Exemplos disso são a introdução de códigos de barras em produtos comerciais, a informatização de transações comerciais, o armazenamento de imagens recuperadas de satélite e a popularização da *World Wide Web*. Tal abundância de dados torna inviável a utilização de métodos tradicionais de análises de dados (Fayyad et al., 1996b). Surgiu, então, uma necessidade de técnicas que possam, de forma eficaz e automática, analisar grandes conjuntos de dados, em busca de conhecimento.

O processo pelo qual se busca extrair conhecimento de um conjunto de dados é usualmente chamado de descoberta de conhecimento em bancos de dados — *Knowledge Discovery in Databases* (KDD). Este processo, interativo e iterativo, incorpora conhecimento de domínio e interpretação de resultados, com ênfase na aplicação dos métodos de Mineração de Dados (MD) (Fayyad et al., 1996b). Pode-se definir a descoberta de conhecimento em bancos de dados como sendo o processo não trivial de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis em grandes bancos de dados (Piatetsky-Shapiro e Frawley, 1991). O processo de descoberta de conhecimento em bancos de dados vem sendo aplicado em diversos domínios da ciência e da indústria. Algumas das aplicações descritas na literatura envolvem a automação de análise e catalogação de imagens de estrelas e galáxias (Fayyad et al., 1996a) e mapeamento de genes (Sevon et al., 2005). Diversas aplicações em outros domínios podem ser encontradas em Kantardzic e Zurada (2005).

A descoberta de conhecimento em bases de dados pode ser dividida nas seguintes etapas (Han e Kamber, 2000):

1. **Limpeza de dados:** remover ruídos e dados inconsistentes;
2. **Integração de dados:** combinar múltiplas fontes de dados;
3. **Seleção de dados:** dados relevantes à tarefa de análise são recuperados da base de dados;
4. **Transformação dos dados:** os dados são transformados ou consolidados para a mineração;
5. **Mineração de Dados (MD):** métodos computacionais são aplicados para extrair padrões dos dados;
6. **Avaliação dos padrões:** identificar padrões realmente interessantes representando conhecimento, baseando-se em alguma medida de interesse;
7. **Representação de conhecimento:** visualização e técnicas de representação de conhecimento são usadas para apresentar o conhecimento minerado para o usuário.

As quatro primeiras etapas são usualmente chamadas de *pré-processamento de dados*, enquanto que as duas últimas são normalmente denominadas de *pós-processamento*. O termo **MD** é comumente utilizado no lugar do termo **KDD**. Conceitualmente, porém, a **MD** é frequentemente considerada uma etapa no processo **KDD**, consistindo no uso de algoritmos específicos, os quais encontram padrões sobre uma coleção de dados (Fayyad et al., 1996b). A **MD** é interdisciplinar, possuindo intersecção com áreas como aprendizado de máquina, reconhecimento de padrões, estatística e bancos de dados (Hand et al., 2001).

Os dados a serem analisados consistem em um conjunto¹ $\mathbf{X}$ de N objetos (também chamados de tuplas, registros ou exemplos), *i.e.*, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ onde cada $\mathbf{x}_i$ é um vetor de valores de um conjunto de M atributos precursores (também chamados de características ou variáveis) $A = \{A_1, A_2, \dots, A_M\}$. Usualmente os dados são representados no formato tabela atributo-valor — como na Tabela 1.1.

	A_1	A_2	$\dots$	A_M
$\mathbf{x}_1$	x_{11}	x_{12}	$\dots$	x_{1M}
$\mathbf{x}_2$	x_{21}	x_{22}	$\dots$	x_{2M}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{x}_N$	x_{N1}	x_{N2}	$\dots$	x_{NM}

Tabela 1.1: Base de dados representada no formato atributo-valor.

Os algoritmos utilizados na etapa de **MD** são usualmente categorizados pelo tipo de modelo de aprendizado que utilizam para encontrar padrões. No *aprendizado não-supervisionado* (agrupamento), apenas os dados são fornecidos para o algoritmo, *i.e.*, o conjunto $\mathbf{X}$. No *aprendizado supervisionado*, para cada objeto é fornecido também a classe à qual o objeto pertence,

¹Multi-conjuntos são usualmente permitidos.

i.e., cada objeto se torna uma tupla $\langle \mathbf{x}_i, c \rangle$, sendo c a classe do objeto, e o problema de inferi-la para objetos em que a classe não é conhecida é comumente chamado de classificação². No *aprendizado semi-supervisionado*, por sua vez, apenas para um pequeno subconjunto dos objetos são fornecidas as respectivas classes.

Os conceitos relativos ao Agrupamento e à Classificação de Dados são introduzidos nas Seções 1.2 e 1.3.

1.2 Agrupamento

Uma das habilidades mais básicas dos seres humanos envolve o agrupamento de objetos similares gerando uma categorização (Everitt, 2001). No entanto, o conceito de grupo é subjetivo; pessoas diferentes podem formar grupos diferentes com o mesmo conjunto de objetos. Dessa forma, o processo de agrupamento pode ser definido como a identificação de um conjunto de categorias (usualmente chamadas de grupos ou *clusters*) que descrevem um conjunto de dados (Fayyad et al., 1996b; Tan et al., 2005). O objetivo desta divisão é maximizar a homogeneidade entre os objetos de um mesmo grupo e, concomitantemente, maximizar a heterogeneidade entre objetos de grupos distintos. Devido ao fato de que não se conhece *a priori* o grupo ao qual cada objeto pertence, este processo pode ser caracterizado como um problema de aprendizado não-supervisionado.

Neste trabalho, são utilizados métodos que envolvem a obtenção de agrupamentos (partições) rígidos dos dados em k grupos, *i.e.*, cada objeto do conjunto de dados pertence a um único grupo. É importante ressaltar que a maioria dos algoritmos descritos na literatura assume que o valor de k é fornecido pelo usuário (Kaufman e Rousseeuw, 1990; Milligan, 1996; Theodoridis e Koutroumbas, 2006). Dessa forma, estes algoritmos se concentram em obter k grupos de objetos semelhantes de acordo com algum critério pré-estabelecido. O número de possibilidades de se agrupar N objetos em k grupos é dado por (Liu, 1968):

$$Num_Agrupamentos(N, k) = \frac{1}{k!} \sum_{i=0}^k (-1)^i \binom{k}{i} (k-i)^N. \quad (1.1)$$

Por exemplo, existem $Num_Agrupamentos(25, 5) = 2.436.684.974.110.751$ formas de agrupar 25 objetos em 5 grupos. Dessa maneira, pode-se perceber a complexidade de se agrupar corretamente uma base de dados de 25 objetos em 5 grupos. É necessário observar, ainda, que normalmente esse problema é considerado simples quando comparado com aqueles encontrados em aplicações de MD (Hruschka, 2001).

Devido ao fato de que o número de agrupamentos possíveis de N objetos em k grupos aumenta aproximadamente na razão $k^N/k!$, a tentativa de se encontrar uma solução ótima globalmente é usualmente inviável sob o ponto de vista computacional (Arabie e Hubert, 1999).

²Em aplicações onde o valor da classe é contínuo (*e.g.* determinar o valor do aluguel de um imóvel), o problema é comumente chamado de regressão.

É importante ressaltar que a definição do valor do parâmetro k não é trivial, especialmente em casos de pouco conhecimento de domínio. De acordo com Kaufman e Rousseeuw (1990), a maioria das abordagens encontradas na literatura adota critérios numéricos que determinam o número de grupos baseando-se em partições obtidas para diversos valores de k . Duas medidas comumente utilizadas são a Silhueta (Kaufman e Rousseeuw, 1990), detalhada na Seção 3.3.3.1, e o índice Dunn (Dunn, 1974). Estudos mais recentes sobre o assunto podem ser encontrados em Halkidi et al. (2001) e Vendramin et al. (2008, 2009).

Técnicas de agrupamento de dados são amplamente utilizadas para processar dados de expressão gênica (bioinformática) (Yeung et al., 2003), para categorizar documentos similares (mineração de texto) (Berry, 2003) e para encontrar perfis de consumidores (marketing) (Letrémy et al., 2007). Para um estudo amplo sobre agrupamento de dados, sugere-se ver, por exemplo Jain e Dubes (1988), Kaufman e Rousseeuw (1990) e Everitt (2001).

1.3 Classificação

A classificação de dados é um processo que consiste de duas etapas (Han e Kamber, 2000): inicialmente, um modelo (classificador) é construído a partir de um conjunto de dados, usualmente chamado de *conjunto de treinamento*. O conjunto de treinamento é formado por uma determinada quantidade de objetos, sendo que cada objeto é rotulado com uma classe $c_j \in C$, sendo $C = \{c_1, c_2, \dots, c_l\}$ o conjunto das l possíveis classes. Devido ao conhecimento *a priori* da informação das classes no conjunto de treinamento, esse processo pode ser caracterizado como um problema de aprendizado supervisionado. Na segunda etapa, o modelo construído é utilizado para inferir a classe de objetos cujas classes são desconhecidas.

Existem diversas abordagens para a construção desses modelos, dentre as quais se destacam aquelas baseadas em: distâncias entre objetos (Aha et al., 1991), regra de Bayes (Bayes, 1763) e árvores de decisão (Quinlan, 1993). Para se escolher os modelos usa-se um conceito muito difundido e conhecido como a Navalha de Occam (*Occam's razor*) (Elder e Pregibon, 1996) que sugere que entre modelos com acurácia similar, o mais simples é preferível. O problema com modelos complexos é que eles tendem a possuir um menor poder de generalização — pois estão potencialmente super-ajustados aos dados de treinamento — o que os torna menos eficazes quando utilizados para fazer previsões sobre novos dados. Tal problema é usualmente chamado de super-ajuste (*overfitting*).

Para tentar obter melhores estimativas de acurácia — diminuindo o viés (*bias*) em relação aos dados de treinamento — são utilizadas técnicas de amostragem na construção do modelo. Dentre elas, a validação cruzada de 10 pastas (*10-fold cross-validation*) é amplamente utilizada (Witten e Frank, 2005). Esta técnica consiste na divisão dos objetos rotulados em 10 pastas (subconjuntos) sem sobreposição³. Em seguida, o modelo é construído 10 vezes, sendo que em cada construção, uma das pastas é utilizada como *conjunto de teste* e as restantes como

³Um objeto pode pertencer a apenas uma pasta.

conjunto de treinamento. Cada pasta é usada uma vez como conjunto de teste. A estimativa de acurácia do modelo pode ser calculada por meio do número de objetos do conjunto de teste que o classificador construído pelo conjunto de treinamento rotula corretamente. Alternativamente, é também comum o uso da estimativa da *taxa de erro*. Usualmente, na divisão dos objetos em 10 pastas, as proporções de objetos de cada classe do conjunto completo são mantidas, *e.g.*, se no conjunto completo 30% dos objetos eram da classe c_1 , aproximadamente 30% dos objetos de cada pasta serão da classe c_1 . Esse procedimento é conhecido como estratificação. O processo final é então denominado validação cruzada estratificada em 10 pastas (*stratified 10-fold cross validation*).

Algoritmos de classificação são muito utilizados em aplicações de diagnóstico médico (Kononenko, 2001) e identificação de SPAM (Zhang et al., 2004). Para um estudo mais amplo sobre algoritmos de classificação sugere-se ver os trabalhos de Hand et al. (2001), Witten e Frank (2005) e Bishop (2007).

1.4 Maldição da Dimensionalidade

Teoricamente, seria intuitivo pensar que, quanto maior a quantidade de atributos, supostamente mais informações estariam disponíveis para o algoritmo de mineração de dados. No entanto, conforme o número de atributos cresce os dados tendem a ficar mais esparsos, dando a impressão que os objetos estão igualmente distantes (Dy, 2007). As dificuldades encontradas em espaços de muitas dimensões são sintetizadas no termo “maldição da dimensionalidade” (*curse of dimensionality*).

Para tentar solucionar ou amenizar tais problemas, pode-se usar métodos de Seleção de Atributos (SA), os quais consistem em algoritmos que buscam remover atributos redundantes e/ou irrelevantes para a tarefa de MD.

A complexidade dos modelos obtidos por meio de um algoritmo de MD, assim como o tempo de processamento necessário para obtê-los, é influenciada pelo número de atributos nas bases de dados utilizadas. Por tal razão, a SA é fundamental para a obtenção de resultados de fácil entendimento e em menor quantidade de tempo.

Mao (2005), Vellido (2006) e Hong et al. (2008) observam que embora a SA tenha sido amplamente estudada em problemas de classificação, a pesquisa sobre SA para problemas de agrupamento é relativamente recente. Uma das dificuldades na SA para problemas de agrupamento deve-se ao fato de que, diferentes subconjuntos de atributos podem indicar diferentes agrupamentos, e não há rótulos para indicar qual agrupamento seria o mais adequado (Dy e Brodley, 2004). Para tentar suprir essa limitação são usualmente utilizados critérios numéricos. No entanto, ainda não há um critério amplamente aceito na literatura.

Para ilustrar o impacto que a SA pode ter em um problema de agrupamento, considere a Figura 1.1, reproduzida de Hruschka e Covões (2005), na qual três casos distintos de relações entre dois atributos são demonstrados. Na Figura 1.1(a), o atributo y é suficiente para a formação

dos dois grupos de dados obtidos por meio de inspeção visual. O atributo x é, portanto, irrelevante. Na Figura 1.1(b), os dois grupos de dados podem ser formados/recuperados utilizando-se qualquer um dos dois atributos (x ou y) separadamente. Neste caso, os atributos são ditos redundantes. Na Figura 1.1(c), o atributo x é suficiente para descrever os três grupos de dados que são encontrados através de inspeção visual. No entanto, o atributo y permite a formação de apenas dois grupos mais compactos e separados entre si. Estes dois critérios (compactação e separação de grupos) são usualmente utilizados como indicadores de qualidade de um agrupamento. Neste caso, portanto, critérios numéricos poderiam sugerir ser mais vantajoso recuperar dois grupos, utilizando somente o atributo y , enquanto que, por inspeção visual existem nitidamente três grupos.

Figura 1.1: Exemplos de relações entre atributos e número de grupos em problemas de agrupamento de dados (Hruschka e Covões, 2005).

1.5 Objetivos

Este trabalho tem como objetivo principal estudar e desenvolver variantes do algoritmo de SA denominado de Filtro Silhueta Simplificada (SSF) (Covões et al., 2009), cujo desenvolvimento também é parte integrante do presente trabalho. O algoritmo em questão parte da premissa de que é possível agrupar atributos, selecionando posteriormente os atributos que melhor representem cada grupo, de forma a remover redundância dos dados. Neste contexto, são investigadas possíveis modificações no SSF, a saber:

- Avaliação de diferentes medidas de correlação (lineares e não-lineares);
- Incorporação de diferentes critérios de seleção de atributos a partir dos grupos obtidos.

São investigadas variantes para problemas de agrupamento (assim como a versão original do SSF) e classificação. Neste último caso, a informação da classe é incorporada no processo de agrupamento ou seleção dos atributos.

Em linhas gerais, são desenvolvidos algoritmos capazes de melhorar a qualidade dos dados disponíveis para o algoritmo de MD favorecendo, assim, a obtenção de melhores resultados.

1.6 Organização do Trabalho

O restante deste trabalho está organizado da seguinte forma:

Capítulo 2 - Seleção de Atributos (SA): neste capítulo é apresentado um estudo sobre a etapa de SA. São também apresentados os conceitos e as terminologias utilizadas neste trabalho. Por fim, é realizado um levantamento bibliográfico sobre diversos trabalhos realizados na área.

Capítulo 3 - Filtro Baseado em Agrupamento de Atributos: um arcabouço teórico para algoritmos de SA baseados no agrupamento de atributos é descrito. Trabalhos relacionados descritos na literatura também são apresentados em maiores detalhes. E, por fim, algumas variantes do algoritmo Filtro Silhueta Simplificada (SSF), que são as principais contribuições desse trabalho, são detalhadas.

Capítulo 4 - Resultados Experimentais em Problemas de Classificação: Neste capítulo são apresentados experimentos comparando o algoritmo proposto com outros algoritmos descritos na literatura para problemas de classificação em bases artificiais e reais. Também são apresentados dois estudos de caso de aplicações reais do algoritmo.

Capítulo 5 - Resultados Experimentais em Problemas de Agrupamento de Dados: Uma série de experimentos comparando o algoritmo proposto com outros algoritmos descritos na literatura para problemas de agrupamento em bases artificiais e reais são descritos nesse capítulo.

Capítulo 6 - Conclusões e Trabalhos Futuros: Neste capítulo são apresentadas as principais conclusões obtidas neste trabalho e sugestões para trabalhos futuros.

Seleção de Atributos

2.1 Considerações Iniciais

Para a redução da dimensionalidade (de atributos) duas abordagens são comumente utilizadas: seleção de atributos (SA) e transformação de atributos¹. Enquanto a SA busca eliminar atributos que não serão úteis para a tarefa de MD, a transformação de atributos busca gerar atributos mais expressivos a partir dos atributos existentes. Embora ambas permitam reduzir dimensionalidade, de um espaço M dimensional para p dimensional ($p \leq M$), contrariamente à SA, a transformação de atributos necessita de todos os atributos para novos objetos (para mapeá-los para a dimensão que está sendo utilizada), além de reduzir a capacidade de interpretação por especialistas de domínio (Diamantini e Potena, 2007). Um exemplo clássico da transformação de atributos é a Análise de Componentes Principais (PCA - *Principal Components Analysis*) (Pearson, 1901). Neste trabalho são abordados somente os métodos de SA, principalmente porque estes frequentemente produzem resultados mais compreensíveis.

Segundo Guyon e Elisseeff (2003) o objetivo da SA é a eliminação de atributos redundantes e não-informativos. A eliminação desses atributos pode trazer diversos benefícios, tais como facilitar o entendimento e a visualização dos dados, bem como reduzir o custo computacional do algoritmo de MD aplicado. Assume-se, então, que para cada base de dados existe um subconjunto A^* de seus atributos, *i.e.*, $A^* \subseteq A$, que melhor caracteriza a base de dados. No entanto, para uma base de dados com M atributos existem $O(2^M)$ subconjuntos de atributos possíveis. Portanto, uma busca exaustiva pelo melhor subconjunto de atributos é normalmente inviável sob o ponto de vista computacional. Um exemplo descrevendo os subconjuntos possíveis de atributos considerando $M=3$ é apresentado na Figura 2.1. Neste exemplo, no topo é indicado

¹Também conhecida como *feature extraction*.

Figura 2.1: Subconjuntos de atributos possíveis considerando $M=3$.

o subconjunto com todos os atributos (todos os círculos preenchidos) e por fim é indicado o subconjunto vazio (nenhum círculo preenchido).

2.2 Busca por Subconjuntos de Atributos

Devido à inviabilidade da busca exaustiva pelo subconjunto A^* , o problema de se selecionar atributos pode ser abordado via métodos de busca heurística (Langley, 1994). Nestes, inicialmente deve-se definir o ponto inicial e a direção em que será realizada a busca. Neste sentido, as abordagens mais comuns são (Han e Kamber, 2000):

- **Seleção *forward*:** a busca começa com um subconjunto vazio de atributos. A cada iteração, a adição de um novo atributo no subconjunto é realizada (possíveis abordagens para determinar qual atributo será adicionado são descritas na Seção 2.3).
- **Eliminação *backward*:** a busca começa com o conjunto completo de atributos. A cada iteração, um atributo é removido do conjunto.
- **Combinação seleção *forward* e eliminação *backward*:** os dois métodos podem ser combinados para que, a cada iteração, o procedimento adicione ou elimine atributos. Essa abordagem pode iniciar em qualquer ponto do espaço de busca.
- **Geração aleatória:** a cada iteração, um subconjunto de atributos é gerado aleatoriamente.

O próximo passo envolve a definição de como a busca será realizada (Langley, 1994), definindo-se um procedimento para se percorrer o espaço de busca. Para realizar este passo, alguns algoritmos usualmente utilizados são (Hall, 1999):

- ***Greedy hill climbing***: a cada iteração é determinado o atributo que melhora o subconjunto de acordo com algum critério, *e.g.*, a taxa do ganho de informação (Quinlan, 1993). Feita a escolha, esta não é revista no decorrer do processo.
- ***Best first***: similarmente ao *greedy hill climbing*, o próximo subconjunto a ser avaliado é gerado de acordo com uma mudança local no subconjunto atual. Porém, esse método permite que, caso o caminho percorrido gere um subconjunto pior que um subconjunto anterior, o processo seja retroagido até o subconjunto anterior, reiniciando a busca a partir deste. Para evitar que todo o espaço de busca seja percorrido, usualmente se define um limite de subconjuntos explorados que não causam melhora ao processo de seleção.
- **Algoritmos genéticos**: baseando-se na teoria da evolução de Darwin, subconjuntos são gerados aleatoriamente e sofrem modificações de acordo com uma distribuição de probabilidades. A idéia básica é favorecer uma competição entre os subconjuntos de atributos, de tal forma que melhores subconjuntos tenham maiores chances de serem escolhidos. A intenção é que as soluções encontradas (subconjuntos de atributos) evoluam de iteração para iteração. Para uma introdução à conceitualização de algoritmos genéticos pode-se ver, por exemplo, os trabalhos de Holland (1992) e Goldberg (1989).

Uma descrição de outros algoritmos de busca podem ser encontradas em (Liu e Motoda, 1998) e (Russell e Norvig, 2003).

Esses algoritmos necessitam de um critério de avaliação de subconjuntos de atributos para que subconjuntos de atributos analisados possam ser comparados, e por fim um subconjunto possa ser considerado ótimo (possivelmente ótimo local). É importante ressaltar que diferentes critérios de avaliação podem levar a diferentes subconjuntos de atributos ótimos. As possíveis abordagens para tais critérios são descritas na Seção 2.3.

2.3 Abordagens para Avaliação de Subconjuntos de Atributos

As abordagens para avaliação de subconjuntos de atributos são categorizadas de acordo com a participação do algoritmo de aprendizado na avaliação. Existem três abordagens fundamentais: *wrappers*, filtro e *embedded* (Guyon e Elisseeff, 2003).

Wrappers são algoritmos de SA que “empacotam” o algoritmo de aprendizado no seu processo de avaliar o subconjunto de atributos. Neste sentido, subconjuntos de atributos são avaliados diretamente pelo algoritmo de aprendizado em questão (*e.g.*, por meio da acurácia do modelo obtido com determinado conjunto de atributos). Estes algoritmos costumam prover bons resultados, mas são custosos computacionalmente, já que para cada subconjunto de atributos a ser avaliado deve-se aplicar o algoritmo de aprendizado a ser usado na modelagem. *Filtros* são algoritmos de SA que utilizam propriedades intrínsecas dos dados, sem utilizar o algoritmo

de aprendizado usado na modelagem para avaliar subconjuntos de atributos. Por tal motivo, são mais eficientes computacionalmente do que os *wrappers*, e geralmente mais indicados para grandes bases de dados (Kohavi e John, 1997). Duas propriedades comumente utilizadas para a avaliação de subconjuntos de atributos são a correlação entre atributos e a consistência dos dados considerando o subconjunto de atributos (este critério é descrito em detalhes na Seção 2.4). É possível combinar as abordagens *wrapper* e filtro, obtendo-se assim abordagens *híbridas* — e.g., ver Hruschka et al. (2005). Alguns algoritmos de aprendizado, tais como o C4.5 (Quinlan, 1993), possuem embutido na construção do próprio modelo um procedimento de SA. Este tipo de SA é usualmente chamada de *embedded*. A Figura 2.2 apresenta as diferentes abordagens de avaliação considerando a participação do algoritmo de aprendizado.

Figura 2.2: Abordagens fundamentais para avaliação de subconjuntos de atributos.

Para uma revisão mais profunda sobre as características da SA sugere-se ver Liu e Motoda (1998, 2007) e Guyon et al. (2006).

Com o objetivo de delimitar a área de estudo desta dissertação, a Seção 2.4 fornece uma visão geral sobre diversos métodos de SA reportados na literatura, direcionando o leitor para os trabalhos que estão mais relacionados aos filtros propostos nesta dissertação.

2.4 Métodos de Seleção de Atributos

Um estudo comparativo entre algoritmos de SA para aprendizado supervisionado e aprendizado não-supervisionado foi realizado por Liu e Yu (2005).

Conforme observado na Seção 1.4, a seleção de atributos para problemas de agrupamento de dados tem recebido maior atenção na literatura apenas recentemente. Especificamente, pode-se citar os filtros propostos por: Basak et al. (1998), Talavera (2000), Traina Jr. et al. (2000), Peña et al. (2001), Questier et al. (2002), Talavera (2005), Butterworth et al. (2005), Questier et al. (2005), Haindl et al. (2006), Yu et al. (2008) e os *wrappers*: Devaney e Ram (1997), Kim et al. (2000), Dash e Liu (2000), Dy e Brodley (2000), Law et al. (2004), Hruschka e Covões (2005), Raftery e Dean (2006), Handl e Knowles (2006), Vellido (2006), Hong et al. (2008), Chow et al. (2008). Além destes métodos baseados em filtros e *wrappers*, pode-se citar como

métodos híbridos os trabalhos de: Xing e Karp (2001), Hruschka et al. (2005, 2006b) e de Zeng e Cheung (2009). Por fim, cumpre mencionar os filtros de Mitra et al. (2002) e de Au et al. (2005) que estão intimamente relacionados ao trabalho a ser desenvolvido nesta dissertação e, por este motivo, serão revistos em maiores detalhes na Seção 3.3. Em linhas gerais, o método proposto por Mitra et al. (2002) utiliza o conceito de vizinhos mais próximos para formar grupos de atributos utilizando uma medida de correlação linear, enquanto que o método proposto por Au et al. (2005) agrupa os atributos de acordo com uma medida de correlação não-linear.

Uma revisão detalhada sobre diversos métodos de seleção de atributos em problemas de classificação foi realizado em Dash e Liu (1997). Diversos outros trabalhos de SA foram desenvolvidos mais recentemente, tais como os filtros propostos por: Lashkia e Anthony (2004), Fleuret (2004), Chow e Huang (2005), Peng et al. (2005), Lee (2005), Ouardighi et al. (2007), Song et al. (2007), Bonev et al. (2008), Liang et al. (2008), Liu et al. (2009), os *wrappers* descritos por: Kohavi e John (1997), Yang e Honavar (1998), Guyon et al. (2002), Chien e Yang (2006), Tahir et al. (2007), Krupka et al. (2008), Giralaldi et al. (2008), Draminski et al. (2008), Maldonado e Weber (2009), e os híbridos desenvolvidos por: Sebban e Nock (2002), Liu e Zheng (2006), Ng et al. (2008). Além destes métodos, Hall (1999), Liu e Setiono (1996) e Yu e Liu (2003) propuseram filtros que serão utilizados como base de comparação nos experimentos realizados nessa dissertação, e, por este motivo, serão aqui abordados em maiores detalhes.

Hall (1999) propôs um algoritmo que avalia subconjuntos de atributos, considerando como bons subconjuntos aqueles que possuem atributos altamente correlacionados com a classe e que, ao mesmo tempo, apresentem baixa correlação entre si. A correlação entre atributo-atributo e atributo-classe é calculada utilizando a medida *Symmetrical Uncertainty* (SU), descrita na Seção 3.4.1. Liu e Setiono (1996) utilizam um critério de consistência para avaliar subconjuntos de atributos. Neste critério, dois objetos são ditos inconsistentes se eles diferem apenas pelo valor da classe. Considerando i como o número de objetos idênticos (independentemente da classe), a contagem de inconsistência pode ser realizada por i menos o número de objetos da classe majoritária entre os i objetos. Finalmente, a taxa de inconsistência é a soma das contagens de inconsistência dividida pelo número de objetos. Os subconjuntos são gerados pelo algoritmo de forma aleatória. Por essa razão, quanto maior o tempo de execução, melhor o resultado obtido. Yu e Liu (2003) descrevem um algoritmo que também utiliza a medida SU para encontrar atributos irrelevantes e redundantes. Atributos irrelevantes são definidos como atributos que possuem correlação com a classe menor do que um limiar definido pelo usuário. Após remover os atributos irrelevantes, os atributos redundantes são definidos como aqueles que possuem correlação com a classe menor do que com os outros atributos.

Filtros Baseados em Agrupamento de Atributos

3.1 Considerações Iniciais

Neste capítulo serão abordados métodos para seleção de atributos que utilizam o conceito de agrupamento de atributos. Para tal, considere uma base de dados formada por N vetores $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, na qual cada $\mathbf{x}_i$ é um vetor de características descritas por M medidas de um conjunto de atributos $A = \{A_1, A_2, \dots, A_M\}$. Levando-se em conta a matriz transposta $\mathbf{X}'$ e uma medida de correlação (*e.g.*, correlação de Pearson), podem ser formados grupos de atributos, de modo que atributos mais correlacionados entre si pertençam ao mesmo grupo (*cluster*). Partindo-se desta premissa, na Seção 3.2 é introduzido um arcabouço teórico para algoritmos de agrupamento de atributos. Na Seção 3.3 são descritos alguns dos trabalhos existentes na literatura que utilizam essa abordagem, especificamente: Filtro proposto por [Mitra et al. \(2002\)](#); Filtro proposto por [Au et al. \(2005\)](#), e o Filtro Silhueta Simplificada (SSF) ([Covões et al., 2009](#)), principal objeto deste trabalho. Na Seção 3.4 são apresentadas as variantes propostas ao SSF.

3.2 Fundamentação Teórica

Considere inicialmente um problema de classificação. Seja $C = \{c_1, \dots, c_l\}$ o conjunto das l possíveis classes. Um classificador construído a partir do conjunto de treinamento recebe como entrada um objeto novo e o classifica em uma destas l possíveis classes. Na teoria, idealmente este objeto deveria fornecer condições para se determinar sua classificação correta. No entanto, na prática, esse é raramente o caso devido ao fato de que, entre outras razões,

frequentemente não se dispõe de todos os atributos necessários para se tomar uma decisão determinística. Além disso, conforme discutido anteriormente, bases de dados reais podem conter atributos irrelevantes e/ou redundantes. Neste cenário, usualmente uma distribuição de probabilidades que modela a função de classificação é usada. Especificamente, assume-se que todos os dados são gerados por uma distribuição de probabilidades sobre um espaço de vetores de atributos. Neste contexto, para cada conjunto de valores $\mathbf{x}_j$ de A existe uma distribuição de probabilidade que pode ser denotada de maneira simplificada por $P(C|A = \mathbf{x}_j)$, onde $A = \mathbf{x}_j$ representa $A_1 = x_{j1} \wedge A_2 = x_{j2} \wedge \dots \wedge A_M = x_{jM}$, para qualquer $j \in \{1, 2, \dots, N\}$.

De acordo com [Yu e Liu \(2004\)](#) e [Koller e Sahami \(1996\)](#), o objetivo da seleção de atributos pode ser formalizado como a seleção do subconjunto mínimo $R \subseteq A$ de forma que $P(C|R)$ é igual ou o mais próxima possível de $P(C|A)$ ¹. $P(C|R)$ é a distribuição de probabilidades para as diferentes classes, dados os valores dos atributos em R , e $P(C|A)$ é a distribuição original, dados os valores dos atributos em A . É importante lembrar que, na prática, essas distribuições de probabilidades obtidas são aproximações das distribuições reais, estimadas por meio de amostras da população. Feita essa observação, as definições 1, 2 e 3 de (ir)relevância usadas por [Kohavi e John \(1997\)](#) e [Yu e Liu \(2004\)](#) são formalizadas na sequência. Seja S_i o subconjunto de atributos obtido de A pela remoção de A_i , i.e., $S_i = A - \{A_i\}$ e, $S_i = \mathbf{s}_{ji}$ qualquer combinação de valores para todos os atributos em S_i para um vetor $j \in \{1, \dots, N\}$.

Definição 1. *Relevância forte.* Um atributo A_i é fortemente relevante se, e somente se, para algum $j \in \{1, \dots, N\}$ existem x_{ji} e $\mathbf{s}_{ji}$ de tal forma que $P(C|S_i = \mathbf{s}_{ji}) \neq P(C|A_i = x_{ji}, S_i = \mathbf{s}_{ji})$.

Definição 2. *Relevância fraca.* Um atributo A_i é fracamente relevante se, e somente se, ele não é fortemente relevante e para algum $j \in \{1, \dots, N\}$ existem x_{ji} e $\mathbf{s}_{ji}$ de tal forma que $P(C|S_i = \mathbf{s}_{ji}) = P(C|A_i = x_{ji}, S_i = \mathbf{s}_{ji})$ e $\exists S'_i \subset S_i$, de tal forma que $P(C|S'_i = \mathbf{s}'_{ji}) \neq P(C|A_i = x_{ji}, S'_i = \mathbf{s}'_{ji})$.

Definição 3. *Irrelevância.* Um atributo A_i é irrelevante se, e somente se, $\forall S'_i \subseteq S_i$, $P(C|S'_i = \mathbf{s}'_{ji}) = P(C|A_i = x_{ji}, S'_i = \mathbf{s}'_{ji})$.

Em outras palavras, um atributo é fortemente relevante se a sua remoção afeta a distribuição de probabilidades de classes original. Um atributo é fracamente relevante se a sua remoção não afeta a distribuição de probabilidades de classes original, mas este afeta a distribuição de probabilidades de algum subconjunto dos atributos. Finalmente, a definição de irrelevância sugere que o atributo não é realmente necessário.

Para exemplificar as definições, será utilizado o problema do XOR Correlacionado ([Kohavi e John, 1997](#)). Considere $A_1, \dots, A_5$ como atributos booleanos. Os possíveis objetos são tais que $A_2 = \bar{A}_4$ e $A_3 = \bar{A}_5$. Existem apenas 8 objetos possíveis e assume-se que eles têm a mesma probabilidade de ocorrência. O conceito alvo é

¹Quando não causar confusão, por conveniência e simplicidade $P(C|A_1 = x_{j1} \wedge A_2 = x_{j2} \wedge \dots \wedge A_M = x_{jM})$, para qualquer $j \in \{1, \dots, N\}$ será representada por $P(C|A)$.

$$Classe = A_1 \text{ XOR } A_2.$$

É possível verificar que existe um conceito alvo equivalente $Classe = A_1 \text{ XOR } \bar{A}_4$. Os exemplos possíveis e suas respectivas classes são apresentados na Tabela 3.1

Objetos	A_1	A_2	A_3	A_4	A_5	Classe
x_1	1	1	1	0	0	0
x_2	1	1	0	0	1	0
x_3	1	0	1	1	0	1
x_4	1	0	0	1	1	1
x_5	0	1	1	0	0	1
x_6	0	1	0	0	1	1
x_7	0	0	1	1	0	0
x_8	0	0	0	1	1	0

Tabela 3.1: Exemplo do XOR Correlacionado.

Considerando, por exemplo, o objeto x_1 é possível verificar que o atributo A_1 é fortemente relevante (Definição 1) pois $P(C|S_1 = s_{11}) = (\frac{1}{2}, \frac{1}{2})$ para as classes 0 e 1 respectivamente, e $P(C|A_1 = x_{11}, S_1 = s_{11}) = (1, 0)$ e portanto são diferentes. Considerando o mesmo objeto (x_1) também é possível verificar que o atributo A_2 é fracamente relevante (Definição 2) pois: (i) $P(C|S_2 = s_{12}) = (1, 0)$ e $P(C|A_2 = x_{12}, S_2 = s_{12}) = (1, 0)$ e portanto são iguais; (ii) considerando o subconjunto $S'_4 \subset S_2$, *i.e.* $S'_4 = A - \{A_2, A_4\}$, $P(C|S'_4 = s'_{14}) = (\frac{1}{2}, \frac{1}{2})$ e $P(C|A_2 = x_{12}, S'_4 = s'_{14}) = (1, 0)$. De forma análoga, pode-se verificar que A_4 também é fracamente relevante. Os atributos A_3 e A_5 são irrelevantes pois seus valores não influenciam a distribuição de probabilidades independentemente do subconjunto de atributos considerado; tal fato é justificado pela ausência de ambos os atributos no conceito alvo original e equivalente. Portanto, para este exemplo existem dois subconjuntos de atributos ótimos, a saber: $A^* = \{A_1, A_2\}$ ou $A^* = \{A_1, A_4\}$, conforme esperado pela própria definição do problema.

Yu e Liu (2004) definem o objetivo da seleção de atributos como a seleção do subconjunto de atributos R que inclui todos os atributos fortemente relevantes, um subconjunto dos atributos fracamente relevantes e nenhum atributo irrelevante. Os autores argumentam que, dentre os atributos fracamente relevantes existem atributos redundantes que podem ser identificados e removidos (como visto no exemplo anterior). É amplamente aceito que atributos redundantes são atributos que possuem (completa) correlação entre seus valores (Yu e Liu, 2004). Como consequência, este trabalho considera que a redundância entre atributos pode ser definida como:

Definição 4. *Redundância.* Dois atributos A_i e A_j são redundantes se, e somente se, eles são completamente correlacionados.

A Definição 4 é a base de métodos que utilizam o conceito de agrupamento de atributos através de medidas de correlação (Mitra et al., 2002; Au et al., 2005; Covões et al., 2009). Na prática, pode não ser claro como determinar redundância entre atributos quando um atributo é correlacionado (possivelmente parcialmente) com um conjunto de atributos. Métodos que utilizam grupos de atributos correlacionados visam tentar capturar esse cenário mais realista. O conceito de agrupamento de atributos pode ser idealmente formalizado como:

Definição 5. *Agrupamento de atributos.* O agrupamento de atributos envolve a separação de um conjunto A de atributos $A = \{A_1, \dots, A_M\}$ em uma coleção $G^A = \{G_1, \dots, G_k\}$ de subconjuntos disjuntos de atributos correlacionados A_i de A , onde k é o número de grupos de atributos, de forma que $G_1 \cup \dots \cup G_k = A$, $G_i \neq \emptyset$ e $G_i \cap G_j = \emptyset$ para $i \neq j$.

Baseando-se nas definições acima, pode-se analisar algumas propriedades teóricas derivadas do agrupamento de atributos correlacionados.

Proposição 1. *Atributos fortemente relevantes só podem ser encontrados em grupos singletons, i.e., grupos formados por um único elemento (atributo).*

Justificativa. (Por contradição). Considerando inicialmente que existe um grupo G_m formado por dois atributos fortemente relevantes A_i e A_j , i.e. $G_m = \{A_i, A_j\}$. De acordo com a Definição 5, se A_i e A_j estão no mesmo grupo, então eles são correlacionados. Portanto, A_i incorpora a informação fornecida por A_j e vice-versa, i.e., eles são redundantes de acordo com a Definição 4. Isto contradiz a Definição 1, que determina que tanto A_i quanto A_j não podem ser removidos sem modificar a distribuição original das classes $P(C|A)$. De forma similar, assumindo que existe um grupo G_m formado por r ($r > 2$) atributos fortemente relevantes, pela Definição 5 tais atributos são correlacionados e, de acordo com a Definição 4 estes são redundantes. Consequentemente, $(r-1)$ desses atributos podem ser removidos sem gerar mudanças em $P(C|A)$. Isto implica que os r atributos não são fortemente relevantes, contradizendo a premissa. ■

Proposição 2. *Atributos irrelevantes e atributos fracamente relevantes não podem ser encontrados no mesmo grupo.*

Justificativa. (Por contradição). Assume-se que exista um grupo G_m formado por atributos correlacionados. Por hipótese, considera-se que $A_i \in G_m$ é irrelevante (Definição 3) e que os demais atributos em G_m são fracamente relevantes (Definição 2). Desses atributos fracamente relevantes, levando-se em consideração qualquer atributo A_j , i.e., $A_j \in \{G_m - A_i\}$. De acordo com a Definição 3, A_i é irrelevante se, e somente se, $\forall S'_i \subseteq S_i$ tem-se que $P(C|S'_i = s'_{ji}) = P(C|A_i = x_{ji}, S'_i = s'_{ji})$. No entanto, dado que A_i e A_j são correlacionados, A_i incorpora a informação fornecida por A_j , e vice-versa. Consequentemente, a mesma condição dada pela Definição 3 é válida para A_j , sendo portanto uma contradição com a definição de fraca relevância (Definição 2). ■

A partir das proposições acima se pode obter o seguinte corolário:

Corolário 1. *Se existem atributos irrelevantes em A , ao menos um deles será incluído em R .*

Justificativa. Através das proposições anteriores é fácil verificar que um dado grupo obtido pelo processo de agrupamento de atributos (completamente) correlacionados não pode ser composto por diferentes tipos de atributos (i.e., fortemente relevantes, fracamente relevantes e irrelevantes). Consequentemente, se existem atributos irrelevantes em A , pelo menos um deles será

incluído no subconjunto de atributos selecionados (*e.g.*, quando algum(ns) atributo(s) representativo(s) de cada grupo é/são selecionados). ■

O Corolário 1 sugere uma possível melhoria para métodos que realizam SA via agrupamento de atributos correlacionados, especificamente: o uso de *wrappers* para definir o subconjunto final de atributos selecionados. Esta abordagem é interessante em aplicações nas quais as bases de dados contêm um número significativo de atributos redundantes. Após a remoção desses atributos, apenas um pequeno conjunto de atributos será utilizado para processamento posterior, justificando assim o uso de um *wrapper* sem (possivelmente) comprometer a eficiência computacional. É importante, entretanto, fazer uma ressalva. Embora, na teoria, métodos de SA que visam obter grupos de atributos correlacionados podem realmente selecionar pelo menos um dos atributos irrelevantes em problemas de classificação, na prática atributos completamente irrelevantes (Definição 3) dificilmente serão encontrados, e a propriedade descrita no Corolário 1 pode não ser estritamente válida. Na realidade, (ir)relevância e redundância, e consequentemente correlação, são uma questão de grau. Desse ponto de vista, métodos heurísticos baseados no agrupamento de atributos frequentemente fornecem bons resultados por meio da *aproximação* do subconjunto de atributos R com a remoção de atributos (parcialmente) redundantes de A .

No caso de problemas de agrupamento as definições e proposições aqui mencionadas em geral não podem ser aplicadas diretamente, pois os rótulos dos grupos, que poderiam ser considerados, em alguns casos, informações análogas àquelas fornecidas pelas classes, são desconhecidos. Em particular, sabe-se que diferentes grupos podem ser encontrados em diferentes subespaços e este fato dificulta sobremaneira a aplicação imediata dos conceitos aqui apresentados em problemas de agrupamento de dados. No entanto, neste trabalho é investigado o uso de métodos de seleção de atributos baseado no agrupamento de atributos nestes problemas sob a premissa de que todos os grupos são formados no mesmo subespaço e que a eliminação de redundância realizada por tais métodos pode ser benéfica para o processo de agrupamento de dados.

3.3 Trabalhos Relacionados

Nesta seção três algoritmos descritos na literatura que utilizam o conceito de agrupamento de atributos são apresentados. As modificações propostas no SSF objetivam suprir (parcialmente) as limitações encontradas nestes algoritmos.

3.3.1 Filtro proposto por Mitra, Murthy e Pal (MMP)

Mitra et al. (2002) propuseram um método de SA, aqui denominado de MMP, que utiliza uma medida de correlação linear entre os atributos como critério de avaliação para uma

eliminação *backward*. Como medida de correlação, os autores utilizam o Índice de Compressão Máxima de Informação (ICMI) (λ_2) entre dois atributos A_i e A_j :

$$2\lambda_2(A_i, A_j) = (var(A_i) + var(A_j)) - \sqrt{(var(A_i) + var(A_j))^2 - 4var(A_i)var(A_j)(1 - \rho(A_i, A_j)^2)}. \quad (3.1)$$

Onde $var(A_i)$ é a variância do atributo A_i , e $\rho(A_i, A_j)$ é o coeficiente de correlação de Pearson entre os atributos A_i e A_j , dado por:

$$\rho(A_i, A_j) = \frac{covariancia(A_i, A_j)}{\sqrt{var(A_i)var(A_j)}}. \quad (3.2)$$

A medida ICMI apresenta o valor zero quando os atributos são linearmente dependentes (*i.e.*, altamente correlacionados) e aumenta à medida que a dependência diminui, tendo como seu valor máximo $0,5(var(A_i) + var(A_j))$ (Mitra et al., 2002).

Seguindo a notação aqui adotada, M apresenta o número de atributos originais, A representa o conjunto de atributos originais e R o conjunto de atributos selecionados. Seja então r_i^k a dissimilaridade entre o atributo A_i e o seu k -ésimo vizinho mais próximo em R . O método de SA proposto por Mitra et al. (2002) pode ser descrito pelo Algoritmo 1.

Algoritmo 1: Algoritmo de Mitra, Murthy e Pal (MMP) (Mitra et al., 2002)

Entrada: $k \leq (M - 1)$

- 1 $R \leftarrow A$;
 - 2 **para cada** $A_i \in R$ **faça** calcular r_i^k ;
 - 3 Encontrar o atributo $A_{i'}$ para o qual $r_{i'}^k$ é mínimo. Guardar este atributo em R e descartar os k atributos mais próximos do atributo $A_{i'}$. (Obs.: $A_{i'}$ é o atributo para o qual a remoção dos k vizinhos mais próximos irá causar a menor perda de informação entre todos os atributos de R);
 - 4 $\epsilon \leftarrow r_{i'}^k$;
 - 5 **se** $k > |R| - 1$ **então** $k \leftarrow |R| - 1$;
 - 6 **se** $k = 0$ **então** Vá para a linha 13;
 - 7 **enquanto** $r_{i'}^k > \epsilon$ **faça**
 - 8 a) $k \leftarrow k - 1$;
 - 9 $r_{i'}^k \leftarrow$ menor r_i^k para um atributo de R ;
 - 10 b) **se** $k = 0$ **então** Vá para a linha 13;
 - 11 **fim**
 - 12 Vá para a linha 3;
 - 13 Retornar R (conjunto reduzido de atributos);
-

Os grupos de atributos são obtidos pelo princípio dos k -Vizinhos mais próximos. Em cada iteração, o atributo que fornece o menor valor de r_i^k é o atributo que fornece o grupo com $k + 1$ atributos mais compacto. Por tal razão, todos os seus k vizinhos são descartados.

Conforme descrito, o método é dependente da definição de um parâmetro k , o qual influen-

cia diretamente o número de atributos selecionados, já que $|R| \approx M - k$ (Mitra et al., 2002). Embora os autores do MMP indiquem que seja útil o controle sobre o nível de representação dos dados, permitindo uma certa análise exploratória, na prática é difícil a definição desse parâmetro pelo usuário, conforme será ilustrado nos experimentos reportados nesta dissertação.

O filtro MMP tem custo computacional estimado de $O(M^2 \cdot N)$ quando o valor do parâmetro k é conhecido (Mitra et al., 2002). No entanto, no caso em que o valor do parâmetro não é conhecido e uma análise exploratória é realizada, o custo computacional é estimado como:

$$O \left(M^2 \cdot N + \sum_{k=k_{min}}^{k_{max}} [M + (k - 1) \cdot (M - k) + \text{CustoClassificador}] \right).$$

Assumindo k_{min} e k_{max} como 1 e $M - 1$, respectivamente, e desconsiderando o custo do classificador para avaliar cada subconjunto de atributos obtido, obtém-se:

$$\begin{aligned} & O \left(M^2 \cdot N + \sum_{k=1}^{M-1} [M + (k - 1) \cdot (M - k)] \right) \\ &= O \left(M^2 \cdot N + M^2 + \left(M + 2M + \dots + \frac{M}{2} M + \left(\frac{M}{2} - 1 \right) M + \dots + 2M + M \right) \right) \\ &= O \left(M^2 \cdot N + M^2 + M \cdot \left(1 + 2 + \dots + \frac{M}{2} \right) \right) \\ &= O \left(M^2 \cdot N + M^2 + M \cdot \left(\frac{1 + \frac{M}{2}}{2} \right) \cdot \frac{M}{2} \right) \\ &= O \left(M^2 \cdot N + M^3 \right). \end{aligned}$$

Devido à dificuldade para definir o valor do parâmetro k , a análise exploratória com esses parâmetros $k_{min} = 1$ e $k_{max} = M - 1$ é utilizada como padrão nos experimentos realizados nesta dissertação.

3.3.2 Filtro proposto por Au et al. (ACA)

Au et al. (2005) propuseram um algoritmo de SA que utiliza uma medida de correlação não-linear para agrupar atributos. Este algoritmo é denominado Algoritmo de Agrupamento de Atributos (ACA). O algoritmo foi proposto para bases de dados apenas com atributos nominais. Caso existam atributos contínuos, estes são discretizados antes de se calcular as correlações. Dessa forma, $\forall A_i \in A$, $dom(A_i) = \{v_{i1}, \dots, v_{im_i}\}$, de forma que $v_{i1}, \dots, v_{im_i}$ são os valores discretos para o atributo A_i .

A medida de correlação utilizada pelo ACA é a Medida de Redundância de Interdependência (MRI), definida como (Au et al., 2005):

$$R(A_i, A_j) = \frac{I(A_i, A_j)}{H(A_i, A_j)}. \quad (3.3)$$

Onde $I(A_i, A_j)$ e $H(A_i, A_j)$, são respectivamente a Informação Mútua (Equação (3.4)) e a Entropia Conjunta (Equação (3.5)) entre os atributos A_i e A_j ²:

²A base do logaritmo pode variar de acordo com a aplicação. Neste trabalho todos os logaritmos são levados a

$$I(A_i, A_j) = \sum_{k=1}^{m_i} \sum_{l=1}^{m_j} P(A_i = v_{ik}, A_j = v_{jl}) \log \frac{P(A_i = v_{ik}, A_j = v_{jl})}{P(A_i = v_{ik})P(A_j = v_{jl})} \quad (3.4)$$

$$H(A_i, A_j) = - \sum_{k=1}^{m_i} \sum_{l=1}^{m_j} P(A_i = v_{ik}, A_j = v_{jl}) \log P(A_i = v_{ik}, A_j = v_{jl}) \quad (3.5)$$

A Equação (3.3) reflete a dependência entre A_i e A_j . O denominador é uma normalização aplicada à Informação Mútua que tem um viés para atributos com muitos valores. A medida **MRI** apresenta valores no intervalo $[0,1]$, sendo que o valor 1 indica atributos completamente correlacionados e 0 atributos independentes (Au et al., 2005).

Para o agrupamento dos atributos é utilizada uma variante do algoritmo *k-means* (Wu et al., 2008; Hartigan e Wong, 1979) denominada de *k-modes*, cujas principais modificações em relação ao *k-means* são:

- Substituição do conceito de centróide pelo atributo moda (*mode*) do grupo. Sendo G_r um grupo de atributos, o atributo moda do grupo é definido como o atributo com maior redundância múltipla, *i.e.*:

$$\eta_r = \arg \max_{A_i} \sum_{A_j \in G_r} R(A_i, A_j) \quad (3.6)$$

- Substituição da medida de distância pela medida de redundância de interdependência.

O Algoritmo *k-modes* pode ser descrito pelo Algoritmo 2.

Algoritmo 2: Algoritmo *k-modes* (Au et al., 2005)

Entrada: $k \geq 2$

- 1 Selecionar k atributos distintos para serem os k *modes* iniciais;
 - 2 **repita**
 - 3 **para cada** $A_i \in A$ **faça**
 - 4 Rotular atributo A_i como pertencente ao grupo G_r se
 $R(A_i, \eta_r) \geq R(A_i, \eta_s) \forall s \in \{1, \dots, k\} - \{r\}$;
 - 5 **fim**
 - 6 Redefinir atributo moda dos grupos através da Equação (3.6);
 - 7 **até os atributos moda sejam os mesmos ou um número máximo de iterações tenha sido alcançado** ;
-

O algoritmo 2 necessita da definição de um parâmetro (k) que corresponde ao número de grupos. No entanto, os autores sugerem que pela utilização do somatório da medida de redundância de interdependência significativa múltipla, para cada valor de k candidato, é possível selecionar o valor de k que a maximize (Au et al., 2005), *i.e.*:

$$k = \arg \max_{k \in \{2, \dots, M\}} \sum_{r=1}^k \sum_{A_i \in \{G_r - \eta_r\}} R(A_i : \eta_r). \quad (3.7)$$

A deficiência do [ACA](#) se encontra na ausência de um critério de seleção de atributos. Após o agrupamento dos atributos, os autores sugerem selecionar os atributos com maior redundância múltipla de cada grupo, porém nenhum critério de definição de quantos atributos devem ser selecionados para cada grupo é abordado no trabalho.

Para avaliar o custo computacional do algoritmo [ACA](#) assume-se que os grupos de atributos serão separados proporcionalmente, isto é, cada grupo possui $\frac{M}{k}$ atributos. Dessa forma, pode-se estimar o custo computacional do [ACA](#), desconsiderando o custo do cálculo da correlação entre atributos (Equação (3.3)), como:

$$O \left(\frac{(k_{min} + k_{max})(k_{max} - k_{min})}{2} \cdot M + M^2 \cdot \ln(k_{max} - k_{min}) + (k_{max} - k_{min}) \cdot M \right).$$

É fácil verificar que caso o valor do parâmetro k seja conhecido, o custo estimado é de $O(M^2)$. Para ilustrar um cenário onde não se conhece nenhuma faixa de valores interessantes para o parâmetro k , será utilizado $k_{min} = 2$ e $k_{max} = M - 1$. Neste caso, pode-se estimar o custo computacional do [ACA](#) como:

$$\begin{aligned} O \left(\frac{(2+(M-1))(M-1-2)}{2} \cdot M + M^2 \cdot \ln((M-1)-2) + ((M-1)-2) \cdot M \right) \\ = O \left(\frac{(M^2-3M)}{2} \cdot M + M^2 \cdot \ln(M) + M^2 - 3M \right) \\ = O(M^3). \end{aligned}$$

Finalmente, para ilustrar um cenário no qual se tem uma faixa de interesse para os valores do parâmetro k , será utilizado $k_{min} = 2$ e $k_{max} = \sqrt{M}$. Sendo assim, o custo computacional é estimado como:

$$\begin{aligned} O \left(\frac{(2+\sqrt{M})(\sqrt{M}-2)}{2} \cdot M + M^2 \cdot \ln(M^{\frac{1}{2}} - 2) + (\sqrt{M} - 2) \cdot M \right) \\ = O \left(\frac{M}{2} \cdot M + M^2 \cdot \frac{1}{2} \ln(M) + M\sqrt{M} - 2M \right) \\ = O(M^2 \cdot \ln(M)). \end{aligned}$$

3.3.3 Filtro Silhueta Simplificada ([SSF](#))

O [SSF](#), introduzido em ([Covões et al., 2009](#)), utiliza uma estratégia de amostragem descrita em [Hruschka et al. \(2006b\)](#) para gerar grupos de atributos correlacionados. A amostragem envolve a indução de partições diferentes para variadas quantidades de grupos. Partições de dados são induzidas por meio do algoritmo *k-medoids*.

O algoritmo *k-medoids* é equivalente ao algoritmo *k-modes*, com a ressalva de que o primeiro foi formulado para minimizar as distâncias entre medóides³ e os objetos do grupo, enquanto o segundo procura maximizar a correlação entre o atributo moda e os objetos do grupo. Os termos medóide e atributo moda são equivalentes. Como consequência, quando necessário o medóide de um grupo G_r será denotado por η_r . Para utilizar medidas de correlação que são maximizadas na medida em que a correlação entre atributos aumenta, é necessário adaptá-la para ser utilizada como medida de distância. Por exemplo, para que a medida de correlação **MRI** (Equação (3.3)) seja utilizada em conjunto com o *k-medoids*, pode-se usar a Equação (3.8):

$$R_d(A_i, A_j) = 1 - R(A_i, A_j) \quad (3.8)$$

O algoritmo **SSF** foi originalmente concebido como uma alternativa ao **MMP**, a fim de reduzir seu custo computacional e, ao mesmo tempo, aliviar o usuário da definição de um parâmetro crítico (k).

Assim como o **ACA**, o **SSF** utiliza um critério de validação de partição para a escolha do melhor agrupamento. No entanto, a medida utilizada no **SSF** é a Silhueta Simplificada (**SS**) (**Hruschka et al., 2006a**) que, diferentemente da medida utilizada no **ACA**, leva em consideração não apenas a compactação dos grupos, mas também a separação entre eles. A **SS** é descrita em detalhes na Subseção 3.3.3.1.

O algoritmo **SSF** está descrito no Algoritmo 3. Para cada valor de k são geradas n_p partições diferentes para tentar evitar mínimos locais, que são um problema bem conhecido do *k-medoids* (**Zhang et al., 1997**). Após avaliar o agrupamento o **SSF** seleciona o medóide de cada grupo como seu atributo representativo.

É possível utilizar diferentes medidas de correlação em conjunto com o **SSF**, bem como selecionar mais de um atributo por grupo, conforme será discutido na Seção 3.4.

Devido à utilização de algoritmos de agrupamento equivalentes, os filtros **SSF** e **ACA** possuem estimativas de custo computacional similares. Mais especificamente, o custo computacional do **SSF** pode ser estimado como:

$$O\left(2 \left[\frac{(k_{min} + k_{max})(k_{max} - k_{min})}{2} \cdot M \right] + M^2 \cdot \ln(k_{max} - k_{min})\right)$$

Conforme se pode observar, valores diferenciados para k_{min} e k_{max} levam a custos computacionais diferenciados. As estimativas de custo computacional relacionadas às atribuições $k_{min} = 2$ e $k_{max} = M - 1$ e $k_{min} = 2$ e $k_{max} = \sqrt{M}$ são similares as apresentadas na Subseção 3.3.2, levando a custos na ordem de $O(M^3)$ e $O(M^2 \cdot \ln(M))$, respectivamente.

³Atributo com menor distância em relação aos demais atributos do seu grupo.

Algoritmo 3: Filtro Silhueta Simplificada (SSF) (Covões et al., 2009)

Entrada: $k_{min} \geq 2$;
 $k_{max} \leq M - 1$;
 $n_p \geq 1$;
1 $MVO \leftarrow -\infty$; // Melhor valor obtido //
2 **para cada** $k \in \{k_{min}, \dots, k_{max}\}$ **faça**
3 **para cada** $nroParticao \in \{1, \dots, n_p\}$ **faça**
4 Gerar partição inicial aleatória $P_{inicial}$;
5 Executar k -medoids utilizando uma medida de correlação (e.g., Equação (3.1) ou Equação (3.8)) e $P_{inicial}$ como partição inicial dos grupos;
6 Seja P_{obtida} a partição obtida no final do agrupamento;
7 $VSS \leftarrow SilhuetaSimplificada(P_{obtida})$;
8 **se** $VSS > MVO$ **então**
9 $MVO \leftarrow VSS$;
10 $k^* \leftarrow k$;
11 $P^* \leftarrow P_{obtida}$;
12 **fim**
13 **fim**
14 **fim**
15 $SELECIONADOS \leftarrow \emptyset$;
16 **para cada** $G_m \in P^*$ **faça**
17 $SELECIONADOS \leftarrow SELECIONADOS \cup \{\eta_m\}$;
18 **fim**
19 Retornar $SELECIONADOS$;

3.3.3.1 Validação de Partição via Silhueta Simplificada

Antes de apresentar a Silhueta Simplificada (SS), será apresentada a silhueta originalmente proposta por Kaufman e Rousseeuw (1990). Considerando o atributo A_i pertencendo ao grupo G_j , a distância média do atributo A_i em relação a todos os outros atributos pertencentes ao grupo G_j será chamada de $a(i)$. Levando em consideração o grupo G_r , a distância média do atributo A_i em relação a todos os outros atributos pertencentes ao grupo G_r pode ser chamada de $d(i, G_r)$. Após calcular $d(i, G_r)$ para todos os grupos $G_r \neq G_j$, o menor valor de $d(i, G_r)$ (aqui chamado de $b(i)$) é escolhido. Mais precisamente, $b(i) = \min d(i, G_r), G_r \neq G_j$. Este valor representa a distância entre o atributo A_i e o seu grupo vizinho. Feitas estas considerações, a silhueta $s(i)$ é dada por:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \quad (3.9)$$

É fácil verificar que $-1 \leq s(i) \leq 1$. Logo, quanto maior o valor de $s(i)$, melhor terá sido a atribuição do atributo A_i para um dado grupo. Se $s(i) = 0$, então não está claro se o atributo A_i deveria ser agrupado com o grupo atual ou com o seu grupo vizinho mais próximo (Everitt, 2001). Finalmente, se o grupo G_j possui um único elemento, então $s(i)$ não é definida e uma escolha adequada é considerar $s(i) = 0$ (Kaufman e Rousseeuw, 1990). A média dos valores de $s(i)$ para todos os atributos, *i.e.* $\frac{1}{M} \sum_{i=1}^M s(i)$, é usada como o critério de avaliação de uma partição, sendo o melhor agrupamento determinado quando o seu valor é maximizado.

A silhueta proposta por Kaufman e Rousseeuw (1990) depende do cálculo de todas as distâncias entre os atributos, levando a um custo computacional de $O(M^2)$, o qual normalmente é muito alto para aplicações de mineração de dados. Para suprir essa limitação, a Silhueta Simplificada (SS) (Hruschka et al., 2006a) pode ser utilizada. A SS é baseada no cálculo das distâncias entre os atributos e os centróides dos grupos (na aplicação em questão, os medóides dos grupos são utilizados em vez dos centróides). Mais especificamente, o termo $a(i)$ se torna a distância entre o atributo A_i e o medóide do seu grupo (η_j). Similarmente, ao invés de calcular $d(i, G_r)$ como a distância média do atributo A_i em relação a todos os atributos de G_r , $G_r \neq G_j$, são calculadas as distâncias entre o atributo A_i e o medóide do grupo G_r (η_r). Essas modificações reduzem o custo computacional de $O(M^2)$ para $O(M)$.

3.4 Modificações para o Filtro Silhueta Simplificada (SSF)

Nesta seção são apresentadas as modificações propostas para o algoritmo SSF. As modificações envolvem a medida utilizada para agrupar os atributos e o critério de escolha dos atributos após a execução do algoritmo de agrupamento de atributos.

3.4.1 Medidas de correlação

Conforme mencionado na Seção 3.2, métodos de seleção de atributos que utilizam o conceito de agrupamento de atributos utilizam como base a Definição 4. Por tal razão, a escolha de uma medida de correlação entre atributos é um item crucial no desenvolvimento de um algoritmo dessa natureza.

Neste trabalho, propõe-se a utilização da medida *Symmetrical Uncertainty* (SU) (Press et al., 1990) já utilizada em outros algoritmos de SA, e.g., Hall (1999), Yu e Liu (2003). Essa medida é baseada na Entropia (Shannon, 1948), a qual mensura a incerteza em relação a uma variável aleatória. Assumindo que os dados são discretos e que $\forall A_i \in A, \text{dom}(A_i) = \{v_{i1}, \dots, v_{im_i}\}$, a Entropia de um atributo A_i pode ser definida como:

$$H(A_i) = - \sum_{k=1}^{m_i} P(A_i = v_{ik}) \log_2 P(A_i = v_{ik}). \quad (3.10)$$

A Entropia condicional do atributo A_i , obtida após observar os valores do atributo A_j , é por sua vez definida como:

$$H(A_i|A_j) = - \sum_{l=1}^{m_j} P(A_j = v_{jl}) \sum_{k=1}^{m_i} P(A_i = v_{ik}|A_j = v_{jl}) \log_2 P(A_i = v_{ik}|A_j = v_{jl}). \quad (3.11)$$

Para mensurar a redução da Entropia de A_i após a observação de A_j é utilizado o Ganho de Informação (Quinlan, 1993), definido como:

$$IG(A_i|A_j) = H(A_i) - H(A_i|A_j). \quad (3.12)$$

Finalmente, a SU é definida como (Press et al., 1990):

$$SU(A_i, A_j) = 2 \left[\frac{IG(A_i|A_j)}{H(A_i) + H(A_j)} \right]. \quad (3.13)$$

Embora o Ganho de Informação possa ser usado como uma medida de correlação, ele sofre da deficiência de favorecer atributos com maior quantidade de valores. Para superar tal limitação é comum o uso da SU, que compensa essa deficiência, adicionalmente normalizando os valores ao intervalo [0,1]. A SU é maximizada quando os atributos são completamente correlacionados (Yu e Liu, 2003).

Apesar da medida SU ser similar a MRI, o seu uso nas variantes propostas ao SSF neste trabalho se deve ao seu amplo uso pela comunidade que estuda métodos para SA. Para a utilização da SU em conjunto com o algoritmo *k-medoids*, propõe-se que a distância entre dois atributos A_i e A_j seja calculada por:

$$SU_d(A_i, A_j) = 1 - SU(A_i, A_j) \quad (3.14)$$

Neste trabalho também são investigados o uso, como medida de distância no algoritmo **SSF**, de duas medidas de correlação adaptadas, a saber: medida de Redundância de Interdependência (Equação (3.8)) e do coeficiente de correlação de Pearson (Equação (3.15)). O uso da primeira medida permite comparações diretas com o algoritmo **ACA**, enquanto que a segunda medida será investigada devido ao seu amplo uso na literatura.

$$\rho_d(A_i, A_j) = 1 - |p(A_i, A_j)|. \quad (3.15)$$

3.4.1.1 Uso de **SU** em Problemas de Classificação

No caso de problemas de classificação, além da medida de distância definida na Equação (3.14) também é analisada a utilização da medida definida na Equação (3.16), que em tese permite que os atributos sejam agrupados levando em conta sua correlação com a classe.

$$SU_s(A_i, A_j) = \frac{SU_d(A_i, A_j) + |SU(A_i, C) - SU(A_j, C)|}{2}. \quad (3.16)$$

Na Equação (3.16) o denominador é uma normalização para que os valores estejam no intervalo $[0,1]$. O uso dessa medida de correlação será denotado por *Symmetrical Uncertainty* considerando supervisão (**SU_s**).

Outra modificação proposta para a utilização do **SSF** em problemas de classificação está na seleção do medóide de cada grupo. Para este tipo de problema deseja-se que este atributo seja o mais correlacionado com os demais do seu grupo e o mais correlacionado com a classe. Sendo assim, o medóide do grupo em problemas de classificação pode ser definido da seguinte forma:

$$\eta_r = \operatorname{argmax}_{A_i} \left[\frac{\sum_{A_i, A_j \in G_r} SU(A_i, A_j)}{|G_r| - 1} + SU(A_i, C) \right] \cdot \frac{1}{2}. \quad (3.17)$$

Na Equação (3.17) o valor é normalizado pelo denominador para que esteja entre $[0,1]$. O uso dessa definição de medóide é denotado por *k-medoids* considerando supervisão (**KS**).

3.4.2 Critério de seleção de atributos

Conforme já mencionado na Seção 3.3, uma deficiência comum dos algoritmos de **SA** correlatos com esse trabalho é a necessidade da definição de um parâmetro que define o número de atributos a serem selecionados. Embora o **SSF** já tenha um critério de seleção dos atributos após o agrupamento (a seleção do medóide de cada grupo), é proposta uma modificação nesse critério para a seleção de dois atributos de cada grupo. O motivo dessa modificação é que resultados em **Covões et al. (2009)** indicam que a **SS** tem certo viés por um número reduzido de grupos, o que resulta em um número pequeno de atributos selecionados.

Supondo um grupo de atributos G_r , formado por m atributos, seja η_r o medóide do grupo, *i.e.*, o atributo com menor distância (maior correlação) com os outros atributos do grupo, e A_i o atributo mais distante (menos correlacionado) com η_r . O atributo A_i será chamado de

atributo de “*fronteira*”. O atributo de fronteira, considerando o uso da *SU* para o cálculo da distância (Equação (3.14)), pode ser então definido pela Equação (3.18). Um exemplo de uma configuração de grupos, em que ambos os atributos são destacados, é ilustrada na Figura 3.1.

$$Fronteira_r = \operatorname{argmax}_{A_i} SU_d(A_i, \eta_r). \quad (3.18)$$

Figura 3.1: Grupos de atributos: medóides representados por losangos; atributos de fronteira representados por triângulos.

A escolha desses dois atributos (medóide e fronteira) baseia-se nas seguintes premissas: o atributo medóide é por definição, o atributo mais correlacionado com os outros atributos, e em tese, fornece *aproximadamente* a mesma informação disponível nos demais atributos do mesmo grupo; o atributo de fronteira é o atributo que teria menos informação representada pelo atributo medóide e, por tal razão, sua seleção pode trazer um certo ganho de informação. Do ponto de vista do agrupamento de atributos, este critério de seleção se torna mais interessante quando considerado o caso de grupos com sobreposição ou com pouca coesão.

Para problemas de classificação, deseja-se não apenas que o atributo de fronteira seja o menos correlacionado com o medóide, mas que ele também seja o mais correlacionado com a classe. Para tal, são investigadas duas abordagens para se determinar o atributo de fronteira de um grupo levando em consideração a correlação em relação à classe. Na primeira abordagem, é considerada a média entre a correlação com o atributo medóide e a correlação com a classe, isto é, a fronteira do grupo G_r é definida como:

$$Fronteira_r = \operatorname{argmax}_{A_i} \frac{SU_d(A_i, \eta_r) + SU(A_i, C)}{2}. \quad (3.19)$$

Na segunda abordagem, é considerada a Informação Mútua Condicionada Normalizada (IMCN) (Jakulin, 2005), definida como:

$$\bar{I}(X; Y|Z) = \frac{H(X|Z) - H(X|Y, Z)}{H(X|Y, Z)}, \quad (3.20)$$

esta medida mensura a redução da incerteza da variável X dado o conhecimento de Y quando

Z é dado (Cover e Thomas, 1991). Neste trabalho, busca-se encontrar o atributo (A_i) que causa maior redução na incerteza do rótulo da classe (C) quando o medóide do grupo é dado (η_r). Portanto, nessa abordagem, a fronteira do grupo G_r é definida como:

$$Fronteira_r = \operatorname{argmax}_{A_i \in G_r} \bar{I}(C; A_i | \eta_r). \quad (3.21)$$

3.5 Considerações Finais

Neste capítulo foi descrito um arcabouço teórico para algoritmos de SA que utilizam conceitos de agrupamento de atributos. Mais especificamente, foram descritas definições para a SA em problemas de classificação. Também foram introduzidos três trabalhos que serão o foco desta dissertação. Em seguida, foi apresentado o algoritmo SSF e também foram introduzidas as modificações propostas ao algoritmo. Especificamente, neste trabalho são investigadas as seguintes variantes do SSF:

- Utilização das seguintes medidas de correlação como medidas de distância:
 - Índice de Compressão Máxima de Informação (λ_2) - Equação (3.1);
 - Coeficiente de correlação de Pearson (ρ) - Equação (3.2);
 - Medida de Redundância de Interdependência (MRI) - Equação (3.3);
 - *Symmetrical Uncertainty* (SU) - Equação (3.14);
 - *Symmetrical Uncertainty* considerando supervisão (SU_S) - Equação (3.16);
- Utilização dos seguintes critérios de seleção de atributos:
 - Atributo medóide;
 - Atributo medóide e atributo de fronteira;
 - * Atributo de fronteira determinado pela correlação com o medóide — Equação (3.18);
 - * Atributo de fronteira determinado pela média entre a correlação com o medóide e com a classe — Equação (3.19);
 - * Atributo de fronteira determinado pela medida Informação Mútua Condicionada Normalizada (IMCN) — Equação (3.21).

Resultados Experimentais em Problemas de Classificação

4.1 Considerações Iniciais

Neste capítulo são descritos os experimentos realizados envolvendo doze bases de dados de classificação, bem como dois estudos de caso de aplicações reais nas quais as variantes do algoritmo [SSF](#) proposto neste trabalho foram aplicadas. O primeiro estudo de caso envolve a classificação de grãos de café, de acordo com seus defeitos, e foi realizado em conjunto com a empresa NatComp¹. O segundo estudo de caso envolve a classificação de substantivos em uma aplicação de mineração de textos e no contexto do projeto “*Read The Web*”². A Seção [4.2](#) descreve a metodologia utilizada para avaliar os algoritmos de seleção de atributos. Na Seção [4.3](#) os algoritmos de seleção de atributos avaliados são especificados, assim como os parâmetros que foram utilizados. A Seção [4.4](#) descreve as bases de dados utilizadas nos experimentos. Finalmente, na Seção [4.5](#) são apresentados e discutidos os resultados obtidos.

4.2 Metodologia de Avaliação

A avaliação dos algoritmos de Seleção de Atributos ([SA](#)) foi realizada considerando os seguintes critérios:

1. Erro obtido por um classificador induzido a partir do subconjunto de atributos selecionados.

¹<http://www.natcomp.com.br>

²<http://rtw.ml.cmu.edu>

2. Número de atributos selecionados;
3. Custo computacional envolvido na SA.

Para estimar a taxa de erro obtida pelos classificadores foi utilizado o procedimento de validação cruzada estratificada de 10 pastas (*stratified 10-fold cross validation*) (ver Seção 1.3), em que a SA é aplicada apenas nos dados do conjunto de treinamento (Reunanen et al., 2003; Singhi e Liu, 2006). Esse procedimento é utilizado para simular uma aplicação real, e assim obter resultados com menor viés em relação à amostra de dados disponível. Uma iteração da validação cruzada é descrita na Figura 4.1. Para avaliar os subconjuntos selecionados pelos algoritmos foram utilizados dois algoritmos de classificação amplamente utilizados na literatura e na prática de mineração de dados (Wu et al., 2008), a saber: *k-Nearest Neighbors* (KNN) (Aha et al., 1991) e *Naïve Bayes* (NB) (Witten e Frank, 2005). Foi utilizada a implementação dos dois classificadores disponíveis no sistema *Waikato Environment for Knowledge Knowledge Analysis* (WEKA) (Witten e Frank, 2005) com seus parâmetros *default*.

Figura 4.1: Uma iteração de validação cruzada com a seleção de atributos integrada.

Para facilitar a comparação entre os métodos de SA foi utilizado o modelo de comparação proposto por Lee (2005), o qual permite realizar uma análise sumarizada dos resultados obtidos. Nesta abordagem multicritério, o desempenho de cada método é avaliado utilizando-se um gráfico bidimensional, no qual são representados: o erro obtido por um classificador induzido pelo subconjunto de atributos selecionados e a porcentagem de atributos selecionados. Neste gráfico, o Eixo “*x*” representa à média do erro obtido e o Eixo “*y*” a porcentagem de atributos selecionados em relação ao total de atributos (Lee, 2005). O modelo do gráfico está ilustrado na Figura 4.2. Nesta, ECM é definido como o menor valor entre 50% e o erro obtido classificando todos os objetos como pertencentes à classe majoritária, e ESA é definido como o erro obtido pelo classificador utilizando todos os atributos. Por fim, os métodos têm seu desempenho categorizado (através dos símbolos entre parênteses) em uma das 5 regiões demarcadas no gráfico:

- Ruim (▼): indica que a redução no erro em relação ao ECM não foi satisfatória em relação ao número de atributos selecionados;

Figura 4.2: Avaliação de seleção de atributos multicritério proposta em (Lee, 2005).

- Regular (◇): indica que foram selecionados menos que 50% dos atributos, porém a redução no erro em relação ao ECM não foi satisfatória;
- Bom (▲): indica que foram selecionados mais que 50% dos atributos, porém o aumento no erro em relação ao ESA foi aceitável;
- Muito Bom (▲▲): indica que menos de 50% dos atributos foram selecionados e o aumento no erro em relação ao ESA foi aceitável;
- Excelente (▲▲▲): indica que o método conseguiu reduzir o erro em relação ao ESA, independentemente do número de atributos selecionados.

Os critérios adotados para categorizar o desempenho de cada método podem ser considerados subjetivos. No entanto, tais critérios são razoáveis e muito úteis, para fim de sumarização de resultados a serem apresentados, neste contexto, serão adotados nesta dissertação.

Para avaliar com certa evidência estatística o efeito da seleção de atributos em um classificador é usual o uso de testes de hipóteses (Liu e Motoda, 1998). Neste sentido foi utilizado neste trabalho o procedimento descrito por Demšar (2006), que consiste na aplicação dos testes não-paramétricos³ de Friedman e de Nemenyi (Hollander e Wolfe, 1991). Para serem utilizados, ambos os testes estatísticos foram implementados em Java.

O teste de Friedman verifica a hipótese de que não há diferença entre os efeitos causados pelos métodos de seleção de atributos. Quando o teste rejeita esta hipótese, é assumida a hipótese alternativa de que ao menos dois dos métodos de seleção de atributos avaliados tem efeitos diferentes (Hollander e Wolfe, 1991). Para a aplicação do teste é necessário que os métodos sejam

³Testes que não assumem que os dados sejam provenientes de determinada distribuição.

ordenados de acordo com seu efeito, *i.e.*, o método com melhor efeito sobre o critério sendo avaliado, *e.g.*, erro obtido em um classificador usando os atributos selecionados, é o número 1 do *ranking*, o segundo melhor o número 2 e assim por diante. No caso de empates, a média dos *ranks* é utilizada (Demšar, 2006). Sendo r_j^i o *ranking* do j -ésimo método de seleção de atributos na i -ésima base de dados, *i.e.*, $i \in \{1, 2, \dots, D\}$ e $j \in \{1, 2, \dots, K\}$, a média dos *rankings* de um algoritmo j é dada por $R_j = \frac{1}{D} \sum_i^D r_j^i$. O teste de Friedman se baseia na Equação (4.1)⁴, tendo seu valor distribuído de acordo com a distribuição F com $K - 1$ e $(K - 1)(D - 1)$ graus de liberdade.

$$F_F = \frac{(D - 1) \left[\frac{12D}{K(K+1)} \left(\sum_{j=1}^K R_j^2 - \frac{K(K+1)^2}{4} \right) \right]}{D(K - 1) - \left[\frac{12D}{K(K+1)} \left(\sum_{j=1}^K R_j^2 - \frac{K(K+1)^2}{4} \right) \right]} \quad (4.1)$$

Caso a hipótese nula (não há diferença entre os efeitos causados pelos métodos de seleção de atributos) seja rejeitada, torna-se necessário o uso de um teste *post-hoc* para fazer as comparações par-a-par a fim de detectar entre quais algoritmos existem diferenças (Demšar, 2006). Neste trabalho é utilizado o teste de Nemenyi para as comparações par-a-par. A diferença do efeito de dois métodos de seleção de atributos é dita estatisticamente significativa se a diferença entre seus *rankings*, *i.e.*, $|R_i - R_j|$, é maior ou igual à diferença crítica dada pela Equação (4.2), onde os valores q_α são baseados na estatística de intervalo estudentizado dividido por $\sqrt{2}$, de acordo com a Tabela 4.1 (Demšar, 2006).

$$CD = q_\alpha \sqrt{\frac{K(K+1)}{6D}} \quad (4.2)$$

Nº de Métodos	2	3	4	5	6	7	8	9	10
$q_{0,05}$	1,960	2,343	2,569	2,728	2,850	2,949	3,031	3,102	3,164
$q_{0,10}$	1,645	2,052	2,291	2,459	2,589	2,693	2,780	2,855	2,920

Tabela 4.1: Valores críticos (q_α) para o teste de Nemenyi (Demšar, 2006).

Para as análises dos tempos computacionais de cada algoritmo serem coerentes, todos os experimentos foram executados no mesmo computador (Opteron 2GHz com 8Gb de RAM), executando apenas o Sistema Operacional em paralelo. Além disso, todos os algoritmos foram implementados em Java. Dessa forma, é possível verificar a diferença em termos das constantes de tempo de cada algoritmo, omitidos nas análises assintóticas do custo computacional.

4.3 Algoritmos e Parâmetros Utilizados

Nestes experimentos foram comparados os algoritmos **SSF** (Seção 3.3.3) e suas variantes (Seção 3.4), **ACA** (Seção 3.3.2) e **MMP** (Seção 3.3.1). Os parâmetros dos algoritmos foram definidos da seguinte forma:

⁴Esta é uma versão corrigida sugerida por Demšar (2006), por ser menos conservadora que a original.

- **SSF** selecionando 1 atributo por grupo:

$$\begin{aligned}
 & - k_{min} = 2; \\
 & - k_{max} = \begin{cases} M - 1, & \text{se } M < 1000 \\ \sqrt{M}, & \text{se } M \geq 1000 \end{cases}; \\
 & - n_p = 20.
 \end{aligned}$$

- **SSF** selecionando 2 atributos por grupo:

$$\begin{aligned}
 & - k_{min} = 2; \\
 & - k_{max} = \begin{cases} M/2, & \text{se } M < 1000 \\ \sqrt{M}, & \text{se } M \geq 1000 \end{cases}; \\
 & - n_p = 20.
 \end{aligned}$$

- **ACA**:

– Parâmetros idênticos ao **SSF**.

- **MMP**:

$$\begin{aligned}
 & - k_{min} = 1; \\
 & - k_{max} = \begin{cases} M - 1, & \text{se } M < 1000 \\ \sqrt{M}, & \text{se } M \geq 1000 \end{cases}.
 \end{aligned}$$

Acredita-se que esses parâmetros sejam suficientes para investigar de maneira controlada, o comportamento dos algoritmos nas bases de dados utilizadas. Em uma aplicação prática, conhecimento de domínio poderia ser incorporado para a determinação de tais parâmetros. Para a discretização dos valores, quando necessária, foi utilizado o algoritmo *Proportional k-Interval Discretization* (**PKID**) (**Yang e Webb, 2001**).

Para facilitar a apresentação dos resultados, foram adotados acrônimos, de acordo com a informação disposta na Tabela 4.2. Tais acrônimos serão usados ao longo do resto desta dissertação.

4.4 Bases de Dados

Foram realizados experimentos em doze bases de dados para efetuar comparações entre os algoritmos abordados nessa dissertação. Cinco bases de dados correspondem a problemas de bioinformática utilizados por **Yeung et al. (2003)**. Os autores criaram cinco conjuntos sintéticos de dados, com distribuições de erros derivadas de bases de dados reais de bioinformática. Neste trabalho estas bases de dados são chamadas de Bio1, ..., Bio5 e são constituídas por 400 genes (objetos), descritos por 20 medidas (atributos) e 6 classes aproximadamente balanceadas.

Foram também utilizadas três bases de dados reais:

Acrônimo	Significado.
M^*	Número de atributos selecionados.
%-KNN	Taxa de erro obtida no classificador KNN.
%-NB	Taxa de erro obtida no classificador NB.
MMP-NB	Algoritmo MMP (Seção 3.3.1) avaliando os subconjuntos de atributos com o classificador NB.
MMP-KNN	Algoritmo MMP (Seção 3.3.1) avaliando os subconjuntos de atributos com o classificador KNN.
MMP (M)	Melhor resultado obtido pelo algoritmo MMP (Seção 3.3.1), i.e. dentre os valores no intervalo $[k_{min}, k_{max}]$ o que apresentou menor erro no classificador sendo considerado.
MMP ($\bar{m}$)	Resultado médio obtido pelo algoritmo MMP (Seção 3.3.1), i.e. média do erro no classificador sendo considerado para todos os valores de k avaliados.
MMP (P)	Pior resultado obtido pelo algoritmo MMP (Seção 3.3.1). Análogo ao MMP (M).
ACA-1	Algoritmo ACA (Seção 3.3.2) selecionando o atributo moda de cada grupo.
ACA-2	Algoritmo ACA (Seção 3.3.2) selecionando os dois atributos com maior redundância de cada grupo.
SSF- λ -1	Algoritmo SSF (Seção 3.3.3), utilizando como medida de correlação a Equação (3.1). É selecionado apenas o atributo medóide de cada grupo.
SSF- λ -2	Algoritmo SSF (Seção 3.3.3), utilizando como medida de correlação a Equação (3.1), e selecionado os atributos medóide e fronteira de cada grupo, sendo o atributo fronteira definido pela Equação (3.18).
SSF- ρ -1	Algoritmo SSF (Seção 3.3.3), utilizando como medida de correlação a Equação (3.2). É selecionado apenas o atributo medóide de cada grupo.
SSF- ρ -2	Algoritmo SSF (Seção 3.3.3), utilizando como medida de correlação a Equação (3.2), e selecionado os atributos medóide e fronteira de cada grupo, sendo o atributo fronteira definido pela Equação (3.18).
SSF-R-1	Algoritmo SSF (Seção 3.3.3), utilizando como medida de correlação a Equação (3.3). É selecionado apenas o atributo medóide de cada grupo.
SSF-R-2	Algoritmo SSF (Seção 3.3.3), utilizando como medida de correlação a Equação (3.3), e selecionado os atributos medóide e fronteira de cada grupo, sendo o atributo fronteira definido pela Equação (3.18).
SSF-SU-1	Algoritmo SSF (Seção 3.3.3), utilizando como medida de correlação a Equação (3.13). É selecionado apenas o atributo medóide de cada grupo.
SSF-SU-2	Algoritmo SSF (Seção 3.3.3), utilizando como medida de correlação a Equação (3.13), e selecionado os atributos medóide e fronteira de cada grupo, sendo o atributo fronteira definido pela Equação (3.18).
SSF-SU _S -1	Algoritmo SSF (Seção 3.3.3), utilizando como medida de correlação a Equação (3.16). É selecionado apenas o atributo medóide de cada grupo.
SSF-SU _S -2	Algoritmo SSF (Seção 3.3.3), utilizando como medida de correlação a Equação (3.16), e selecionado os atributos medóide e fronteira de cada grupo, sendo o atributo fronteira definido pela Equação (3.19).
SSF-SU _S -KS-1	Algoritmo SSF (Seção 3.3.3), utilizando como medida de correlação a Equação (3.16). É selecionado apenas o atributo medóide de cada grupo, sendo este definido de acordo com a Equação (3.17).
SSF-SU _S -KS-2	Algoritmo SSF (Seção 3.3.3), utilizando como medida de correlação a Equação (3.16), e selecionado os atributos medóide e fronteira de cada grupo, sendo o atributo medóide definido pela Equação 3.17 e o atributo fronteira definido pela Equação (3.19).
SSF-SU _S -I	Algoritmo SSF (Seção 3.3.3), utilizando como medida de correlação a Equação (3.16), e selecionado os atributos medóide e fronteira de cada grupo, sendo o atributo fronteira definido pela Equação (3.21).
SSF-SU _S -KS-I	Algoritmo SSF (Seção 3.3.3), utilizando como medida de correlação a Equação (3.16), e selecionado os atributos medóide e fronteira de cada grupo, sendo o atributo medóide definido pela Equação 3.17 e o atributo fronteira definido pela Equação (3.21).
Todos	Resultados obtidos considerando o uso de todos os atributos.

Tabela 4.2: Acrônimos utilizados nas tabelas de resultados.

- (i) *Yeast Galactose Data* (Yeung et al., 2003), composta por 205 genes (objetos), 20 medidas (atributos) e separados em 4 classes;
- (ii) *Colon Cancer* (Alon et al., 1999), composta por 2000 genes (atributos), 62 medidas (objetos) e 2 classes;
- (iii) *Leukemia* (Golub et al., 1999), composta por 7129 genes (atributos), 72 medidas (objetos) e 2 classes.

As quatro bases de dados restantes foram obtidas do *UCI Machine Learning Repository* (Asuncion e Newman, 2007):

- (i) *Ionosphere*, composta por 351 sinais de radares (objetos), 17 pulsos (representados por 2 atributos cada) e 2 classes;
- (ii) *Pima Diabetes*, composta por 768 pacientes (objetos), 8 atributos e 2 classes;

- (iii) *Wisconsin Breast Cancer*, composta por 683 pacientes (objetos), 9 atributos e 2 classes;
- (iv) *Spambase*, composta por 4.601 e-mails (objetos), 57 atributos e 2 classes.

As características das bases de dados estão sumarizadas na Tabela 4.3. Nesta tabela, “identificador” refere-se ao nome pelo qual a base será referenciada nas próximas seções, N é o número de objetos e M o número de atributos. Todas as bases utilizadas não possuem valores ausentes.

Nome	Identificador	N	M	# classes (distribuição - %)
Bio1, ..., Bio5	Bio1, ..., Bio5	400	20	6 ($\approx$ igualmente distribuídas)
Yeast	Yeast	205	20	4 (40,5 - 7,3 - 45,4 - 6,8)
Ionosphere	Iono	351	34	2 (35,9 - 64,1)
Pima Diabetes	Pima	768	8	2 (65,1 - 34,9)
Wisconsin	Wisc	683	9	2 (65 - 35)
Spambase	Spam	4.601	57	2 (39,4 - 60,6)
Colon Cancer	Colon	62	2.000	2 (64,5 - 35,5)
Leukemia	Leu	72	7.129	2 (65,3 - 34,7)

Tabela 4.3: Características das bases de dados utilizadas nos experimentos.

4.5 Resultados e Discussão

Os experimentos realizados foram divididos de forma a facilitar a comparação entre os algoritmos. Sob esta perspectiva, na próxima seção os algoritmos **MMP** (Mitra et al., 2002) e **SSF** são comparados. Nas comparações realizadas o **SSF** utiliza dois coeficientes de correlação lineares — Pearson (ρ) definido na Equação (3.2) e Índice de Compressão Máxima de Informação (**ICMI**) (λ) definido na Equação (3.1)). Na Seção 4.5.2 os algoritmos **SSF** e **ACA** são comparados. Nesta comparação o **SSF** utiliza dois coeficientes de correlação não-lineares — (Medida de Redundância de Interdependência (**MRI**) (R) definido na Equação (3.3) e *Symmetrical Uncertainty* (**SU**) definido na Equação (3.13)). A análise de dois coeficientes de correlação em cada comparação se dá para permitir comparar o **SSF** com os outros dois métodos (**MMP** e **ACA**) utilizando os coeficientes de correlação definidos nesses trabalhos (**ICMI** e **MRI**), além de permitir comparar o uso de coeficientes de correlação amplamente utilizados na prática (Pearson e **SU**). Na Seção 4.5.3 as diferentes variantes supervisionadas do **SSF** propostas neste trabalho são comparadas. Os valores correspondentes às médias e aos desvios padrão do número de atributos selecionados por cada algoritmo, assim como as taxas de erro em cada classificador obtidas durante o processo de validação cruzada e o tempo computacional (em segundos) utilizados para a **SA** são apresentados no Apêndice A. As seções seguintes sintetizam os principais resultados obtidos, analisando os resultados detalhados apresentados neste apêndice.

4.5.1 Comparações entre **SSF** e **MMP**

Nos experimentos envolvendo o algoritmo **MMP** (Mitra et al., 2002), para cada valor de k do intervalo $[k_{min}, k_{max}]$, todos os subconjuntos de atributos selecionado foram avaliados por meio

dos classificadores. No entanto, são utilizados nas comparações apenas o melhor caso, denotado por “MMP (M)”, e o caso médio, denotado por “MMP ($\bar{m}$)” obtidos considerando-se a taxa de erro obtida por cada classificador. É importante ressaltar que esses valores foram obtidos através de uma busca guiada por um classificador e, portanto, todo o custo computacional necessário para encontrar o valor deve ser considerado, *i.e.*, a aplicação do Algoritmo 1 (Seção 3.3.1) e a construção do classificador para cada valor de k . As Figuras 4.3 e 4.4 apresentam, para cada valor de k analisado, o erro obtido pelos classificadores induzidos utilizando o subconjunto de atributos selecionados. O impacto que cada subconjunto de atributos causa na indução do classificador ilustra a dificuldade encontrada pelo usuário em definir o parâmetro “ k ”.

Nas Tabelas 4.4, 4.5 e 4.6 são apresentados o número de Vitórias/Empates/Derrotas comparando-se os algoritmos par-a-par, considerando os critérios de número de atributos selecionados, taxa de erro obtida pelo *Naïve Bayes* (NB) e taxa de erro obtida pelo k -*Nearest Neighbors* (KNN), respectivamente. Por exemplo, na Tabela 4.4 o valor “9/1/2” na 4ª linha e 3ª coluna significa que o algoritmo SSF- ρ -1 ganhou (selecionou menos atributos) nove, empatou (selecionou a mesma quantidade de atributos) uma e perdeu (selecionou menos atributos) duas vezes para o algoritmo SSF- λ -2.

Analisando a Tabela 4.4 é possível constatar que o algoritmo SSF seleciona menos atributos que o algoritmo MMP na maioria (57%) dos casos. Conforme abordado anteriormente na Seção 3.4, o número reduzido de atributos selecionados é um viés já conhecido do SSF e é uma das motivações para um dos critérios de seleção de atributos propostos neste trabalho (seleção de dois atributos por grupo: medóide e atributo menos correlacionado com o medóide). Quando este critério de seleção é utilizado (SSF- λ -2, SSF- ρ -2), o número de vitórias e derrotas entre os algoritmos SSF e MMP se nivelam. Além disso, é possível constatar que o algoritmo SSF usando o coeficiente de correlação de Pearson (SSF- ρ -*) seleciona menos atributos do que o algoritmo SSF utilizando o Índice de Compressão Máxima de Informação (ICMI).

Nas Tabelas 4.5 e 4.6 é possível constatar que quando o valor do parâmetro k para o MMP é desconhecido (MMP ($\bar{m}$)) o SSF é uma boa alternativa, principalmente utilizando Pearson como medida de correlação. Especificamente, o algoritmo SSF- ρ -2 foi superior ao MMP ($\bar{m}$) em aproximadamente 80% dos casos (onze vezes considerando o NB e oito vezes considerando o KNN— totalizando 19 melhores resultados dentre os 24 analisados).

Algoritmo	SSF- λ -1	SSF- λ -2	SSF- ρ -1	SSF- ρ -2	MMP-KNN	MMP-NB
SSF- λ -1	—	12/0/0	3/1/8	9/0/3	8/0/4	8/0/4
SSF- λ -2	0/0/12	—	2/1/9	4/0/8	6/0/6	6/0/6
SSF- ρ -1	8/1/3	9/1/2	—	12/0/0	7/0/5	9/0/3
SSF- ρ -2	3/0/9	8/0/4	0/0/12	—	6/0/6	5/0/7
MMP-KNN	4/0/8	6/0/6	5/0/7	6/0/6	—	4/2/6
MMP-NB	4/0/8	6/0/6	3/0/9	7/0/5	6/2/4	—

Tabela 4.4: Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o número de atributos selecionados entre SSF- λ , SSF- ρ e MMP (Melhor caso).

As Tabelas 4.7 e 4.8 apresentam as categorias nas quais são classificados os resultados para cada algoritmo de acordo com a avaliação multicritério proposta por Lee (2005) e descrita na

(a) Bio1

(b) Bio2

(c) Bio3

(d) Bio4

(e) Bio5

(f) Yeast

Figura 4.3: Resultados (erro de classificação - validação cruzada) obtidos nas bases Bio1, ..., Bio5 e Yeast utilizando o algoritmo MMP para cada valor do parâmetro k .

(a) Iono

(b) Pima

(c) Wisc

(d) Spam

(e) Colon

(f) Leu

Figura 4.4: Resultados (erro de classificação - validação cruzada) obtidos nas bases de dados Iono, Pima, Wisc, Spam, Colon e Leu utilizando o algoritmo **MMP** para cada valor do parâmetro k .

Algoritmo	SSF- λ -1	SSF- λ -2	SSF- ρ -1	SSF- ρ -2	MMP (M)	MMP ($\bar{m}$)
SSF- λ -1	—	2/4/6	6/2/4	1/2/9	1/2/9	5/0/7
SSF- λ -2	6/4/2	—	7/2/3	3/2/7	1/2/9	7/0/5
SSF- ρ -1	4/2/6	3/2/7	—	2/2/8	1/3/8	9/0/3
SSF- ρ -2	9/2/1	7/2/3	8/2/2	—	1/3/8	11/0/1
MMP (M)	9/2/1	9/2/1	8/3/1	8/3/1	—	10/2/0
MMP ($\bar{m}$)	7/0/5	5/0/7	3/0/9	1/0/11	0/2/10	—

Tabela 4.5: Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o erro obtido pelo classificador **NB** utilizando os subconjuntos de atributos selecionados pelos algoritmos **SSF- λ** , **SSF- ρ** e **MMP** (Melhor caso e caso médio).

Algoritmo	SSF- λ -1	SSF- λ -2	SSF- ρ -1	SSF- ρ -2	MMP (M)	MMP ($\bar{m}$)
SSF- λ -1	—	1/4/7	2/3/7	2/3/7	0/3/9	4/0/8
SSF- λ -2	7/4/1	—	6/3/3	5/3/4	0/3/9	7/0/5
SSF- ρ -1	7/3/2	3/3/6	—	4/3/5	0/3/9	8/0/4
SSF- ρ -2	7/3/2	4/3/5	5/3/4	—	0/3/9	8/0/4
MMP (M)	9/3/0	9/3/0	9/3/0	9/3/0	—	12/0/0
MMP ($\bar{m}$)	8/0/4	5/0/7	4/0/8	4/0/8	0/0/12	—

Tabela 4.6: Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o erro obtido pelo classificador **KNN** utilizando os subconjuntos de atributos selecionados pelos algoritmos **SSF- λ** , **SSF- ρ** e **MMP** (Melhor caso e caso médio).

Seção 4.2. É possível notar que, apesar do algoritmo **MMP** ter apresentado em geral os melhores resultados, em diversas bases de dados o **SSF** foi capaz de produzir resultados competitivos. Vale lembrar que, para esta avaliação, foi considerado apenas o melhor resultado obtido para o **MMP**, *i.e.* o valor de k que apresentou o menor erro nos classificadores.

Bases	SSF- λ -1	SSF- λ -2	SSF- ρ -1	SSF- ρ -2	MMP (M)
Bio1	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲
Bio2	▲▲	▲▲▲	▲▲	▲▲	▲▲▲
Bio3	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲
Bio4	▲▲	▲	▲▲	▲▲	▲▲▲
Bio5	▲▲▲	▲▲▲	▲▲	▲▲▲	▲▲▲
Yeast	▲▲	▲▲	▲▲	▲▲	▲▲▲
Iono	▲▲	▲▲	▲▲▲	▲	▲▲▲
Pima	◇	▼	▼	▼	▲▲▲
Wisc	▲▲	▲	▲▲	▲▲	▲▲▲
Spam	▼	▼	◇	▲	▲▲▲
Colon	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲
Leu	▲▲	▲▲	◇	◇	▲▲▲

Tabela 4.7: Avaliação multicritério (Seção 4.2) considerando o erro obtido pelo classificador **NB** utilizando os atributos selecionados pelos algoritmos **SSF- λ** , **SSF- ρ** e **MMP** (Melhor caso).

Os testes estatísticos de Friedman e de Nemenyi (abordados na Seção 4.2) sugerem que considerando o erro obtido pelos classificadores, tanto para o **NB** quanto para o **KNN**, diferenças estatisticamente significantes ($\alpha=5\%$) só foram observadas entre o algoritmo **MMP** e os algoritmos **SSF- λ -1** e **SSF- ρ -1**. Portanto, quando o critério de seleção de dois atributos de cada grupo — seleção do atributo medóide e do atributo menos correlacionado com o medóide — foi utilizado, não houve diferenças significantes entre os algoritmos **SSF** e **MMP**. Em contrapartida, o algoritmo **SSF- λ -1** obteve resultados significativamente piores ($\alpha = 5\%$) em relação à utilização de todos os atributos de acordo com as taxas de erro obtidas no classificador **KNN**. Levando em consideração o tempo computacional (apresentado na Tabela A.13), os algoritmos

Bases	SSF- λ -1	SSF- λ -2	SSF- ρ -1	SSF- ρ -2	MMP
Bio1	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲
Bio2	▲▲	▲▲▲	▲▲	▲▲	▲▲▲
Bio3	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲
Bio4	▲▲	▲	▲▲	▲▲	▲▲▲
Bio5	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲
Yeast	▲▲	▲▲	▲▲	▲▲	▲▲▲
Iono	◇	▲▲	▲▲▲	▲▲▲	▲▲▲
Pima	▼	▼	▼	▼	▲▲▲
Wisc	▲▲	▲	▲▲	▲▲	▲▲▲
Spam	◇	◇	▲▲	▲	▲▲▲
Colon	◇	▼	▼	▼	▲▲▲
Leu	◇	◇	◇	▼	▲▲▲

Tabela 4.8: Avaliação multicritério considerando o erro obtido pelo classificador KNN utilizando os atributos selecionados pelos algoritmos SSF- λ , SSF- ρ e MMP (Melhor caso).

SSF- λ -2 e SSF- ρ -2 foram significativamente ($\alpha=5\%$) mais rápidos em relação aos algoritmos SSF- λ -1, SSF- ρ -1 e ao MMP quando o classificador KNN foi utilizado para avaliar os subconjuntos de atributos. Levando-se em conta todas essas constatações, o SSF selecionando dois atributos por grupo é uma excelente alternativa ao algoritmo MMP, principalmente se classificadores menos escaláveis que o NB forem utilizados, tornando a busca exaustiva pelo melhor valor de k proibitiva.

Os resultados apresentados são interessantes sob a perspectiva do uso do SSF em aplicações práticas, em especial naquelas em que o usuário desconhece o número de atributos a serem selecionados. Sob o ponto de vista acadêmico, pode ser também de interesse avaliar os dois filtros (SSF e MMP) forçando uma situação em que ambos selecionam subconjuntos de atributos cujas cardinalidades sejam (aproximadamente) iguais.

Neste sentido, realizou-se uma investigação adicional, na qual objetiva-se verificar se algum dos algoritmos seleciona um melhor subconjunto de atributos para determinada cardinalidade. Para tal, considera-se como parâmetro o número de atributos selecionados pelo SSF- ρ -2 sendo o M^* . Na sequência, seleciona-se dentre os subconjuntos de atributos selecionados pelo MMP aquele de cardinalidade mais próxima de M^* . Este procedimento é executado em cada pasta da validação cruzada, com o objetivo de se manter boas estimativas de capacidade de generalização dos classificadores usados. Nos casos das bases de dados com mais de 1000 atributos (Colon e Leu), para as quais não foi realizada uma busca exaustiva pelo melhor valor para o parâmetro k , não estão disponíveis todos os subconjuntos de atributos selecionados pelo MMP devido ao alto custo computacional. Nestes casos, o valor de k foi definido como $k = M - M^*$. Dessa forma, o MMP seleciona no máximo M^* atributos. As médias e os desvios padrão obtidos na validação cruzada estão dispostos na Tabela 4.9. Utilizando o teste estatístico Wilcoxon Signed-Rank⁵ (Hollander e Wolfe, 1991) não foi possível determinar diferença estatística entre as taxas de erro obtidas tanto para o classificador KNN quanto para o NB. No entanto, é possível verificar que o algoritmo SSF- ρ -2 obteve resultados iguais ou melhores em relação ao MMP em aproximadamente 70% das bases de dados, considerando as taxas de erros obtidas

⁵Por se tratar de um par de algoritmos este teste é recomendado ao invés do teste de Friedman (Demšar, 2006).

pelos classificadores empregados.

Tabela 4.9: Erros de classificação — média (desvio padrão) — obtidos pelo **SSF- ρ -2** e **MMP** selecionando (aproximadamente) o mesmo número de atributos.

Base	SSF-ρ-2		MMP	
	%-KNN	%-NB	%-KNN	%-NB
Bio1	0,00 (0,00)	0,00 (0,00)	0,00 (0,00)	0,00 (0,00)
Bio2	4,00 (2,29)	2,75 (2,08)	4,00 (2,42)	3,00 (3,07)
Bio3	0,00 (0,00)	0,00 (0,00)	0,00 (0,00)	0,00 (0,00)
Bio4	2,75 (2,08)	3,00 (1,87)	5,50 (3,07)	4,50 (1,58)
Bio5	0,00 (0,00)	0,00 (0,00)	0,00 (0,00)	0,00 (0,00)
Yeast	5,93 (4,38)	5,43 (4,16)	9,36 (5,07)	7,36 (4,28)
Iono	11,66 (4,93)	17,63 (5,82)	12,81 (6,32)	23,93 (6,88)
Pima	36,46 (4,49)	30,74 (6,05)	30,48 (6,92)	23,70 (6,29)
Wisc	7,03 (2,27)	5,57 (2,62)	6,01 (3,06)	4,69 (2,17)
Spam	11,02 (2,54)	25,97 (4,54)	12,30 (1,20)	24,36 (4,15)
Colon	36,9 (17,63)	32,14 (14,3)	55,00 (11,74)	48,33 (10,04)
Leu	37,5 (12,68)	30,89 (16,94)	31,96 (16,42)	30,36 (15,52)

4.5.2 Comparações entre **SSF** e **ACA**

Seguindo procedimento análogo àquele usado na Seção 4.5.1 para analisar os resultados obtidos, nas Tabelas 4.10, 4.11 e 4.12 são apresentados o número de Vitórias/Empates/Derrotas comparando-se os algoritmos par-a-par considerando, respectivamente, os critérios número de atributos selecionados, a taxa de erro obtida pelo **NB** e a taxa de erro obtida pelo **KNN**.

Algoritmo	SSF-R-1	SSF-R-2	SSF-SU-1	SSF-SU-2	ACA-1	ACA-2
SSF-R-1	—	12/0/0	0/6/6	10/0/2	2/1/9	6/0/6
SSF-R-2	0/0/12	—	0/0/12	0/5/7	2/0/10	2/1/9
SSF-SU-1	6/6/0	12/0/0	—	12/0/0	2/3/7	7/0/5
SSF-SU-2	2/0/10	7/5/0	0/0/12	—	2/0/10	2/3/7
ACA-1	9/1/2	10/0/2	7/3/2	10/0/2	—	12/0/0
ACA-2	6/0/6	9/1/2	5/0/7	7/3/2	0/0/12	—

Tabela 4.10: Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o número de atributos selecionados entre **SSF-R**, **SSF-SU** e **ACA**.

Algoritmo	SSF-R-1	SSF-R-2	SSF-SU-1	SSF-SU-2	ACA-1	ACA-2
SSF-R-1	—	4/4/4	4/6/2	5/2/5	8/1/3	6/1/5
SSF-R-2	4/4/4	—	7/1/4	4/6/2	9/0/3	8/0/4
SSF-SU-1	2/6/4	4/1/7	—	4/1/7	7/3/2	4/1/7
SSF-SU-2	5/2/5	2/6/4	7/1/4	—	8/0/4	5/1/6
ACA-1	3/1/8	3/0/9	2/3/7	4/0/8	—	3/1/8
ACA-2	5/1/6	4/0/8	7/1/4	6/1/5	8/1/3	—

Tabela 4.11: Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o erro obtido pelo classificador **NB** utilizando os subconjuntos de atributos selecionados pelos algoritmos **SSF-R**, **SSF-SU** e **ACA**.

A Tabela 4.10 mostra que o algoritmo **ACA** tende a selecionar menos atributos do que as variantes do algoritmo **SSF**. O **ACA** selecionou menos ou o mesmo número de atributos em aproximadamente 74% dos casos em relação às outras variantes do **SSF**. Em contrapartida, quando utilizada a medida **SU** no lugar da medida **R** no algoritmo **SSF** o número de atributos selecionados diminuiu na maior parte (56%) das bases de dados.

Algoritmo	SSF-R-1	SSF-R-2	SSF-SU-1	SSF-SU-2	ACA-1	ACA-2
SSF-R-1	—	3/4/5	4/7/1	5/2/5	9/1/2	9/0/3
SSF-R-2	5/4/3	—	6/3/3	4/6/2	9/1/2	7/1/4
SSF-SU-1	1/7/4	3/3/6	—	3/3/6	9/1/2	7/0/5
SSF-SU-2	5/2/5	2/6/4	6/3/3	—	10/0/2	6/1/5
ACA-1	2/1/9	2/1/9	2/1/9	2/0/10	—	2/1/9
ACA-2	3/0/9	4/1/7	5/0/7	5/1/6	9/1/2	—

Tabela 4.12: Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o erro obtido pelo classificador **KNN** utilizando os subconjuntos de atributos selecionados pelos algoritmos **SSF-R**, **SSF-SU** e **ACA**.

Quando o classificador **NB** é utilizado para avaliar os subconjuntos de atributos, o algoritmo **SSF-R** apresentou superioridade em relação ao **ACA**, obtendo resultados melhores ou iguais em aproximadamente 70% das bases de dados, e quando o classificador **KNN** é utilizado, o número de vitórias ou empates aumenta para aproximadamente 80% das bases. Tais resultados sugerem que as diferenças do algoritmo **SSF** em relação ao algoritmo **ACA** (critério de validação de partição e critério para a seleção dos atributos dos grupos) são importantes e permitem a identificação de melhores subconjuntos de atributos pelo **SSF**.

Nas Tabelas 4.13 e 4.14 são apresentadas as categorias da avaliação multicritério obtidas para os classificadores **NB** e **KNN**, respectivamente. É possível constatar que o algoritmo **SSF-R** obteve em geral os melhores resultados. Os testes estatísticos sugerem ($\alpha=10\%$) que os algoritmos **ACA-1**, **ACA-2** e **SSF-SU-1** obtiveram resultados significativamente piores do que a utilização de todos os atributos quando o **KNN** foi utilizado para avaliar os subconjuntos. Quando considerado o classificador **NB**, apenas o algoritmo **ACA-1** obteve resultados significativamente ($\alpha=5\%$) piores do que o uso de todos os atributos. Os testes estatísticos não indicaram diferenças estatisticamente significantes entre os resultados obtidos pelo **ACA** e as variantes do **SSF**. Sob o ponto de vista de número de atributos selecionados, o algoritmo **SSF-R-2** foi pior ($\alpha=5\%$) do que os algoritmos **ACA-1**, **SSF-R-1** e **SSF-SU-1**, assim como o **SSF-SU-2** foi pior do que o **ACA-1** e o **SSF-SU-1**. Conforme esperado, as variantes que selecionam dois atributos por grupo foram mais eficientes computacionalmente ($\alpha=5\%$) do que as variantes que selecionam um atributo por grupo. A única exceção se deu para o par **SSF-SU-1** e **SSF-R-2** que não apresentou diferença estatisticamente significativa.

Base	SSF-R-1	SSF-R-2	SSF-SU-1	SSF-SU-2	ACA-1	ACA-2
Bio1	▲▲▲	▲▲▲	▲▲	▲▲	▲▲	▲▲
Bio2	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲
Bio3	▲▲▲	▲▲▲	▲▲	▲▲	▲▲	▲▲
Bio4	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲
Bio5	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲▲
Yeast	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲	▲▲
Iono	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲	▼
Pima	◇	▼	◇	▼	▼	▼
Wisc	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲
Spam	◇	▲	◇	▲	▼	◇
Colon	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲
Leu	▼	▼	▼	▼	▲▲	▲▲

Tabela 4.13: Avaliação multicritério (Seção 4.2) considerando o erro obtido pelo classificador **NB** utilizando os atributos selecionados pelos algoritmos **SSF-R**, **SSF-SU** e **ACA**.

Algoritmo	SSF-R-1	SSF-R-2	SSF-SU-1	SSF-SU-2	ACA-1	ACA-2
Bio1	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲	▲▲
Bio2	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲
Bio3	▲▲▲	▲▲▲	▲▲	▲▲	▲▲	▲▲
Bio4	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲
Bio5	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲
Yeast	▲▲	▲	▲▲	▲	▲▲	▲▲
Iono	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲	▲▲▲
Pima	▼	▼	▼	▼	▼	▼
Wisc	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲
Spam	▲▲	▲	▲▲	▲	▲▲	▲▲
Colon	▼	▼	▼	◇	▲▲▲	▲▲▲
Leu	◇	▼	◇	▼	▲▲	▲▲

Tabela 4.14: Avaliação multicritério (Seção 4.2) considerando o erro obtido pelo classificador KNN utilizando os atributos selecionados pelos algoritmos SSF-R, SSF-SU e ACA.

Conforme realizado na seção anterior, deseja-se também comparar os dois algoritmos quando estes selecionam (aproximadamente) o mesmo número de atributos. Neste caso, são comparados os algoritmos SSF-R-2 e ACA-2. Para realizar essa comparação foram realizados dois experimentos. No primeiro, o valor de k (número de grupos) é definido pelo algoritmo SSF-R-2 e então o ACA-2 realiza o agrupamento de atributos para este valor de número de grupos. Na sequência, os subconjuntos de atributos selecionados são comparados considerando o erro obtido pelos classificadores. O segundo experimento é análogo, mas neste o valor de k é definido pelo ACA-2. Dessa forma, ambos os algoritmos atuam como se conhecimento de domínio estivesse disponível. Em ambos os experimentos foi considerado o valor de k obtido em cada pasta da validação cruzada, objetivando-se obter estimativas adequadas da capacidade de generalização dos classificadores obtidos. Nas Tabelas 4.15 e 4.16 são apresentados as médias e desvios padrão dos resultados obtidos. O teste estatístico adotado (Wilcoxon *Signed-Rank*) sugere que quando o valor de k foi definido pelo SSF-R-2, o algoritmo SSF-R-2 obteve melhores resultados ($\alpha=10\%$) do que o ACA-2 considerando a taxa de erro obtido pelo classificador NB. Quando o valor de k foi definido pelo ACA-2, considerando a taxa de erro obtida no KNN, o ACA-2 obteve melhores resultados que o SSF-R-2 ($\alpha=5\%$). Tais resultados demonstram uma relação de equilíbrio entre os dois algoritmos quando o valor de “ k ” pode ser definido *a priori* fazendo uso de, por exemplo, conhecimento de domínio disponível.

Tabela 4.15: Erros de classificação — média (desvio padrão) — obtidos pelo SSF-R-2 e ACA-2 utilizando valor de k definido pelo SSF-R-2.

Base	SSF-R-2		ACA-2	
	%-KNN	%-NB	%-KNN	%-NB
Bio1	0,00 (0,00)	0,00 (0,00)	0,25 (0,75)	0,00 (0,00)
Bio2	3,00 (1,87)	2,75 (2,08)	2,00 (1,50)	1,75 (1,95)
Bio3	0,00 (0,00)	0,00 (0,00)	0,00 (0,00)	0,00 (0,00)
Bio4	4,50 (2,69)	5,00 (3,35)	4,75 (2,61)	3,50 (2,00)
Bio5	0,75 (1,14)	0,50 (1,00)	0,25 (0,75)	0,00 (0,00)
Yeast	2,43 (3,23)	1,95 (2,39)	2,95 (2,41)	4,45 (5,22)
Iono	11,39 (5,25)	14,22 (3,91)	11,94 (7,70)	15,08 (6,94)
Pima	33,46 (5,22)	30,21 (6,10)	34,51 (5,91)	29,69 (3,57)
Wisc	6,00 (1,67)	5,28 (2,11)	6,89 (1,76)	5,86 (1,75)
Spam	10,54 (1,29)	22,34 (2,10)	10,17 (1,81)	31,04 (3,17)
Colon	43,10 (24,22)	38,81 (21,3)	42,14 (21,48)	57,14 (21,08)
Leu	37,68 (9,46)	36,07 (15,36)	37,50 (14,29)	38,57 (12,45)

Tabela 4.16: Erros de classificação — média (desvio padrão) — obtidos pelo **SSF-R-2** e **ACA-2** utilizando valor de k definido pelo **ACA-2**.

Base	SSF-R-2		ACA-2	
	%-KNN	%-NB	%-KNN	%-NB
Bio1	0,00 (0,00)	0,25 (0,75)	0,25 (0,75)	0,25 (0,75)
Bio2	2,75 (2,08)	2,50 (2,24)	1,75 (1,14)	1,50 (2,00)
Bio3	1,75 (2,51)	2,50 (1,94)	0,50 (1,00)	1,00 (1,22)
Bio4	8,00 (3,67)	6,00 (1,66)	4,25 (2,25)	4,00 (1,66)
Bio5	0,75 (1,14)	0,50 (1,00)	0,25 (0,75)	0,00 (0,00)
Yeast	10,33 (7,86)	6,38 (5,02)	4,45 (3,49)	7,40 (7,16)
Iono	10,83 (4,76)	12,25 (2,54)	12,83 (5,33)	32,48 (4,49)
Pima	38,02 (5,01)	32,30 (5,31)	37,25 (5,25)	34,52 (5,54)
Wisc	6,00 (1,67)	5,28 (2,11)	6,89 (1,76)	5,86 (1,75)
Spam	19,82 (2,89)	43,73 (6,04)	13,39 (1,03)	34,10 (1,84)
Colon	25,48 (14,13)	30,71 (26,03)	24,29 (13,38)	42,38 (17,65)
Leu	25,00 (5,53)	20,71 (12,86)	19,29 (13,87)	8,04 (8,65)

4.5.3 Comparações entre Variantes Supervisionadas do **SSF**

Para avaliar potenciais ganhos obtidos com a incorporação da informação da classe no filtro **SSF** (Seção 3.4) foram executados experimentos análogos àqueles reportados previamente.

Nas Tabelas 4.17, 4.18 e 4.19 são apresentados o número de Vitórias/Empates/Derrotas comparando-se os algoritmos par-a-par considerando, respectivamente, o número de atributos selecionados, a taxa de erro obtida pelo classificador **NB** e a taxa de erro obtida pelo classificador **KNN**. Na Tabela 4.17, é possível verificar que a incorporação da informação da classe teve como consequência um número maior de atributos selecionados. Mais especificamente, ao se comparar cada variante supervisionada do **SSF** com a sua correspondente não-supervisionada (*e.g.*, **SSF-SU_S-2** e **SSF-SU-2**), pode-se observar que as variantes supervisionadas selecionaram mais atributos em pelo menos 67% dos casos. Por sua vez, os algoritmos **SSF-SU_S-KS-2** e **SSF-SU_S-KS- $\bar{I}$** , que utilizam informação da classe no cálculo da correlação e do medóide, obtiveram o pior desempenho dentre todos, obtendo resultados piores que os outros algoritmos em pelo menos 80% dos casos.

Algoritmo	SSF-SU-1	SSF-SU-2	SSF-SU_S-1	SSF-SU_S-2	SSF-SU_S-KS-1	SSF-SU_S-KS-2	SSF-SU_S-$\bar{I}$	SSF-SU_S-KS-$\bar{I}$
SSF-SU-1	—	12/0/0	8/1/3	12/0/0	10/1/1	12/0/0	12/0/0	12/0/0
SSF-SU-2	0/0/12	—	6/0/6	8/1/3	7/0/5	10/1/1	8/1/3	10/1/1
SSF-SU_S-1	3/1/8	6/0/6	—	12/0/0	7/2/3	12/0/0	12/0/0	12/0/0
SSF-SU_S-2	0/0/12	3/1/8	0/0/12	—	1/0/11	10/1/1	0/12/0	10/1/1
SSF-SU_S-KS-1	1/1/10	5/0/7	3/2/7	11/0/1	—	12/0/0	11/0/1	12/0/0
SSF-SU_S-KS-2	0/0/12	1/1/10	0/0/12	1/1/10	0/0/12	—	1/1/10	0/12/0
SSF-SU_S-$\bar{I}$	0/0/12	3/1/8	0/0/12	0/12/0	1/0/11	10/1/1	—	10/1/1
SSF-SU_S-KS-$\bar{I}$	0/0/12	1/1/10	0/0/12	1/1/10	0/0/12	0/12/0	1/1/10	—

Tabela 4.17: Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o número de atributos selecionados entre as variantes supervisionadas do **SSF** e o **SSF-SU**.

Alguns resultados reportados nas Tabelas 4.18 e 4.19 merecem destaque. A incorporação da supervisão (informação da classe) no cálculo da correlação (uso da medida **SU_S** no lugar da medida **SU**) trouxe apenas uma pequena redução nas taxas de erro obtidas, *i.e.* **SSF-SU_S-1** obteve resultados melhores do que o algoritmo **SSF-SU-1** em pelo menos 50% dos casos (para ambos os classificadores). No entanto, a modificação na determinação do “atributo fronteira”

Algoritmo	SSF-SU-1	SSF-SU-2	SSF-SU _S -1	SSF-SU _S -2	SSF-SU _S -KS-1	SSF-SU _S -KS-2	SSF-SU _S -I	SSF-SU _S -KS-I
SSF-SU-1	—	4/1/7	4/1/7	3/0/9	5/0/7	2/0/10	4/0/8	4/0/8
SSF-SU-2	7/1/4	—	5/0/7	2/0/10	4/1/7	0/1/11	3/0/9	2/1/9
SSF-SU _S -1	7/1/4	7/0/5	—	1/3/8	4/4/4	0/3/9	3/2/7	1/3/8
SSF-SU _S -2	9/0/3	10/0/2	8/3/1	—	7/3/2	2/4/6	7/4/1	3/5/4
SSF-SU _S -KS-1	7/0/5	7/1/4	4/4/4	2/3/7	—	1/3/8	3/2/7	1/4/7
SSF-SU _S -KS-2	10/0/2	11/1/0	9/3/0	6/4/2	8/3/1	—	8/4/0	5/6/1
SSF-SU _S -I	8/0/4	9/0/3	7/2/3	1/4/7	7/2/3	0/4/8	—	2/3/7
SSF-SU _S -KS-I	8/0/4	9/1/2	8/3/1	4/5/3	7/4/1	1/6/5	7/3/2	—

Tabela 4.18: Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando a taxa de erro obtida pelo NB entre as variantes supervisionadas do SSF e o SSF-SU.

Algoritmo	SSF-SU-1	SSF-SU-2	SSF-SU _S -1	SSF-SU _S -2	SSF-SU _S -KS-1	SSF-SU _S -KS-2	SSF-SU _S -I	SSF-SU _S -KS-I
SSF-SU-1	—	3/3/6	5/1/6	1/1/10	2/1/9	1/1/10	2/1/9	2/1/9
SSF-SU-2	6/3/3	—	6/1/5	1/1/10	4/2/6	1/1/10	2/1/9	3/1/8
SSF-SU _S -1	6/1/5	5/1/6	—	1/3/8	3/3/6	1/3/8	3/2/7	1/3/8
SSF-SU _S -2	10/1/1	10/1/1	8/3/1	—	7/3/2	2/3/7	4/4/4	5/3/4
SSF-SU _S -KS-1	9/1/2	6/2/4	6/3/3	2/3/7	—	2/3/7	3/2/7	2/3/7
SSF-SU _S -KS-2	10/1/1	10/1/1	8/3/1	7/3/2	7/3/2	—	6/3/3	6/5/1
SSF-SU _S -I	9/1/2	9/1/2	7/2/3	4/4/4	7/2/3	3/3/6	—	4/2/6
SSF-SU _S -KS-I	9/1/2	8/1/3	8/3/1	4/3/5	7/3/2	1/5/6	6/2/4	—

Tabela 4.19: Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando a taxa de erro obtida pelo KNN entre as variantes supervisionadas do SSF e o SSF-SU.

para considerar a correlação com a classe trouxe uma sensível melhora nos resultados, *i.e.* SSF-SU_S-2 obteve resultados melhores do que o algoritmo SSF-SU-2 em mais de 80% dos casos. A modificação na definição do atributo medóide também trouxe melhoras importantes. Em particular, os algoritmos que consideram a correlação com a classe para a definição dos medóides (SSF-SU_S-KS-*) tiveram resultados iguais ou melhores do que seus correspondentes utilizando a definição original de medóide (SSF-SU_S-*) em pelo menos 67% dos casos. A utilização da Informação Mútua Condicionada Normalizada (IMCN) — definida na Seção 3.4 como $\bar{I}$ — para a seleção do “atributo fronteira” não trouxe ganhos significativos.

As tabelas 4.20 e 4.21 apresentam os resultados de acordo com a avaliação multicritério considerando, respectivamente, os erros obtidos pelos classificadores NB e KNN. De acordo com os testes de Friedman e de Nemenyi, o algoritmo SSF-SU_S-KS-2 obteve taxas de erro significativamente menores ($\alpha=10\%$) do que os algoritmos SSF-SU-1, SSF-SU-2 e SSF-SU_S-1 em ambos os classificadores. Isso demonstra que a incorporação da informação da classe (supervisão) não apenas no cálculo da correlação mas também durante o processo de agrupamento pode trazer benefícios significativos. Ao usar o classificador NB para avaliar os subconjuntos, os testes estatísticos indicam que o algoritmo SSF-SU_S-KS-2 apresentou melhores resultados do que o algoritmo SSF-SU_S-KS-1 ($\alpha=10\%$). Quando o KNN foi utilizado, o algoritmo SSF-SU-1 apresentou piores resultados em relação à utilização de todos os atributos ($\alpha=5\%$). Levando em consideração o número de atributos selecionados, o algoritmo SSF-SU-1 apresentou melhores resultados do que todos os algoritmos que selecionam dois atributos por grupo, com exceção do SSF-SU-2 ($\alpha=5\%$). Assim como os algoritmos SSF-SU-2, SSF-SU_S-1 e SSF-SU_S-KS-1 selecionaram significativamente menos atributos que os algoritmos SSF-SU_S-KS-2 e SSF-SU_S-KS- $\bar{I}$. Sob o ponto de vista de custo computacional, os algoritmos que selecionam dois atributos por

grupo foram significativamente mais eficientes ($\alpha=5\%$) do que seus correspondentes que selecionam apenas um atributo por grupo.

Algoritmo	Bio1	Bio2	Bio3	Bio4	Bio5	Yeast	Iono	Pima	Wisc	Spam	Colon	Leu
SSF-SU-1	▲▲	▲▲	▲▲	▲▲	▲▲▲	▲▲▲	▲▲▲	◇	▲▲	◇	▲▲▲	▼
SSF-SU-2	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲▲	▲▲▲	▼	▲▲	▲	▲▲▲	▼
SSF-SU _S -1	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲▲	▲▲▲	▼	▲▲	◇	▲▲▲	▼
SSF-SU _S -2	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲	▲▲▲	▲	▲▲▲	▲▲
SSF-SU _S -KS-1	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲	▲▲	▼	▲▲	▲▲	▲▲▲	▲▲
SSF-SU _S -KS-2	▲▲▲	▲▲▲	▲▲▲	▲	▲▲▲	▲▲▲	▲▲▲	▲	▲▲▲	▲▲▲	▲▲▲	▲▲
SSF-SU _S -I	▲▲▲	▲▲▲	▲▲▲	▲	▲▲▲	▲▲▲	▲▲▲	▼	▲▲▲	▲	▲▲▲	▲▲
SSF-SU _S -KS-I	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▼	▲▲▲	▲	▲▲▲	▲▲

Tabela 4.20: Avaliação multicritério (Seção 4.2) considerando o erro obtido pelo classificador NB utilizando os atributos selecionados pelas variantes supervisionadas do SSF e o SSF-SU.

Algoritmo	Bio1	Bio2	Bio3	Bio4	Bio5	Yeast	Iono	Pima	Wisc	Spam	Colon	Leu
SSF-SU-1	▲▲▲	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲▲	▼	▲▲	▲▲	▼	◇
SSF-SU-2	▲▲▲	▲▲	▲▲	▲▲	▲▲	▲	▲▲▲	▼	▲▲	▲	◇	▼
SSF-SU _S -1	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▼	▲▲	▲▲	▼	◇
SSF-SU _S -2	▲▲▲	▲	▲▲▲	▲	▲▲▲	▲▲▲	▲	▼	▲▲▲	▲	▲▲	▲▲▲
SSF-SU _S -KS-1	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▼	▲▲	▲▲	◇	▲▲▲
SSF-SU _S -KS-2	▲▲▲	▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲	▲▲▲	▲	▲▲▲	▲▲▲
SSF-SU _S -I	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▼	▲▲▲	▲	▲▲	▲▲▲
SSF-SU _S -KS-I	▲▲▲	▲	▲▲▲	▲▲▲	▲▲▲	▲	▲▲▲	▼	▲▲▲	▲	▲▲	▲▲▲

Tabela 4.21: Avaliação multicritério (Seção 4.2) considerando o erro obtido pelo classificador KNN utilizando os atributos selecionados pelas variantes supervisionadas do SSF e o SSF-SU.

Sob a perspectiva de uma aplicação de SA em um problema de classificação, com base nos resultados discutidos, pode-se concluir que caso o objetivo principal da aplicação de SA seja reduzir ao máximo o número de atributos sem afetar significativamente a taxa de erro obtida pelo classificador a ser utilizado, a variante SSF-SU_S-2 é a mais indicada. Caso contrário, se o objetivo principal for reduzir a taxa de erro do classificador, mesmo que poucos atributos sejam eliminados, a variante SSF-SU_S-KS-2 é a mais recomendada.

4.6 Estudos de Caso

4.6.1 Classificador Inteligente de Amostras de Café

Nesta seção é apresentado um estudo de caso de seleção de atributos para a classificação de grãos de café. A base de dados real utilizada neste estudo de caso, está vinculada ao projeto CIAC (Classificador Inteligente de Amostras de Café) desenvolvido pela empresa NatComp⁶ com apoio da FAPESP⁷ e a ACS (Associação Comercial de Santos). Por razões de confidencialidade, não são descritos com maiores detalhes os atributos utilizados nas bases de dados, assim como detalhes sobre a forma de obtenção dos mesmos.

O agronegócio mundial do café engloba, anualmente, recursos que chegam a 91 bilhões de dólares e envolve meio bilhão de pessoas (aproximadamente 8% da população mundial). É

⁶<http://www.natcomp.com.br>

⁷Fundação de Amparo à Pesquisa do Estado de São Paulo, Processo nº 2006/06703-0.

nesse mercado gigantesco que está centrado o interesse da cadeia produtiva do café brasileiro, que contribuiu com mais de 30% da produção mundial nas últimas safras. O mercado do café é caracterizado por um conjunto de atividades de enorme complexidade, dinamismo, e um crescente nível de exigência dos consumidores quanto à qualidade da bebida. Isso impõe aos países produtores, consumidores e exportadores um elevado controle de qualidade. Atualmente, a definição da qualidade e, portanto, do valor do café é baseada na classificação manual, ou seja, uma pessoa exerce o papel de um classificador treinado (certificado) para qualificar as amostras de café. Assim, o processo atual de classificação de café sofre com a subjetividade dos classificadores e uma grande dificuldade de padronização do processo devido a possíveis inconsistências dos classificadores.

Diante deste cenário, o projeto CIAC propõe desenvolver uma máquina (conjunto *hardware* + *software*), para classificar amostras de café. A máquina proposta do ponto de vista de *software* consiste na extração de características de imagens obtidas dos grãos de café e na aplicação de algoritmos de reconhecimento de padrões para a classificação dos grãos. Neste estudo de caso, será abordada apenas a etapa de seleção de atributos para o reconhecimento de padrões.

A classificação dos grãos de café consiste, especificamente, na discriminação dos grãos de café entre 14 classes possíveis. Sendo 13 dessas classes determinadas por defeitos existentes nos grãos, enquanto a 14^a classe refere-se a grãos sem defeitos.

As análises preliminares em relação à seleção de atributos para este problema, considerando uma base de dados formada por 771 grãos e 35 atributos, foram publicadas em (Covões et al., 2008). Nessa etapa preliminar foi possível verificar que a seleção de atributos, através dos métodos *Correlation based Feature Selection* (CFS) (Hall, 1999), *Consistency Based Evaluation* (CBE) (Liu e Setiono, 1996) e *Wrapper* (WRP) (Kohavi e John, 1997) — uma breve descrição desses métodos é apresentada na Seção 2.4 —, afeta positivamente a acurácia obtida pelos classificadores *Naïve Bayes* (NB) e *k-Nearest Neighbors* (KNN) na classificação dos grãos dentre as 14 classes possíveis.

Neste trabalho, a comparação entre os algoritmos de SA é realizada em uma base de dados formada por 70 atributos e 1884 grãos de café. Nesta comparação foram considerados além dos três métodos utilizados nas análises preliminares (CFS, CBE e WRP), os métodos *Fast Correlation Based Filter* (FCBF) (Yu e Liu, 2003), ACA (Au et al., 2005), MMP (Mitra et al., 2002) e SSF. Para a classificação foram analisados os classificadores NB e *Support Vector Machine* (SVM).

A classificação dos grãos foi separada em dois problemas de classificação distintos. No primeiro, um classificador deve diferenciar entre duas classes: grão defeituoso ou grão normal. No segundo, um classificador deve diferenciar entre as 13 classes de defeitos possíveis.

Os resultados obtidos para a classificação em duas classes estão sumarizados na Tabela 4.22. Para o classificador NB, os algoritmos de seleção de atributos FCBF, CBE e WRP obtiveram os melhores resultados, não apenas diminuindo o erro obtido pelo classificador (em relação à utilização de todos os atributos) como também reduzindo sensivelmente o número de atributos,

Método	M^*	%-NB	%-SVM
CFS	18,0 (2,3)	9,34 (1,71)	6,58 (2,10)
FCBF	5,9 (0,5)	9,61 (2,54)	7,86 (2,16)
CBE	9,2 (0,6)	9,61 (1,94)	5,68 (2,50)
ACA-1	18,8 (2,9)	14,07 (2,03)	8,39 (2,25)
ACA-2	37,3 (5,7)	14,33 (2,11)	6,16 (1,75)
SSF- ρ -1	23,9 (2,1)	12,74 (1,88)	5,95 (1,58)
SSF- ρ -2	45,5 (3,4)	12,42 (2,03)	3,82 (1,38)
SSF- λ -1	2,0 (0,0)	19,48 (2,68)	23,35 (0,00)
SSF- λ -2	3,0 (0,0)	13,91 (1,96)	15,61 (2,95)
SSF-R-1	34,1 (0,3)	12,74 (2,35)	4,56 (1,59)
SSF-R-2	60,1 (0,3)	11,79 (1,97)	3,29 (1,32)
SSF-SU-1	34,1 (0,3)	12,37 (2,36)	4,88 (1,95)
SSF-SU-2	59,9 (0,5)	11,79 (1,97)	3,29 (1,32)
SSF-SU _S -1	33,3 (3,0)	12,64 (2,01)	5,36 (1,47)
SSF-SU _S -2	59,2 (3,5)	11,57 (1,88)	3,40 (1,14)
SSF-SU _S -KS-1	32,3 (3,9)	11,57 (1,87)	5,79 (1,47)
SSF-SU _S -KS-2	56,9 (4,0)	11,68 (2,01)	3,50 (1,07)
SSF-SU _S -I	59,2 (3,5)	11,57 (1,88)	3,29 (1,05)
SSF-SU _S -KS-I	56,9 (4,0)	11,84 (2,05)	3,66 (1,15)
WRP-NB	6,3 (0,8)	5,57 (1,65)	—
WRP-SVM	43,8 (7,9)	—	2,92 (1,10)
MMP (M)	41,9 (26,4)	10,41 (2,10)	—
MMP (M)	54,0 (4,1)	—	3,08 (1,17)
Todos	70	11,89 (2,05)	3,24 (1,36)

Tabela 4.22: Erros de classificação — média (desvio padrão) — obtidos na base de dados do Projeto CIAC considerando a classificação em 2 classes.

selecionando em média menos do que 10 atributos. Em relação ao classificador SVM, apenas os métodos WRP e MMP conseguiram reduzir o erro obtido pelo classificador (em relação à utilização de todos os atributos) selecionando aproximadamente 43 e 54 atributos, respectivamente. As variantes do SSF que selecionam dois atributos por grupo (com exceção do SSF- λ -2) obtiveram resultados muito próximos à utilização de todos os atributos, selecionando no geral mais de 56 atributos. Embora tenham removido poucos atributos, algumas dessas variantes, por exemplo SSF-SU_S-I, obtiveram quase a metade do erro obtido pelos filtros supervisionados avaliados (CFS, FCBF e CBE) quando o classificador SVM foi utilizado para avaliar os subconjuntos.

Os resultados obtidos para a classificação em treze classes (defeitos) estão sumarizados na Tabela 4.23. Para o classificador NB, os subconjuntos de atributos obtidos pelo CFS, WRP e MMP obtiveram melhores resultados que a utilização de todos os atributos. Para o classificador SVM o método MMP foi o único a produzir redução do número de erros do classificador, removendo aproximadamente 10 atributos. Assim como ocorreu para o problema de classificação em duas classes, algumas variantes do SSF, por exemplo SSF-SU_S-KS-2, removeram poucos atributos (aproximadamente 10 atributos) porém obtiveram taxas de erro menores que os filtros supervisionados em ambos os classificadores, sendo a única exceção o CFS quando avaliado pelo classificador NB.

Por fim, uma categorização dos resultados obtidos por cada método de seleção de atributos de acordo com a avaliação multicritério é descrita na Tabela 4.24. É possível verificar que, segundo essa forma de avaliação, os métodos CFS, WRP e MMP foram os melhores, obtendo Excelente (▲▲▲) em pelo menos 2 dos 4 experimentos e Muito Bom (▲▲) nos demais. O

Método	M^*	%-NB	%-SVM
CFS	30,3 (1,4)	24,38 (3,32)	24,52 (2,46)
FCBF	9,0 (0,4)	29,23 (4,11)	28,95 (2,88)
CBE	12,6 (1,2)	30,33 (4,73)	29,23 (4,03)
ACA-1	17,7 (2,1)	33,10 (2,53)	30,82 (4,08)
ACA-2	34,7 (3,2)	33,59 (2,71)	26,66 (3,89)
SSF- ρ -1	23,8 (1,8)	29,15 (3,50)	28,05 (2,94)
SSF- ρ -2	46,2 (4,0)	29,29 (3,39)	22,71 (2,91)
SSF- λ -1	2,0 (0,0)	63,09 (4,63)	64,33 (2,81)
SSF- λ -2	3,0 (0,0)	60,94 (3,87)	62,40 (2,87)
SSF-R-1	34,1 (0,3)	29,71 (2,64)	26,87 (2,49)
SSF-R-2	60,1 (0,3)	27,29 (3,05)	21,95 (3,49)
SSF-SU-1	34,1 (0,3)	29,78 (2,53)	26,59 (2,22)
SSF-SU-2	60,0 (0,4)	27,43 (2,93)	21,95 (3,31)
SSF-SU _S -1	31,0 (3,1)	29,99 (2,52)	26,94 (2,53)
SSF-SU _S -2	56,3 (3,7)	28,19 (3,25)	22,30 (2,18)
SSF-SU _S -KS-1	34,0 (0,0)	27,98 (2,63)	24,79 (3,02)
SSF-SU _S -KS-2	61,0 (0,0)	27,29 (2,25)	21,05 (3,76)
SSF-SU _S -I	56,3 (3,7)	28,46 (3,40)	22,58 (2,39)
SSF-SU _S -KS-I	61,0 (0,0)	27,43 (2,10)	21,88 (4,04)
WRP-NB	15,9 (1,7)	24,10 (2,60)	—
WRP-SVM	29,3 (9,1)	—	23,20 (2,57)
MMP (M)	55,4 (9,0)	25,63 (2,17)	—
MMP (M)	60,1 (6,1)	—	19,94 (3,13)
Todos	70	26,59 (2,08)	20,50 (3,36)

Tabela 4.23: Erros de classificação — média (desvio padrão) — obtidos na base de dados do Projeto CIAC considerando a classificação em 13 classes.

Método	Classificação em 2 classes		Classificação em 13 classes	
	NB	SVM	NB	SVM
CFS	▲▲▲	▲▲	▲▲▲	▲▲
FCBF	▲▲▲	▲▲	▲▲	▲▲
CBE	▲▲▲	▲▲	▲▲	▲▲
ACA-1	▲▲	▲▲	▲▲	▲▲
ACA-2	▲	▲	▲▲	▲▲
SSF- ρ -1	▲▲	▲▲	▲▲	▲▲
SSF- ρ -2	▲	▲	▲	▲
SSF- λ -1	◇	▼	▼	▼
SSF- λ -2	▲▲	◇	▼	▼
SSF-R-1	▲▲	▲▲	▲▲	▲▲
SSF-R-2	▲▲▲	▲	▲	▲
SSF-SU-1	▲▲	▲▲	▲▲	▲▲
SSF-SU-2	▲▲▲	▲	▲	▲
SSF-SU _S -1	▲▲	▲▲	▲▲	▲▲
SSF-SU _S -2	▲▲▲	▲	▲	▲
SSF-SU _S -KS-1	▲▲▲	▲▲	▲▲	▲▲
SSF-SU _S -KS-2	▲▲▲	▲	▲	▲
SSF-SU _S -I	▲▲▲	▲	▲	▲
SSF-SU _S -KS-I	▲▲▲	▲	▲	▲
WRP	▲▲▲	▲▲▲	▲▲▲	▲▲
MMP	▲▲▲	▲▲▲	▲▲▲	▲▲▲

Tabela 4.24: Avaliação multicritério (Seção 4.2) dos resultados na base de dados do Projeto CIAC.

SSF- λ obteve os piores resultados. As demais variantes do SSF tiveram resultados semelhantes entre si, sendo categorizados como Bom (▲) ou Muito Bom (▲▲) na maioria dos experimentos.

4.6.2 Read the Web

O projeto *Read the Web*⁸ é desenvolvido na Universidade de Carnegie Mellon, e seu principal objetivo é a criação de uma base de conhecimento simbólica probabilística que contenha parte significativa do conteúdo da internet. Para tal, um sistema de classificação semi-supervisionada utilizando o conceito de *aprendizado sem fim* vem sendo desenvolvido.

O sistema opera sobre os seguintes conceitos básicos (Carlson et al., 2009):

- Ontologia: coleção de predicados unários (categorias) e binários (relações);
- Instâncias de categorias: são substantivos. Por exemplo, para a categoria “empresa”, uma instância possível é “Disney”;
- Instância de uma Relação: é um par de substantivos. Por exemplo, para a relação indústriaEmpresa(empresa, indústria), uma instância dessa relação é <“Disney”, “entretenimento”>;
- Padrões: conjunto de palavras com lacunas, aqui denotadas por “arg”, associados às categorias e a relações (e.g., são padrões: “jogo contra arg1”, “arg1, treinador de arg2”).

A idéia básica do sistema é que, dada uma ontologia, o sistema deve ser capaz de populá-la, realizando uma varredura em páginas da internet, com novas instâncias de categorias e de relações de alta confiança (Carlson et al., 2009). Classificadores são usados no processo de aprendizado e, portanto, técnicas de SA podem ser úteis ao sistema.

Neste estudo de caso é analisado um pequeno subconjunto da base de dados utilizada no sistema real. Especificamente, é utilizada a base de dados que consiste de substantivos que devem ser classificados como pertencentes ou não à categoria “cidade”. A base de dados é formada por 9415 substantivos (objetos) e 22 padrões (atributos), sendo as duas classes distribuídas em 75,8% de substantivos que não são cidades e 24,2% de substantivos que são cidades.

Além das variantes do SSF, foi avaliado o uso do algoritmo WRP (Kohavi e John, 1997) para a seleção de atributos. Os subconjuntos de atributos foram avaliados utilizando-se de três classificadores, a saber: *Naïve Bayes* (NB), Regressão Logística (RL) e *Classification and Regression Trees* (CART).

As Tabelas 4.25 apresenta os valores das médias e desvios padrão obtidos na validação cruzada, enquanto que a Tabela 4.26 reporta os resultados obtidos a partir da categorização dos resultados de acordo com a avaliação multicritério. Os algoritmos SSF-SU_g-KS-2 e SSF-SU_g-KS- $\bar{I}$ se destacaram entre as variantes do algoritmo SSF por terem obtido as menores taxas de erro nos três classificadores. Ambos os algoritmos obtiveram resultados muito similares ao WRP (com exceção do WRP-NB), com um custo computacional inferior, oferecendo portanto uma boa relação de custo/benefício.

⁸<http://rtw.ml.cmu.edu>

Tabela 4.25: Erros de classificação — média (desvio padrão) — obtidos na base de dados do Projeto “*Read the Web*”.

Algoritmo	M^*	%-NB	%-RL	%-CART
SSF- λ -1	2,0 (0,0)	20,77 (2,57)	18,15 (2,29)	18,15 (2,29)
SSF- λ -2	3,0 (0,0)	19,18 (2,06)	17,05 (1,80)	17,05 (1,80)
SSF- ρ -1	2,2 (0,6)	20,21 (2,13)	18,57 (2,08)	18,57 (2,08)
SSF- ρ -2	3,5 (1,5)	19,71 (1,34)	17,58 (2,64)	17,58 (2,64)
SSF-R-1	8,6 (1,2)	20,34 (0,54)	10,65 (0,52)	10,65 (0,52)
SSF-R-2	15,6 (1,2)	18,87 (0,53)	7,71 (0,60)	7,75 (0,61)
SSF-SU-1	8,6 (1,2)	20,34 (0,54)	10,65 (0,52)	10,65 (0,52)
SSF-SU-2	15,6 (1,2)	18,87 (0,53)	7,71 (0,60)	7,75 (0,61)
SSF-SU _S -1	10,4 (0,9)	19,54 (0,67)	10,37 (0,54)	10,37 (0,54)
SSF-SU _S -2	17,0 (0,6)	18,81 (0,79)	7,34 (0,41)	7,42 (0,41)
SSF-SU _S -KS-1	9,2 (0,8)	19,19 (0,54)	8,86 (1,03)	8,87 (1,04)
SSF-SU _S -KS-2	17,3 (1,1)	18,01 (0,81)	6,59 (0,58)	6,68 (0,61)
SSF-SU _S - $\bar{I}$	17,0 (0,6)	18,76 (0,71)	7,39 (0,44)	7,45 (0,45)
SSF-SU _S -KS- $\bar{I}$	17,3 (1,1)	18,00 (0,81)	6,57 (0,59)	6,67 (0,62)
WRP-NB	3,1 (0,3)	14,63 (0,89)	—	—
WRP-RL	18,9 (0,7)	—	6,32 (0,47)	—
WRP-CART	13,7 (0,4)	—	—	6,39 (0,48)
Todos	22	17,90 (0,80)	6,29 (0,47)	6,36 (0,48)

Tabela 4.26: Avaliação multicritério (Seção 4.2) dos resultados na base de dados do Projeto “*Read the Web*”.

	NB	RL	CART
SSF- λ -1	▲▲	◇	◇
SSF- λ -2	▲▲	◇	◇
SSF- ρ -1	▲▲	◇	◇
SSF- ρ -2	▲▲	◇	◇
SSF-R-1	▲▲	▲▲	▲▲
SSF-R-2	▲	▲	▲
SSF-SU-1	▲▲	▲▲	▲▲
SSF-SU-2	▲	▲	▲
SSF-SU _S -1	▲▲	▲▲	▲▲
SSF-SU _S -2	▲	▲	▲
SSF-SU _S -KS-1	▲▲	▲▲	▲▲
SSF-SU _S -KS-2	▲	▲	▲
SSF-SU _S - $\bar{I}$	▲	▲	▲
SSF-SU _S -KS- $\bar{I}$	▲	▲	▲
WRP	▲▲▲	▲	▲

Uma importante característica de algoritmos de SA baseados no agrupamento de atributos é a possibilidade da realização de um pós-processamento, baseando-se no agrupamento de atributos encontrado. Para ilustrar esta característica, na Figura 4.5 é apresentado um grafo não direcionado, onde cada vértice representa um atributo. Este grafo representa a melhor partição encontrada pelo algoritmo SSF-R-2 em cada pasta da validação cruzada. Atributos que estiveram no mesmo grupo em pelo menos duas pastas da validação cruzada possuem um aresta conectando-os. A espessura dessa aresta é proporcional ao número de vezes em que os atributos estiveram no mesmo grupo. De forma similar, a espessura do contorno do vértice de cada atributo indica o número de vezes em que o atributo foi selecionado como medóide do grupo. Por exemplo, como esperado, os atributos “cities like _” e “cities such as _” são correlacionados, e portanto, “cities like _” foi selecionado como medóide em diversas pastas da validação cruzada. Este tipo de visualização de relações entre atributos podem auxiliar especialistas de domínio na determinação sobre inclusão/exclusão de atributos na base de dados sendo considerada.

Figura 4.5: Visualização dos grupos de atributos obtidos pelo SSF-R-2 na base de dados do Projeto “Read the Web”.

Resultados Experimentais em Problemas de Agrupamento de Dados

5.1 Considerações Iniciais

Neste capítulo são descritos e avaliados os experimentos realizados envolvendo bases de dados artificiais e reais de agrupamento de dados. Os resultados aqui apresentados foram publicados em (Covões e Hruschka, 2009). A Seção 5.2 descreve a metodologia utilizada para avaliar os algoritmos de seleção de atributos. Na Seção 5.3 os algoritmos de seleção de atributos avaliados são especificados, assim como os parâmetros que foram utilizados. A Seção 5.4 descreve as bases de dados utilizadas nos experimentos. Finalmente, na Seção 5.5 são apresentados e discutidos os resultados obtidos.

5.2 Metodologia de Avaliação

Os subconjuntos de atributos selecionados pelos algoritmos foram avaliados utilizando o algoritmo *k-means* (Hartigan e Wong, 1979), que é amplamente conhecido na literatura e usado na prática (Wu et al., 2008). Foi utilizada a implementação do *k-means* disponível no sistema WEKA. O número de grupos foi definido como o número real de grupos de cada base de dados conhecido *a priori*. Os demais parâmetros permaneceram com os valores *default* do WEKA. Como o *k-means* sofre da limitação de poder ficar preso em mínimos locais, foram executadas 50 repetições, inicializadas aleatoriamente, para cada base de dados utilizada na avaliação. A melhor partição (dentre as 50) foi determinada considerando o menor erro quadrático, definido como:

$$ErroQuadrático = \sum_{r=1}^k \sum_{\mathbf{x}_i \in G_r} d(\mathbf{x}_i, \bar{\mathbf{g}}_r), \quad (5.1)$$

onde k é o número de grupos da partição, $\bar{\mathbf{g}}_r$ denota o centróide do r -ésimo grupo e $d(\mathbf{x}_i, \bar{\mathbf{g}}_r)$ denota a distância Euclidiana entre o objeto $\mathbf{x}_i$ e o centróide $\bar{\mathbf{g}}_r$.

Para avaliar os resultados obtidos por um algoritmo de agrupamento tendo uma partição como referência, como é o caso neste trabalho, podem ser utilizados *critérios externos* para comparar o agrupamento obtido com a partição de referência (Jain e Dubes, 1988). Por exemplo, considere que $\mathcal{U} = \{u_1, u_2, \dots, u_q\}$ e $\mathcal{V} = \{v_1, v_2, \dots, v_z\}$ são duas partições de dados com, respectivamente, q e z grupos. Uma destas partições (e.g., $\mathcal{U}$) pode ser considerada a partição de referência, enquanto que a outra (e.g., $\mathcal{V}$) é considerada a partição obtida a partir dos dados. A tabela de contingência obtida a partir das duas partições é apresentada na Tabela 5.1, na qual a célula n_{ij} contém o número de objetos que estão em ambos os grupos u_i e v_j . O termo $n_{i.}$ é o número total de objetos no grupo u_i , assim como $n_{.j}$ é o número total de objetos no grupo v_j (Jain e Dubes, 1988).

	v_1	v_2	$\dots$	v_z	
u_1	n_{11}	n_{12}	$\dots$	n_{1z}	$n_{1.}$
u_2	n_{21}	n_{22}	$\dots$	n_{2z}	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
u_q	n_{q1}	n_{q2}	$\dots$	n_{qz}	$n_{q.}$
	$n_{.1}$	$n_{.2}$	$\dots$	$n_{.z}$	$n_{..} = N$

Tabela 5.1: Tabela de contingência para duas partições. Adaptada de Jain e Dubes (1988).

Os critérios externos são usualmente descritos utilizando as seguintes quantidades deduzidas da Tabela 5.1:

- (a) número de pares de objetos que estão no mesmo grupo em ambas as partições:

$$a = \frac{1}{2} \left[\sum_{i=1}^q \sum_{j=1}^z n_{ij}^2 - N \right];$$

- (b) número de pares de objetos que estão no mesmo grupo em $\mathcal{U}$ mas em diferentes grupos em $\mathcal{V}$:

$$b = \frac{1}{2} \left[\sum_{j=1}^z n_{.j}^2 - \sum_{i=1}^q \sum_{j=1}^z n_{ij}^2 \right];$$

- (c) número de pares de objetos que estão em grupos diferentes em $\mathcal{U}$ mas no mesmo grupo em $\mathcal{V}$:

$$c = \frac{1}{2} \left[\sum_{i=1}^q n_{i.}^2 - \sum_{i=1}^q \sum_{j=1}^z n_{ij}^2 \right];$$

- (d) número de pares de objetos que estão em grupos diferentes em ambas as partições:

$$d = \binom{n}{2} - a - b - c = \frac{n(n+1)}{2} - \frac{1}{2} \left[\sum_{i=1}^q n_{i.}^2 + \sum_{j=1}^z n_{.j}^2 \right].$$

Dois critérios externos usualmente utilizados na literatura são: Coeficiente de Jaccard (**CJ**) (**Jaccard, 1908**) e *Adjusted Rand Index* (**ARI**)¹ (**Hubert e Arabie, 1985**). Ambos critérios apresentam resultados no intervalo [0,1], sendo que quanto maior a concordância entre as duas partições, maior o valor obtido. Mais especificamente, **CJ** e **ARI** podem ser descritos da seguinte forma (**Steinley, 2004**):

$$Jaccard = \frac{a}{a + b + c}, \quad (5.2)$$

$$ARI = \frac{\binom{N}{2}(a + d) - [(a + b)(a + c) + (c + d)(b + d)]}{\binom{N}{2}^2 - [(a + b)(a + c) + (c + d)(b + d)]}. \quad (5.3)$$

Dessa forma a melhor partição, encontrada pelo *k-means* dentre as 50 repetições considerando a Equação 5.1, foi comparada com a partição de referência da base de dados utilizando os critérios Coeficiente de Jaccard (**CJ**) e *Adjusted Rand Index* (**ARI**).

Para as análises estatísticas foi utilizado o procedimento descrito por **Demšar (2006)**, que consiste na aplicação dos testes não-paramétricos de Friedman e de Nemenyi (**Hollander e Wolfe, 1991**) (previamente descritos na Seção 4.2). Todos os experimentos foram executados no mesmo computador Opteron 2GHz com 8Gb de RAM executando apenas o Sistema Operacional em paralelo.

Resumidamente, a avaliação dos algoritmos de **SA** foi realizada considerando os seguintes critérios:

- (i) Acurácia, de acordo com os critérios externos Coeficiente de Jaccard (**CJ**) e *Adjusted Rand Index* (**ARI**) (descritos nesta seção), no agrupamento obtido pelo algoritmo *k-means* utilizando o subconjunto de atributos selecionados.
- (ii) Número de atributos selecionados;
- (iii) Custo computacional envolvido na **SA**.

5.3 Algoritmos e Parâmetros Utilizados

Nestes experimentos foram comparados os algoritmos **SSF** (Seção 3.3.3) e suas variantes não-supervisionadas apresentadas na Seção 3.4, **ACA** (Seção 3.3.2) e **MMP** (Seção 3.3.1). Os parâmetros dos algoritmos foram definidos da seguinte forma:

¹O ajuste referido é uma tentativa de produzir um valor esperado igual a zero nos casos em que a concordância entre as duas partições pode ser dada de forma aleatória (**Milligan, 1996**).

- **SSF** selecionando 1 atributo por grupo:

- $k_{min} = 2$;
- $k_{max} = M - 1$;
- $n_p = 20$.

- **SSF** selecionando 2 atributos por grupo:

- $k_{min} = 2$;
- $k_{max} = M/2$;
- $n_p = 20$.

- **ACA**:

- Parâmetros idênticos ao SSF.

- **MMP**:

- $k_{min} = 1$;
- $k_{max} = M - 1$.

Acredita-se que esses parâmetros sejam suficientes para investigar o comportamento dos algoritmos nas bases de dados utilizadas. Em uma aplicação prática, conhecimento de domínio poderia ser incorporado para a determinação destes parâmetros. Para a discretização dos valores, quando necessária, foi utilizado o algoritmo **PKID** (Yang e Webb, 2001). Os acrônimos utilizados neste capítulo seguem as descrições apresentadas na Seção 4.2.

5.4 Bases de Dados

As seguintes bases de dados, inspiradas nas idéias de Milligan (1980, 1981), foram geradas artificialmente e usadas nos experimentos aqui reportados:

10_250: composta por 250 objetos (50 objetos em cada grupo) e 10 atributos. Os dois primeiros atributos descrevem cinco grupos quando combinados, mas, se considerados individualmente cada um deles permite recuperar três grupos. Os oito atributos restantes representam *ruído*, tendo seus valores derivados de uma distribuição normal que é independente da distribuição dos grupos. Uma parte representativa dos gráficos de dispersão da base de dados é apresentada na Figura 5.1.

12_200: composta por 200 objetos (50 objetos em cada grupo) e doze atributos. O primeiro atributo foi criado utilizando o gerador de dados proposto em Milligan (1981), e descreve quatro grupos. O segundo atributo descreve dois grupos. Os atributos 3, 5, 7, 9 e 11 são

Figura 5.1: Base de dados 10_250.

correlacionados com o primeiro atributo tendo, respectivamente, 10%, 20%, 30%, 40% e 50% de valores ruidosos. Os atributos 4, 6, 8, 10 e 12 são correlacionados com o segundo atributo tendo, respectivamente, 10%, 20%, 30%, 40% e 50% de valores ruidosos. Uma parte representativa dos gráficos de dispersão da base de dados é apresentada na Figura 5.2.

20_250: composta por 250 objetos (50 objetos em cada grupo) e vinte atributos. Os dois primeiros atributos descrevem cinco grupos quando combinados, mas, se considerados individualmente, cada um deles permite recuperar três grupos. Os atributos 3, 5, 7, 9 e 11 são correlacionados com o primeiro atributo tendo, respectivamente, 10%, 20%, 30%, 40% e 50% de valores ruidosos. Os atributos 4, 6, 8, 10 e 12 são correlacionados com o segundo atributo tendo, respectivamente, 10%, 20%, 30%, 40% e 50% de valores ruidosos. Os valores dos atributos 13,14,...,20 são obtidos de uma distribuição de probabilidades uniforme. Uma parte representativa dos gráficos de dispersão da base de dados é apresentada na Figura 5.3.

1000_1000: composta por 1.000 objetos e 1.000 atributos, e foi obtida pelo gerador proposto por **Milligan (1981)**. O primeiro atributo descreve os cinco grupos perfeitamente, enquanto os atributos restantes os fazem se sobrepor. Uma parte representativa dos gráficos de dispersão da base de dados é apresentada na Figura 5.4.

As bases de dados Bio1, Bio2, Bio3, Bio4, Bio5 e Yeast, utilizadas no Capítulo 4 e descritas na Seção 4.4, também foram utilizadas nos experimentos. O uso destas bases de dados justifica-se pelo fato de que os rótulos das classes correspondem a grupos de objetos.

Figura 5.2: Base de dados 12_200.

Figura 5.3: Base de dados 20_250.

Figura 5.4: Base de dados 1000_1000.

Um resumo das características de cada uma das bases acima descritas é apresentado na Tabela 5.2. Nesta tabela, seguindo a notação adotada neste trabalho, N é o número de objetos e M o número de atributos. Todas as bases utilizadas não possuem valores ausentes.

Base	N	M	# grupos (distribuição - %)
Bio1, ..., Bio5	400	20	6 ($\approx$ igualmente distribuídos)
Yeast	205	20	4 (40,5 - 7,3 - 45,4 - 6,8)
10_250	250	10	5 (igualmente distribuídos)
12_200	200	12	4 (igualmente distribuídos)
20_250	250	20	5 (igualmente distribuídos)
1000_1000	1000	1000	5 (igualmente distribuídos)

Tabela 5.2: Sumários das bases de dados de agrupamento utilizadas.

5.5 Resultados e Discussão

Similarmente ao procedimento adotado para avaliar os experimentos envolvendo problemas de classificação, as análises aqui reportadas estão divididas em duas seções. Na primeira seção, o algoritmo **SSF** utilizando duas medidas de correlação lineares — Pearson (ρ) definido na Equação (3.2) e Índice de Compressão Máxima de Informação (**ICMI**) (λ) definido na Equação (3.1)) — é comparado com o algoritmo **MMP**. Na segunda seção, o algoritmo **SSF** utilizando duas medidas de correlação não-lineares — (Medida de Redundância de Interdependência (**MRI**) (R) definido na Equação (3.3) e *Symmetrical Uncertainty* (**SU**) definido na Equação (3.13)) — é comparado com o algoritmo **ACA**. As tabelas contendo todos os resultados estão disponíveis no Apêndice B.

Antes de iniciar as comparações específicas entre os algoritmos de **SA**, é importante tecer alguns comentários gerais sobre o comportamento geral destes nas bases artificiais. Conside-

rando as bases de dados 10_250 e 20_250, **SSF- λ** foi o único algoritmo que não identificou os dois atributos relevantes. Na base de dados 12_200, apenas o **ACA** e o **SSF** (todas as variantes, exceto **SSF- λ**) foram capazes de identificar corretamente os grupos de atributos (dados pelos atributos de “rótulos” pares e ímpares). Nesta base de dados em particular, as diferenças observadas para os valores do critério **ARI** (Tabela B.2) obtidas pelo **ACA-2** e as variantes do **SSF** que selecionam dois atributos podem ser explicadas analisando-se as características da base de dados. Especificamente, o **SSF- ρ -2** seleciona, de cada grupo, seu medóide e seu atributo menos correlacionado com o medóide. Portanto, **SSF- ρ -2** seleciona atributos “ruidosos”, que, pela construção da base de dados, são os atributos menos correlacionados com os medóides. Tais atributos ruidosos tendem naturalmente a deteriorar a qualidade das partições dos dados induzidas pelo *k-means*.

5.5.1 Comparações entre **SSF** e **MMP**

De forma análoga àquela realizada nos experimentos envolvendo o algoritmo **MMP** em problemas de classificação, nos experimentos aqui descritos o algoritmo **MMP** foi executado para cada valor de k do intervalo $[k_{min}, k_{max}]$. Feito isto, o subconjunto de atributos selecionado foi avaliado por meio das partições obtidas pelo algoritmo *k-means*. No entanto, são utilizados nas comparações apenas o melhor caso, denotado por “**MMP** (M)”, e o caso médio, denotado por “**MMP** ($\bar{m}$)” obtidos considerando os critérios **ARI** e **CJ**. É importante ressaltar que esses valores foram obtidos através de uma busca guiada por um algoritmo de agrupamento e, portanto, todo o custo computacional necessário para encontrar o valor deve ser considerado, *i.e.*, a aplicação do Algoritmo 1 e a execução do algoritmo de agrupamento para cada valor de k . As Figuras 5.5 e 5.6 apresentam, para cada valor de k analisado, o valor correspondente dos critérios **ARI** e **CJ** obtidos utilizando-se o subconjunto de atributos selecionados.

Nas Tabelas 5.3 e 5.4 são apresentadas, respectivamente, as comparações de Vitórias/Empates/Derrotas em relação aos valores de **ARI** e **CJ** obtidos após executar o algoritmo *k-means* nos subconjuntos de atributos selecionados. É possível observar que os dois critérios (**ARI** e **CJ**) apresentaram, em praticamente todos os casos, as mesmas indicações em termos de Vitórias/Empates/Derrotas para cada par de algoritmos avaliados. Nestas tabelas também é possível constatar que o algoritmo **SSF- ρ -*** apresentou resultados melhores do que o algoritmo **SSF- λ -***, além de resultados competitivos com o resultado médio obtido pelo **MMP** (**MMP** ($\bar{m}$)). Os melhores resultados foram obtidos pelos melhores subconjuntos de atributos selecionados pelo algoritmo **MMP**.

Os números de Vitórias/Empates/Derrotas para cada par de algoritmos considerando o número de atributos selecionados são apresentados na Tabela 5.5. De forma similar aos resultados obtidos nos experimentos envolvendo problemas de classificação, o algoritmo **SSF** selecionou menos atributos do que o algoritmo **MMP**. No geral, o algoritmo **SSF- ρ -1** selecionou menos atributos do que os demais.

Figura 5.5: Índices externos (ARI e CJ) obtidos nas bases de dados artificiais utilizando o algoritmo MMP para cada valor do parâmetro k .

Utilizando os testes estatísticos de Friedman e de Nemenyi foi possível constatar diferenças estatisticamente significativas ($\alpha=5\%$), considerando os valores de ARI e CJ, apenas entre os algoritmos MMP e SSF- λ -. Considerando o custo computacional envolvido (Tabela B.4), as variantes do SSF que selecionam dois atributos por grupo e a variante SSF- λ -1 foram significativamente ($\alpha=5\%$) mais eficientes do que o MMP. Considerando o número de atributos selecionados por cada algoritmo, os algoritmos SSF- λ -1 e SSF- ρ -1 selecionaram significativamente ($\alpha=10\%$) menos atributos que os demais algoritmos. Considerando-se todos esses aspectos em conjunto, o algoritmo SSF utilizando o coeficiente de correlação de Pearson (SSF- ρ *) é uma boa alternativa ao MMP, já que além de ter apresentado acurácia similar ao MMP, apresentou maior eficiência computacional e selecionou menos atributos.

Como abordado previamente na Seção 4.5.1 os resultados apresentados acima são interessantes sob a perspectiva de aplicações práticas, em especial naquelas em que o usuário desconhece o número de atributos a ser selecionados. No entanto, uma investigação adicional avaliando os dois filtros (SSF e MMP) forçando uma situação em que ambos selecionam subconjuntos de atributos cujas cardinalidades sejam (aproximadamente) iguais, é interessante do

(a) Base de dados Bio1

(b) Base de dados Bio2

(c) Base de dados Bio3

(d) Base de dados Bio4

(e) Base de dados Bio5

(f) Base de dados Yeast

Figura 5.6: Índices externos (**ARI** e **CJ**) obtidos nas bases de Bio1,...,Bio5 e Yeast utilizando o algoritmo **MMP** para cada valor do parâmetro k .

Tabela 5.3: Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando os valores de **ARI** obtido pelos algoritmos **SSF- λ** , **SSF- ρ** e **MMP**.

Algoritmo	MMP (M)	MMP ($\bar{m}$)	SSF- λ -1	SSF- λ -2	SSF- ρ -1	SSF- ρ -2
MMP (M)	—	10/0/0	9/1/0	9/1/0	6/3/1	7/2/1
MMP ($\bar{m}$)	0/0/10	—	7/0/3	6/0/4	3/1/6	4/1/5
SSF- λ -1	0/1/9	3/0/7	—	2/1/7	1/2/7	1/2/7
SSF- λ -2	0/1/9	4/0/6	7/1/2	—	3/1/6	3/1/6
SSF- ρ -1	1/3/6	6/1/3	7/2/1	6/1/3	—	4/6/0
SSF- ρ -2	1/2/7	5/1/4	7/2/1	6/1/3	0/6/4	—

Tabela 5.4: Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando os valores de **CJ** obtido pelos algoritmos **SSF- λ** , **SSF- ρ** e **MMP**.

Algoritmo	MMP (M)	MMP ($\bar{m}$)	SSF- λ -1	SSF- λ -2	SSF- ρ -1	SSF- ρ -2
MMP (M)	—	10/0/0	9/1/0	9/1/0	6/3/1	7/2/1
MMP ($\bar{m}$)	0/0/10	—	7/0/3	6/0/4	3/1/6	4/1/5
SSF- λ -1	0/1/9	3/0/7	—	2/2/6	1/2/7	1/2/7
SSF- λ -2	0/1/9	4/0/6	6/2/2	—	3/1/6	3/1/6
SSF- ρ -1	1/3/6	6/1/3	7/2/1	6/1/3	—	5/5/0
SSF- ρ -2	1/2/7	5/1/4	7/2/1	6/1/3	0/5/5	—

Tabela 5.5: Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando o número de atributos seleccionados pelos algoritmos **SSF- λ** , **SSF- ρ** e **MMP**.

Algoritmo	MMP (M)	SSF- λ -1	SSF- λ -2	SSF- ρ -1	SSF- ρ -2
MMP (M)	—	1/0/9	4/1/5	1/1/8	4/1/5
SSF- λ -1	9/0/1	—	10/0/0	4/2/4	9/0/1
SSF- λ -2	5/1/4	0/0/10	—	1/3/6	4/2/4
SSF- ρ -1	8/1/1	4/2/4	6/3/1	—	10/0/0
SSF- ρ -2	5/1/4	1/0/9	4/2/4	0/0/10	—

ponto de vista acadêmico.

Neste sentido, realizou-se tal investigação, considerando como parâmetro o número de atributos seleccionados pelo **SSF- ρ -1** sendo o M^* . Na sequência, selecciona-se dentre os subconjuntos de atributos seleccionados pelo **MMP** aquele de cardinalidade mais próxima de M^* . As médias e desvios padrão dos resultados obtidos estão dispostos na Tabela 5.6. Utilizando o teste estatístico Wilcoxon *Signed-Rank* foi possível detectar ($\alpha=5\%$) que o **SSF- ρ -1** obteve resultados melhores que o **MMP** independente do critério (**ARI** ou **CJ**) utilizado na avaliação.

Tabela 5.6: Índices externos (**ARI** e **CJ**) obtidos pelo **SSF- ρ -1** e **MMP** seleccionando (aproximadamente) o mesmo número de atributos.

Base	MMP		SSF- ρ -1	
	ARI	CJ	ARI	CJ
10_250	0,49	0,43	1,00	1,00
20_250	0,23	0,24	1,00	1,00
12_200	0,26	0,29	1,00	1,00
Bio1	0,78	0,70	0,53	0,47
Bio2	0,88	0,82	0,91	0,86
Bio3	0,78	0,69	1,00	1,00
Bio4	0,77	0,68	0,82	0,73
Bio5	0,50	0,44	0,54	0,47
Yeast	0,11	0,29	0,75	0,72
1000_1000	0,93	0,88	0,78	0,64

5.5.2 Comparações entre SSF e ACA

Nas Tabelas 5.7 e 5.8 são apresentadas, respectivamente, as comparações de Vitórias/Empates/Derrotas em relação aos valores de **ARI** e **CJ** obtidos executando-se o *k-means* nos subconjuntos de atributos selecionados. É possível verificar que o **ACA** obteve melhores resultados em pelo menos 60% dos casos em relação a cada variação do **SSF**. O uso da medida de correlação **SU** (Equação 3.13) no **SSF** não apresentou diferenças quando comparado ao uso da medida de correlação **MRI** (Equação 3.3).

Os números de Vitórias/Empates/Derrotas entre os pares de algoritmos, considerando-se o número de atributos selecionados, são apresentados na Tabela 5.9. O algoritmo **ACA** selecionou em pelo menos 70% dos casos menos atributos do que os demais algoritmos, independentemente do critério de seleção adotado (um ou dois atributos por grupo). O algoritmo **SSF-SU** selecionou um número menor de atributos em relação ao **SSF-R** em duas base de dados.

Tabela 5.7: Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando os valores de **ARI** obtidos pelos algoritmos **ACA**, **SSF-R** e **SSF-SU**.

Algoritmo	ACA-1	ACA-2	SSF-R-1	SSF-R-2	SSF-SU-1	SSF-SU-2
ACA-1	—	4/1/5	4/2/4	6/2/2	3/3/4	6/2/2
ACA-2	5/1/4	—	4/2/4	7/0/3	4/2/4	6/0/4
SSF-R-1	4/2/4	4/2/4	—	6/0/4	1/8/1	5/0/5
SSF-R-2	2/2/6	3/0/7	4/0/6	—	4/0/6	1/8/1
SSF-SU-1	4/3/3	4/2/4	1/8/1	6/0/4	—	5/1/4
SSF-SU-2	2/2/6	4/0/6	5/0/5	1/8/1	4/1/5	—

Tabela 5.8: Vitórias/Empates/Derrotas para os algoritmos da 1ª coluna considerando os valores de **CJ** obtidos pelos algoritmos **ACA**, **SSF-R** e **SSF-SU**.

Algoritmo	ACA-1	ACA-2	SSF-R-1	SSF-R-2	SSF-SU-1	SSF-SU-2
ACA-1	—	4/1/5	4/2/4	6/1/3	3/3/4	6/1/3
ACA-2	5/1/4	—	4/2/4	7/0/3	4/2/4	6/0/4
SSF-R-1	4/2/4	4/2/4	—	6/0/4	1/8/1	5/0/5
SSF-R-2	3/1/6	3/0/7	4/0/6	—	4/0/6	1/8/1
SSF-SU-1	4/3/3	4/2/4	1/8/1	6/0/4	—	5/1/4
SSF-SU-2	3/1/6	4/0/6	5/0/5	1/8/1	4/1/5	—

Tabela 5.9: Vitórias/Empates/Derrotas para algoritmos da 1ª coluna considerando o número de atributos selecionados pelos algoritmos **ACA**, **SSF-R** e **SSF-SU**.

Algoritmo	ACA-1	ACA-2	SSF-R-1	SSF-R-2	SSF-SU-1	SSF-SU-2
ACA-1	—	10/0/0	8/2/0	10/0/0	7/3/0	10/0/0
ACA-2	0/0/10	—	6/2/2	8/1/1	6/1/3	7/2/1
SSF-R-1	0/2/8	2/2/6	—	10/0/0	0/8/2	9/1/0
SSF-R-2	0/0/10	1/1/8	0/0/10	—	0/0/10	0/8/2
SSF-SU-1	0/3/7	3/1/6	2/8/0	10/0/0	—	10/0/0
SSF-SU-2	0/0/10	1/2/7	0/1/9	2/8/0	0/0/10	—

Seguindo a metodologia adotada neste trabalho, foram aplicados os testes estatísticos de Friedman e Nemenyi. Considerando os critérios **ARI** e **CJ**, não foi encontrada diferença estatisticamente significativa entre os algoritmos. Levando em consideração o número de atributos selecionados, o algoritmo **SSF-R-2** selecionou significativamente mais atributos ($\alpha=10\%$) do

que os algoritmos [ACA-1](#), [ACA-2](#), [SSF-R-1](#) e [SSF-SU-1](#). Além disso, o algoritmo [SSF-SU-2](#) também selecionou significativamente mais atributos do que o [ACA-1](#) e o [SSF-SU-1](#). Em relação ao custo computacional, todos os algoritmos que selecionam dois atributos por grupo foram mais eficientes do que os algoritmos que selecionam um atributo por grupo. Tais resultados sugerem que os algoritmos [SSF](#) e [ACA](#) obtiveram performances similares.

Analogamente ao que foi realizado na Seção 4.5.2, a qualidade dos subconjuntos de atributos selecionados pelos dois algoritmos quando o número de grupos é pré-determinado é também analisada. A motivação dessa comparação é avaliar a qualidade de subconjuntos de atributos de mesma cardinalidade selecionados pelos dois algoritmos. Neste caso, são comparados os algoritmos [SSF-SU-2](#) e [ACA-2](#). Para realizar essa comparação foram realizados dois experimentos. No primeiro, o valor de k (número de grupos) é definido pelo algoritmo [SSF-SU-2](#) e então o [ACA-2](#) realiza o agrupamento de atributos para este valor de número de grupos. Na sequência, os subconjuntos de atributos selecionados são comparados pelos critérios [ARI](#) e [CJ](#) obtidos pela melhor partição induzida pelo k -means. O segundo experimento é análogo, mas, o valor de k é então definido pelo [ACA-2](#). Os resultados obtidos são apresentados nas Tabelas 5.10 e 5.11. Utilizando o teste estatístico Wilcoxon *Signed-Rank* é possível verificar que o [SSF-SU-2](#) obteve resultados estatisticamente piores ($\alpha=10\%$) em ambos os casos. Este resultado sugere que o critério de selecionar o atributo menos correlacionado com o medóide não é o mais indicado para o problema aqui investigado.

Tabela 5.10: Índices externos ([ARI](#) e [CJ](#)) obtidos pelo [SSF-SU-2](#) e [ACA-2](#) utilizando o valor de k definido pelo [SSF-SU-2](#).

Base	SSF-SU-2		ACA-2	
	ARI	CJ	ARI	CJ
10_250	0,25	0,26	1,00	1,00
20_250	1,00	1,00	0,99	0,98
12_200	0,52	0,49	0,99	0,98
Bio1	0,82	0,75	0,80	0,72
Bio2	0,92	0,88	0,78	0,69
Bio3	0,78	0,70	0,80	0,72
Bio4	0,79	0,71	0,92	0,88
Bio5	0,56	0,49	0,79	0,71
Yeast	0,97	0,96	0,96	0,95
1000_1000	0,67	0,51	1,00	1,00

Tabela 5.11: Índices externos ([ARI](#) e [CJ](#)) obtidos pelo [SSF-SU-2](#) e [ACA-2](#) utilizando o valor de k definido pelo [ACA-2](#).

Base	SSF-SU-2		ACA-2	
	ARI	CJ	ARI	CJ
10_250	0,44	0,39	1,00	1,00
20_250	1,00	1,00	0,99	0,98
12_200	0,52	0,49	0,99	0,98
Bio1	0,64	0,56	1,00	1,00
Bio2	0,68	0,59	0,54	0,47
Bio3	0,95	0,92	0,81	0,73
Bio4	0,79	0,71	0,92	0,88
Bio5	0,53	0,46	0,54	0,47
Yeast	0,51	0,51	0,71	0,69
1000_1000	0,67	0,51	1,00	1,00

Conclusões e Trabalhos Futuros

Aplicações de Mineração de Dados usualmente envolvem dados de alta dimensionalidade. O uso de algoritmos de Seleção de Atributos (SA) torna-se de grande importância para o sucesso de tais aplicações. A alta dimensionalidade pode não apenas dificultar o processo de identificação de padrões nos dados, como também pode torná-lo proibitivo computacionalmente. Por tal razão, a tarefa de SA vem recebendo, a partir da década de 90, grande atenção da comunidade científica, e foi também objeto de estudo desta dissertação, cujas principais contribuições são abordadas na Seção 6.1. A Seção 6.2, por sua vez, apresenta as principais conclusões derivadas deste trabalho. Finalmente, na Seção 6.3, algumas sugestões de possíveis trabalhos futuros são descritas.

6.1 Principais Contribuições

Neste trabalho foram propostos e estudados algoritmos de SA que se baseiam na identificação de grupos de atributos correlacionados. O conceito de agrupamento de atributos foi formalizado, introduzindo-se também uma fundamentação teórica para sua aplicação em problemas de classificação. Com o objetivo de aperfeiçoar filtros baseados no agrupamento de atributos descritos na literatura – Algoritmo de Agrupamento de Atributos (ACA) proposto por Au et al. (2005) e o Algoritmo de Mitra, Murthy e Pal (MMP) (Mitra et al., 2002) — foi proposto o Filtro Silhueta Simplificada (SSF). Para o SSF, foram propostas variantes a fim de suprir algumas limitações encontradas nos outros algoritmos do gênero, bem como para incorporar a informação da classe ao processo de agrupamento de atributos. As análises de complexidade assintótica de todos os algoritmos estudados foram desenvolvidas, sendo também uma contribuição deste trabalho. Outras contribuições deste trabalho envolvem a avaliação dos algoritmos estudados em problemas de classificação e de agrupamento de dados, assim como a

aplicação do filtro proposto em dois estudos de casos de aplicações reais.

Por fim, vale mencionar que uma parte significativa dos resultados apresentados nesta dissertação foram publicados em anais de conferências nacionais e internacionais (Covões et al., 2008, 2009; Covões e Hruschka, 2009).

6.2 Conclusões

Os resultados dos experimentos executados em problemas de classificação indicam que o algoritmo SSF, selecionando o atributo medóide de cada grupo e o atributo menos correlacionado com o medóide, é uma boa alternativa ao algoritmo MMP (Mitra et al., 2002), por dois motivos: (i) o SSF é mais eficiente computacionalmente; (ii) não foram identificadas diferenças significativas na qualidade dos subconjuntos selecionados por cada algoritmo ao se considerar as taxas de erro obtidas pelos classificadores *Naïve Bayes* e *k-Nearest Neighbors*. Embora os testes estatísticos não tenham indicado diferenças estatisticamente significantes entre o SSF e o ACA (Au et al., 2005), o SSF, utilizando a Medida de Redundância de Interdependência também utilizada em (Au et al., 2005) como medida de correlação, foi superior ao ACA em aproximadamente 70% das bases de dados analisadas. Isso demonstra que o SSF oferece uma excelente alternativa de uso prático em relação ao ACA, principalmente pelo fato do ACA não possuir um critério de seleção de atributos explícito e automático a ser executado após o agrupamento. As modificações propostas ao SSF para que a correlação atributo-classe fosse incorporada ao processo de agrupamento originaram melhores resultados, sem afetar significativamente o custo computacional. A alternativa de selecionar o atributo com maior Informação Mútua Condicionada Normalizada (IMCN) em relação ao medóide, por sua vez, não trouxe melhorias nos resultados quando comparada à seleção do atributo menos correlacionado com o medóide.

Adicionalmente aos experimentos envolvendo problemas de classificação em bases de dados de domínio público, o algoritmo SSF e as variantes propostas foram avaliados em dois estudos de casos de aplicações reais. O primeiro estudo de caso envolve a classificação de grãos de café, no qual os filtros propostos apresentaram resultados competitivos quando comparados com alguns filtros supervisionados descritos na literatura. O segundo estudo de caso envolve uma aplicação de mineração de textos, no qual o SSF e suas variantes apresentaram não apenas uma excelente relação de custo/benefício em relação ao uso de *wrappers*, mas também demonstraram que em uma etapa de pós-processamento os algoritmos de SA baseados no agrupamento de atributos podem auxiliar especialistas de domínio na inclusão/exclusão de atributos.

Finalmente, os experimentos envolvendo problemas de agrupamento de dados indicam que o SSF utilizando o coeficiente de correlação de Pearson apresenta resultados competitivos quando comparado ao MMP. Porém, quando se compara o ACA com o SSF, este apresentou resultados muito similares ou piores, principalmente quando dois atributos são selecionados. Isso sugere que a seleção do atributo menos correlacionado com o medóide pode não ser a melhor estratégia em problemas de agrupamento.

6.3 Trabalhos Futuros

Apesar do algoritmo [SSF](#) estimar o número de grupos de atributos de forma automática, essas estimativas são obtidas através de uma estratégia de amostragem. Neste sentido, o uso de algoritmos de agrupamento que sejam capazes de estimar automaticamente o valor de k de forma mais eficiente pode melhorar significativamente o algoritmo, e merece ser estudado. Uma investigação mais profunda sobre o desempenho dos algoritmos estudados em problemas de agrupamento de dados, considerando mais bases de dados com características diversificadas, também pode permitir conclusões mais interessantes, se constituindo em um trabalho futuro importante. O uso de diferentes medidas de correlação no processo de agrupamento é um trabalho futuro promissor, bem como o uso de algoritmos de agrupamento que permitam induzir grupos com sobreposição. Por fim, a definição de novos critérios para a seleção dos atributos a partir dos grupos induzidos, principalmente para problemas de agrupamento, merece atenção.

Referências Bibliográficas

- Aha, D. W., Kibler, D. F., e Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6:37–66. Citado nas páginas 4 e 32.
- Alon, U., Barkai, N., Notterman, D. A., Gishdagger, K., Ybarradagger, S., Mackdagger, D., e Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 96(12):6745–6750. Citado na página 36.
- Arabie, P. e Hubert, L. J. (1999). *An Overview of Combinatorial Data Analysis*, capítulo 1, páginas 5–64. World Scientific Publishing Company. Citado na página 3.
- Asuncion, A. e Newman, D. (2007). UCI machine learning repository. <http://www.ics.uci.edu/~mllearn/MLRepository.html>. Citado na página 36.
- Au, W.-H., Chan, K. C. C., Wong, A. K. C., e Wang, Y. (2005). Attribute clustering for grouping, selection, and classification of gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(2):83–101. Citado nas páginas 13, 15, 17, 21, 22, 49, 69, e 70.
- Basak, J., De, R. K., e Pal, S. K. (1998). Unsupervised feature selection using a neuro-fuzzy approach. *Pattern Recognition Letters*, 19(11):997 – 1006. Citado na página 12.
- Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418. Citado na página 4.
- Berry, M. (2003). *Survey of Text Mining : Clustering, Classification, and Retrieval*. Springer. Citado na página 4.
- Bishop, C. M. (2007). *Pattern Recognition and Machine Learning*. Springer, 1 edition. Citado na página 5.

- Bonev, B., Escolano, F., e Cazorla, M. (2008). Feature selection, mutual information, and the classification of high-dimensional patterns. *Pattern Analysis & Applications*, 11(3):309–319. Citado na página 13.
- Butterworth, R., Piatetsky-Shapiro, G., e Simovici, D. A. (2005). On feature selection through clustering. Em *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'2005)*, páginas 581–584, Washington, DC, USA. IEEE Computer Society. Citado na página 12.
- Carlson, A., Betteridge, J., Jr., E. R. H., e Mitchell, T. M. (2009). Coupling semi-supervised learning of categories and relations. Em *Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing*. Citado na página 52.
- Chien, B. e Yang, J. (2006). Features selection based on rough membership and genetic programming. Em *IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC '06.*, volume 5, páginas 4124–4129. Citado na página 13.
- Chow, T. e Huang, D. (2005). Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information. *IEEE Transactions on Neural Networks*, 16(1):213–224. Citado na página 13.
- Chow, T. W. S., Wang, P., e Ma, E. W. M. (2008). A new feature selection scheme using a data distribution factor for unsupervised nominal data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B.*, 38(2):499–509. PMID: 18348931. Citado na página 12.
- Cover, T. M. e Thomas, J. A. (1991). *Elements of Information Theory*. Wiley-Interscience. Citado na página 30.
- Covões, T. F. e Hruschka, E. R. (2009). An experimental study on unsupervised clustering-based feature selection methods. Em *9th International Conference on Intelligent Systems Design and Applications (ISDA'09)*, páginas 993–1000. IEEE Press. Citado nas páginas 55 e 70.
- Covões, T. F., Hruschka, E. R., de Castro, L. N., e de Carvalho, A. C. P. L. F. (2008). Seleção de atributos para classificação de grãos. Em *IV Workshop em Algoritmos e Aplicações de Mineração de Dados - WAAMD'2008 (em conjunto com o XXIII Simpósio Brasileiro de Banco de Dados - SBBD'2008)*, páginas 64–69. Citado nas páginas 49 e 70.
- Covões, T. F., Hruschka, E. R., de Castro, L. N., e Átila Menezes dos Santos (2009). A cluster-based feature selection approach. Em *Hybrid Artificial Intelligence Systems, 4th International Conference (HAIS'2009)*, volume 5572 of *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, páginas 169–176. Citado nas páginas 6, 15, 17, 23, 25, 28, e 70.

- Dash, M. e Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, 1:131–156. Citado na página 13.
- Dash, M. e Liu, H. (2000). Feature selection for clustering. Em *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications (PADKK'2000)*, páginas 110–121. Springer-Verlag. Citado na página 12.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30. Citado nas páginas 33, 34, 42, e 57.
- Devaney, M. e Ram, A. (1997). Efficient feature selection in conceptual clustering. Em *Proceedings of the 14th International Conference on Machine Learning (ICML'1997)*, páginas 92–97. Morgan Kaufmann. Citado na página 12.
- Diamantini, C. e Potena, D. (2007). A study of feature extraction techniques based on decision border estimate. Em *Computational Methods for Feature Selection*, capítulo 6, páginas 109–130. Chapman & Hall/CRC. Citado na página 9.
- Draminski, M., Rada-Iglesias, A., Enroth, S., Wadelius, C., Koronacki, J., e Komorowski, J. (2008). Monte carlo feature selection for supervised classification. *Bioinformatics*, 24(1):110–117. Citado na página 13.
- Dunn, J. C. (1974). Well separated clusters and optimal fuzzy-partitions. *Journal of Cybernetics*, 4:95–104. Citado na página 4.
- Dy, J. G. (2007). Unsupervised feature selection. Em *Computational Methods of Feature Selection*, capítulo 2, páginas 19–39. Chapman & Hall/CRC. Citado na página 5.
- Dy, J. G. e Brodley, C. E. (2000). Visualization and interactive feature selection for unsupervised data. Em *Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'2000)*, páginas 360–364, New York, NY, USA. ACM. Citado na página 12.
- Dy, J. G. e Brodley, C. E. (2004). Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889. Citado na página 5.
- Elder, J. F. e Pregibon, D. (1996). A statistical perspective on knowledge discovery in databases. Em *Advances in Knowledge Discovery and Data Mining*, páginas 83–113. AAAI/MIT Press. Citado na página 4.
- Everitt, B. S. (2001). *Cluster Analysis*. Edward Arnold and Halsted Press. Citado nas páginas 3, 4, e 26.

- Fayyad, U. M., Djorgovski, S. G., e Weir, N. (1996a). Automating the analysis and cataloging of sky surveys. Em *Advances in Knowledge Discovery and Data Mining*, páginas 471–493. AAAI/MIT Press. Citado na página [1](#).
- Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., e Uthurusamy, R., editores (1996b). *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press. Citado nas páginas [1](#), [2](#), e [3](#).
- Fleuret, F. (2004). Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555. Citado na página [13](#).
- Giraldi, G., Rodrigues, P., Kitani, E., Sato, J., e Thomaz, C. (2008). Statistical learning approaches for discriminant features selection. *Journal of the Brazilian Computer Society*, 14(2):7–22. Citado na página [13](#).
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. Citado na página [11](#).
- Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., e Lander, E. S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537. Citado na página [36](#).
- Guyon, I. e Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182. Citado nas páginas [9](#) e [11](#).
- Guyon, I., Gunn, S., Nikraves, M., e Zadeh, L. A. (2006). *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. Citado na página [12](#).
- Guyon, I., Weston, J., Barnhill, S., e Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422. Citado na página [13](#).
- Haindl, M., Somol, P., Ververidis, D., e Kotropoulos, C. (2006). Feature selection based on mutual correlation. Em *Progress in Pattern Recognition, Image Analysis and Applications, 11th Iberoamerican Congress in Pattern Recognition, CIARP*, páginas 569–577. Citado na página [12](#).
- Halkidi, M., Batistakis, Y., e Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems*, 17:107–145. Citado na página [4](#).
- Hall, M. A. (1999). *Correlation-based Feature Selection for Machine Learning*. Tese de Doutorado, University of Waikato. Citado nas páginas [10](#), [13](#), [27](#), e [49](#).
- Han, J. e Kamber, M. (2000). *Data Mining: Concepts and Techniques*. Morgan Kaufmann. Citado nas páginas [1](#), [4](#), e [10](#).

- Hand, D. J., Mannila, H., e Smyth, P. (2001). *Principles of Data Mining (Adaptive Computation and Machine Learning)*. The MIT Press. Citado nas páginas 2 e 5.
- Handl, J. e Knowles, J. (2006). Feature subset selection in unsupervised learning via multiobjective optimization. *International Journal of Computational Intelligence Research*, 2(3):217–238. Citado na página 12.
- Hartigan, J. A. e Wong, M. A. (1979). A k-means clustering algorithm. *Applied Statistics*, 28:100–108. Citado nas páginas 22 e 55.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA. Citado na página 11.
- Hollander, M. e Wolfe, D. A. (1991). *Nonparametric statistical methods (2nd Edition)*. Wiley, New York. Citado nas páginas 33, 42, e 57.
- Hong, Y., Kwong, S., Chang, Y., e Ren, Q. (2008). Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm. *Pattern Recognition*, 41(9):2742 – 2756. Citado nas páginas 5 e 12.
- Hruschka, E. R. (2001). *Algoritmos Genéticos de Agrupamento para Extração de Regras de Redes Neurais*. Tese de Doutorado, Universidade Federal do Rio de Janeiro. Citado na página 3.
- Hruschka, E. R., Campello, R. J., e de Castro, L. N. (2006a). Evolving clusters in gene-expression data. *Information Sciences*, 176(13):1898 – 1927. Citado nas páginas 24 e 26.
- Hruschka, E. R. e Covões, T. F. (2005). Feature selection for cluster analysis: an approach based on the simplified silhouette criterion. *International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA'2005)*, 1:32–38. Citado nas páginas 5, 6, e 12.
- Hruschka, E. R., Covões, T. F., Jr., E. R. H., e Ebecken, N. F. F. (2005). Feature selection for clustering problems: a hybrid algorithm that iterates between k-means and a bayesian filter. Em *Proceedings of the 5th International Conference on Hybrid Intelligent Systems (HIS'2005)*, páginas 405–410. Citado nas páginas 12 e 13.
- Hruschka, E. R., Jr., E. R. H., Covões, T. F., e Ebecken, N. F. F. (2006b). Bayesian feature selection for clustering problems. *Journal of Information & Knowledge Management (JIKM)*, 5:315–327. Citado nas páginas 13 e 23.
- Hubert, L. e Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218. Citado na página 57.

- Jaccard, P. (1908). Nouvelles recherches sur la distribution florale. *Bulletin de la Société Vaudoise de Sciences Naturelles*, 44:223–370. Citado na página 57.
- Jain, A. K. e Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall. Citado nas páginas 4 e 56.
- Jakulin, A. (2005). *Machine Learning Based on Attribute Interactions*. Tese de Doutorado, University of Ljubljana. Citado na página 29.
- Kantardzic, M. e Zurada, J. (2005). *Next Generation of Data-Mining Applications*. Wiley-IEEE Press. Citado na página 1.
- Kaufman, L. e Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley. Citado nas páginas 3, 4, e 26.
- Kim, Y., Street, W. N., e Menczer, F. (2000). Feature selection in unsupervised learning via evolutionary search. Em *Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'2000)*, páginas 365–369, New York, NY, USA. ACM. Citado na página 12.
- Kohavi, R. e John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324. Citado nas páginas 12, 13, 16, 49, e 52.
- Koller, D. e Sahami, M. (1996). Toward optimal feature selection. Em *Proceedings of the 13th International Conference on Machine Learning (ICML'1996)*, páginas 284–292. Morgan Kaufmann. Citado na página 16.
- Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89 – 109. Citado na página 5.
- Krupka, E., Navot, A., e Tishby, N. (2008). Learning to select features using their properties. *Journal of Machine Learning Research*, 9:2349–2376. Citado na página 13.
- Langley, P. (1994). Selection of relevant features in machine learning. Em *Proceedings of the AAAI Fall symposium on relevance*, páginas 140–144. AAAI Press. Citado na página 10.
- Lashkia, G. V. e Anthony, L. (2004). Relevant, irredundant feature selection and noisy example elimination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(2):888–897. Citado na página 13.
- Law, M. H. C., Figueiredo, M. A. T., e Jain, A. K. (2004). Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1154–1166. Citado na página 12.

- Lee, H. D. (2005). *Seleção de atributos importantes para a extração de conhecimento de base de dados*. Tese de Doutorado, Universidade de São Paulo, <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-22022006-172219/>. Citado nas páginas 13, 32, 33, e 38.
- Letrémy, P., Cottrell, M., Esposito, E., Laffite, V., e Showk, S. (2007). Consumer profile identification and allocation. Em Sandoval, F., Prieto, A., Cabestany, J., e Grana, M., editores, *Computational and Ambient Intelligence - 9th International Work-Conference on Artificial Neural Networks (IWANN-2007)*, Lecture Notes in Computer Science - 4507, páginas 530–538. Springer. Citado na página 4.
- Liang, J., Yang, S., e Winstanley, A. (2008). Invariant optimal feature selection: A distance discriminant and feature ranking based solution. *Pattern Recognition*, 41(5):1429–1439. Citado na página 13.
- Liu, C. L. (1968). *Introduction to Combinatorial Mathematics (Computer Science Series)*. Mcgraw-Hill College. Citado na página 3.
- Liu, H. e Motoda, H. (1998). *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers. Citado nas páginas 11, 12, e 33.
- Liu, H. e Motoda, H. (2007). *Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series)*. Chapman & Hall/CRC. Citado na página 12.
- Liu, H. e Setiono, R. (1996). A probabilistic approach to feature selection - a filter solution. Em *Proceedings of the 13th International Conference on Machine Learning (ICML'1996)*, páginas 319–327. Citado nas páginas 13 e 49.
- Liu, H., Sun, J., Liu, L., e Zhang, H. (2009). Feature selection with dynamic mutual information. *Pattern Recognition*, 42(7):1330 – 1339. Citado na página 13.
- Liu, H. e Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502. Citado na página 12.
- Liu, Y. e Zheng, Y. F. (2006). FS-SFS: A novel feature selection method for support vector machines. *Pattern Recognition*, 39(7):1333 – 1345. Citado na página 13.
- Maldonado, S. e Weber, R. (2009). A wrapper method for feature selection using support vector machines. *Information Sciences*, 179(13):2208 – 2217. Special Section on High Order Fuzzy Sets. Citado na página 13.

- Mao, K. (2005). Identifying critical variables of principal components for unsupervised feature selection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 35(2):339–344. Citado na página 5.
- Milligan, G. (1980). An examination of the effect of six types of error perturbation on fifteen clustering algorithms. *Psychometrika*, 45(3):325–342. Citado na página 58.
- Milligan, G. (1981). A monte carlo study of thirty internal criterion measures for cluster analysis. *Psychometrika*, 46(2):187–199. Citado nas páginas 58 e 59.
- Milligan, G. (1996). Clustering validation: Results and implications for applied analyses. Em *Clustering and Classification*, páginas 341–373. World Scientific Publishing Company. Citado nas páginas 3 e 57.
- Mitra, P., Murthy, C. A., e Pal, S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):301–312. Citado nas páginas 13, 15, 17, 19, 20, 21, 37, 49, 69, e 70.
- Ng, W. W., Yeung, D. S., Firth, M., Tsang, E. C., e Wang, X.-Z. (2008). Feature selection using localized generalization error for supervised classification problems using rbfn. *Pattern Recognition*, 41(12):3706 – 3719. Citado na página 13.
- Ouardighi, A. E., Akadi, A. E., e Aboutajdine, D. (2007). Feature selection on supervised classification using wilks lambda statistic. Em *International Symposium on Computational Intelligence and Intelligent Informatics (ISCIH'2007)*, páginas 51–55. Citado na página 13.
- Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572. Citado na página 9.
- Peng, H., Long, F., e Ding, C. (2005). Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1226–1238. Citado na página 13.
- Peña, J. M., Lozano, J. A., Larrañaga, P., e Inza, I. (2001). Dimensionality reduction in unsupervised learning of conditional gaussian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):590–603. Citado na página 12.
- Piatetsky-Shapiro, G. e Frawley, W. J., editores (1991). *Knowledge Discovery in Databases*. AAAI/MIT Press. Citado na página 1.
- Press, W. H., Flannery, B. P., Teukolsky, S. A., e Vetterling, W. T. (1990). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge. Citado na página 27.

- Questier, F., Arnaut-Rollier, I., Walczak, B., e Massart, D. L. (2002). Application of rough set theory to feature selection for unsupervised clustering. *Chemometrics and Intelligent Laboratory Systems*, 63(2):155 – 167. Citado na página 12.
- Questier, F., Put, R., Coomans, D., Walczak, B., e Heyden, Y. V. (2005). The use of CART and multivariate regression trees for supervised and unsupervised feature selection. *Chemometrics and Intelligent Laboratory Systems*, 76(1):45 – 54. Citado na página 12.
- Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. Citado nas páginas 4, 11, 12, e 27.
- Raftery, A. E. e Dean, N. (2006). Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473):168–178. Citado na página 12.
- Reunanen, J., Guyon, I., e Elisseeff, A. (2003). Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3:1371–1382. Citado na página 32.
- Russell, S. e Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2 edition. Citado na página 11.
- Sebban, M. e Nock, R. (2002). A hybrid filter/wrapper approach of feature selection using information theory. *Pattern Recognition*, 35(4):835 – 846. Citado na página 13.
- Sevon, P., Toivonen, H., e Onkamo, P. (2005). Gene mapping by pattern discovery. Em *Data Mining in Bioinformatics*, páginas 105–126. Springer. Citado na página 1.
- Shannon, C. E. (1948). The mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656. Citado na página 27.
- Singhi, S. K. e Liu, H. (2006). Feature subset selection bias for classification learning. Em *Proceedings of the 23rd International Conference on Machine learning (ICML'2006)*, páginas 849–856, New York, NY, USA. ACM. Citado na página 32.
- Song, L., Smola, A., Gretton, A., Borgwardt, K. M., e Bedo, J. (2007). Supervised feature selection via dependence estimation. Em *Proceedings of the 24th International Conference on Machine learning (ICML'2007)*, páginas 823–830, New York, NY, USA. ACM. Citado na página 13.
- Steinley, D. (2004). Properties of the Hubert-Arabie Adjusted Rand Index. *Psychological methods*, 9(3):386–396. Citado na página 57.
- Tahir, M. A., Bouridane, A., e Kurugollu, F. (2007). Simultaneous feature selection and feature weighting using hybrid tabu search/k-nearest neighbor classifier. *Pattern Recognition Letters*, 28(4):438 – 446. Citado na página 13.

- Talavera, L. (2000). Dependency-based feature selection for clustering symbolic data. *Intelligent Data Analysis*, 4(1):19–28. Citado na página [12](#).
- Talavera, L. (2005). An evaluation of filter and wrapper methods for feature selection in categorical clustering. Em Famili, A. F., Kok, J. N., Peña, J. M., Siebes, A., e Feelders, A. J., editores, *Advances in Intelligent Data Analysis VI, 6th International Symposium on Intelligent Data Analysis*, volume 3646 of *Lecture Notes in Computer Science*, páginas 440–451. Springer. Citado na página [12](#).
- Tan, P.-N., Steinbach, M., e Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wesley. Citado na página [3](#).
- Theodoridis, S. e Koutroumbas, K. (2006). *Pattern Recognition*. Academic Press. Citado na página [3](#).
- Traina Jr., C., Traina, A. J. M., Wu, L., e Faloutsos, C. (2000). Fast feature selection using the fractal dimension. Em *XV Simpósio Brasileiro de Banco de Dados (SBBD'2000)*, páginas 158–171. Citado na página [12](#).
- Vellido, A. (2006). Assessment of an unsupervised feature selection method for generative topographic mapping. Em Kollias, S. D., Stafylopatis, A., Duch, W., e Oja, E., editores, *ICANN (2)*, volume 4132 of *Lecture Notes in Computer Science*, páginas 361–370. Springer. Citado nas páginas [5](#) e [12](#).
- Vendramin, L., Campello, R. J. G. B., e Hruschka, E. R. (2008). A robust methodology for comparing performances of clustering validity criteria. Em *Simpósio Brasileiro de Inteligência Artificial (SBIA)*, Lecture Notes in Artificial Intelligence (LNCS/LNAI), Salvador. Citado na página [4](#).
- Vendramin, L., Campello, R. J. G. B., e Hruschka, E. R. (2009). On the comparison of relative clustering validity criteria. Em *Proceedings of the SIAM International Conference on Data Mining (SDM'2009)*, páginas 733–744. SIAM. Citado na página [4](#).
- Witten, I. H. e Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann. Citado nas páginas [4](#), [5](#), e [32](#).
- Wu, X., Kumar, V., Ross, Ghosh, J., Yang, Q., Motoda, H., Mclachlan, G., Ng, A., Liu, B., Yu, P., Zhou, Z.-H., Steinbach, M., Hand, D., e Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37. Citado nas páginas [22](#), [32](#), e [55](#).
- Xing, E. P. e Karp, R. M. (2001). CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, 17:306–315. Citado na página [13](#).

- Yang, J. e Honavar, V. (1998). Feature subset selection using a genetic algorithm. Em *IEEE Intelligent Systems*, páginas 380–385. Citado na página 13.
- Yang, Y. e Webb, G. (2001). Proportional k-interval discretization for naive-bayes classifiers. Em *Proceedings of the 12th European Conference on Machine Learning (ECML'2001)*, páginas 564–575. Springer-Verlag. Citado nas páginas 35 e 58.
- Yeung, K., Medvedovic, M., e Bumgarner, R. (2003). Clustering gene-expression data with repeated measurements. *Genome Biology*, 4(5):R34. Citado nas páginas 4, 35, e 36.
- Yu, L., Ding, C., e Loscalzo, S. (2008). Stable feature selection via dense feature groups. Em *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'2008)*, páginas 803–811, New York, NY, USA. ACM. Citado na página 12.
- Yu, L. e Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. Em *Proceedings of the 20th International Conference on Machine Learning (ICML'2003)*, Washington DC, USA. Citado nas páginas 13, 27, e 49.
- Yu, L. e Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224. Citado nas páginas 16 e 17.
- Zeng, H. e Cheung, Y.-M. (2009). A new feature selection method for gaussian mixture clustering. *Pattern Recognition*, 42(2):243 – 250. Learning Semantics from Multimedia Content. Citado na página 13.
- Zhang, L., Zhu, J., e Yao, T. (2004). An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(4):243–269. Citado na página 5.
- Zhang, T., Ramakrishnan, R., e Livny, M. (1997). BIRCH: a new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, 1(2):141–182. Citado na página 24.

Tabelas dos Resultados dos Experimentos de Classificação

Este apêndice contém as tabelas utilizadas na avaliação dos experimentos de [SA](#) em problemas de classificação. As Tabelas [A.1-A.12](#) apresentam, para cada base analisada, as médias e desvios padrão (entre parênteses) do número de atributos selecionados (M^*), a taxa de erro obtida pelo classificador [KNN](#) utilizando os subconjuntos de atributos selecionados ([%-KNN](#)) e a taxa de erro obtida pelo classificador [NB](#) utilizando os subconjuntos de atributos selecionados ([%-NB](#)). A Tabela [A.13](#) apresenta os tempos computacionais (em segundos) despendidos por cada algoritmo analisado. As Tabelas [A.14](#) e [A.15](#) apresentam a categorização dos resultados obtidos por cada algoritmo de [SA](#) em cada base de dados analisada, de acordo com a avaliação multicritério adotada neste trabalho.

Tabela A.1: Resultados obtidos — Bio1.

Algoritmo	M^*	%-KNN	%-NB
SSF- λ -1	9,5 (1,1)	0,00 (0,00)	0,00 (0,00)
SSF- λ -2	17,2 (4,1)	0,00 (0,00)	0,00 (0,00)
SSF- ρ -1	6,0 (0,0)	0,00 (0,00)	0,00 (0,00)
SSF- ρ -2	12,0 (0,0)	0,00 (0,00)	0,00 (0,00)
SSF-R-1	3,9 (3,1)	0,00 (0,00)	0,00 (0,00)
SSF-R-2	4,4 (2,8)	0,00 (0,00)	0,00 (0,00)
ACA-1	2,0 (0,0)	0,25 (0,75)	0,75 (1,14)
ACA-2	3,0 (0,0)	0,25 (0,75)	0,25 (0,75)
SSF-SU-1	2,0 (0,0)	0,00 (0,00)	0,75 (1,14)
SSF-SU-2	3,0 (0,0)	0,00 (0,00)	0,25 (0,75)
SSF-SU _S -1	9,5 (0,9)	0,00 (0,00)	0,00 (0,00)
SSF-SU _S -2	17,4 (1,1)	0,00 (0,00)	0,00 (0,00)
SSF-SU _S -KS-1	9,5 (0,6)	0,00 (0,00)	0,00 (0,00)
SSF-SU _S -KS-2	18,5 (1,0)	0,00 (0,00)	0,00 (0,00)
SSF-SU _S -I	17,4 (1,1)	0,00 (0,00)	0,00 (0,00)
SSF-SU _S -KS-I	18,5 (1,0)	0,00 (0,00)	0,00 (0,00)
MMP (M)	2,7 (0,4)	0,00 (0,00)	—
MMP (M)	2,5 (0,5)	—	0,00 (0,00)
MMP ($\bar{m}$)	—	1,49 (0,41)	1,09 (0,45)
MMP (P)	1,1 (0,3)	15,00 (4,47)	—
MMP (P)	1,3 (0,4)	—	8,75 (2,56)
Todos	20	0,00 (0,00)	0,00 (0,00)

Tabela A.2: Resultados obtidos — Bio2.

Algoritmo	M^*	%-KNN	%-NB
SSF- λ -1	9,8 (0,4)	6,00 (3,00)	5,50 (3,67)
SSF- λ -1	9,8 (0,4)	6,00 (3,00)	5,50 (3,67)
SSF- λ -2	19,6 (0,8)	1,00 (1,66)	1,25 (1,25)
SSF- ρ -1	3,6 (0,4)	3,75 (2,30)	4,00 (3,00)
SSF- ρ -2	7,2 (0,9)	4,00 (2,29)	2,75 (2,08)
SSF-R-1	2,3 (0,9)	3,00 (1,87)	2,50 (1,94)
SSF-R-2	3,6 (1,8)	3,00 (1,87)	2,75 (2,08)
ACA-1	2,0 (0,0)	3,00 (1,87)	2,25 (2,08)
ACA-2	3,0 (0,0)	1,75 (1,14)	1,50 (2,00)
SSF-SU-1	2,0 (0,0)	3,00 (1,87)	2,25 (2,08)
SSF-SU-2	3,0 (0,0)	2,75 (2,08)	2,50 (2,24)
SSF-SU _S -1	8,8 (0,6)	3,25 (2,25)	2,25 (2,08)
SSF-SU _S -2	17,0 (1,0)	1,25 (2,30)	1,25 (1,68)
SSF-SU _S -KS-1	9,4 (0,4)	2,25 (2,61)	2,50 (2,24)
SSF-SU _S -KS-2	17,7 (0,6)	1,50 (1,66)	0,75 (1,14)
SSF-SU _S -I	17,0 (1,0)	1,00 (2,29)	1,00 (1,22)
SSF-SU _S -KS-I	17,7 (0,6)	1,50 (1,66)	0,75 (1,14)
MMP (M)	3,1 (1,0)	0,25 (0,75)	—
MMP (M)	3,7 (2,1)	—	0,00 (0,00)
MMP ($\bar{m}$)	—	5,72 (1,46)	4,90 (1,18)
MMP (P)	1,0 (0,0)	39,75 (5,86)	37,00 (6,60)
Todos	20	1,00 (1,66)	1,25 (1,25)

Tabela A.3: Resultados obtidos — Bio3.

Algoritmo	M^*	%-KNN	%-NB
SSF- λ -1	8,6 (0,4)	0,00 (0,00)	0,00 (0,00)
SSF- λ -2	17,2 (0,9)	0,00 (0,00)	0,00 (0,00)
SSF- ρ -1	7,7 (0,9)	0,00 (0,00)	0,00 (0,00)
SSF- ρ -2	15,4 (1,8)	0,00 (0,00)	0,00 (0,00)
SSF-R-1	8,5 (1,1)	0,00 (0,00)	0,00 (0,00)
SSF-R-2	12,9 (1,3)	0,00 (0,00)	0,00 (0,00)
ACA-1	2,0 (0,0)	2,75 (2,61)	2,25 (2,61)
ACA-2	3,0 (0,0)	0,50 (1,00)	1,00 (1,22)
SSF-SU-1	4,7 (3,3)	2,00 (2,69)	1,75 (2,51)
SSF-SU-2	7,3 (5,3)	1,50 (2,55)	1,75 (2,25)
SSF-SU _S -1	8,8 (1,3)	0,00 (0,00)	0,00 (0,00)
SSF-SU _S -2	16,4 (1,1)	0,00 (0,00)	0,00 (0,00)
SSF-SU _S -KS-1	9,5 (1,3)	0,00 (0,00)	0,00 (0,00)
SSF-SU _S -KS-2	16,6 (1,0)	0,00 (0,00)	0,00 (0,00)
SSF-SU _S -I	16,4 (1,1)	0,00 (0,00)	0,00 (0,00)
SSF-SU _S -KS-I	16,6 (1,0)	0,00 (0,00)	0,00 (0,00)
MMP (M)	2,2 (0,4)	0,00 (0,00)	—
MMP (M)	2,2 (0,4)	—	0,00 (0,00)
MMP ($\bar{m}$)	—	2,85 (0,66)	2,25 (0,47)
MMP (P)	1,0 (0,0)	33,5 (6,24)	28,5 (3,91)
Todos	20	0,00 (0,00)	0,00 (0,00)

Tabela A.4: Resultados obtidos — Bio4.

Algoritmo	M^*	%-KNN	%-NB
SSF- λ -1	5,4 (2,1)	5,50 (3,50)	5,25 (3,61)
SSF- λ -2	10,8 (4,3)	4,00 (2,78)	4,00 (3,74)
SSF- ρ -1	3,0 (0,0)	8,25 (3,72)	8,00 (3,84)
SSF- ρ -2	6,0 (0,0)	2,75 (2,08)	3,00 (1,87)
SSF-R-1	5,3 (3,1)	4,00 (2,00)	5,00 (2,24)
SSF-R-2	7,2 (3,7)	4,50 (2,69)	5,00 (3,35)
ACA-1	2,0 (0,0)	9,25 (2,25)	5,75 (1,95)
ACA-2	3,0 (0,0)	4,25 (2,25)	4,00 (1,66)
SSF-SU-1	3,8 (2,8)	7,00 (2,92)	5,00 (1,94)
SSF-SU-2	5,8 (4,3)	7,00 (4,30)	4,75 (2,84)
SSF-SU _S -1	8,5 (1,7)	4,25 (3,54)	4,00 (1,66)
SSF-SU _S -2	15,0 (1,5)	2,75 (1,75)	2,25 (2,36)
SSF-SU _S -KS-1	8,4 (1,3)	5,50 (2,92)	4,00 (2,29)
SSF-SU _S -KS-2	16,0 (1,9)	2,50 (2,24)	2,75 (2,08)
SSF-SU _S -I	15,0 (1,5)	2,50 (1,58)	2,75 (2,36)
SSF-SU _S -KS-I	16,0 (1,9)	2,00 (1,50)	2,00 (2,18)
MMP (M)	6,9 (3,4)	0,50 (1,00)	—
MMP (M)	5,8 (3,5)	—	0,75 (1,60)
MMP ($\bar{m}$)	—	6,23 (1,72)	5,13 (1,64)
MMP (P)	1,0 (0,0)	32,25 (3,94)	28,25 (4,88)
Todos	20	2,50 (2,50)	2,50 (2,96)

Tabela A.5: Resultados obtidos — Bio5.

Algoritmo	M^*	%-KNN	%-NB
SSF- λ -1	8,0 (0,0)	0,00 (0,00)	0,25 (0,75)
SSF- λ -2	16,0 (0,0)	0,00 (0,00)	0,25 (0,75)
SSF- ρ -1	5,0 (0,0)	0,00 (0,00)	0,50 (1,00)
SSF- ρ -2	10,0 (0,0)	0,00 (0,00)	0,00 (0,00)
SSF-R-1	2,0 (0,0)	0,50 (1,00)	0,00 (0,00)
SSF-R-2	3,0 (0,0)	0,75 (1,14)	0,50 (1,00)
ACA-1	2,0 (0,0)	0,50 (1,00)	0,00 (0,00)
ACA-2	3,0 (0,0)	0,25 (0,75)	0,00 (0,00)
SSF-SU-1	2,0 (0,0)	0,50 (1,00)	0,00 (0,00)
SSF-SU-2	3,0 (0,0)	0,75 (1,14)	0,50 (1,00)
SSF-SU _S -1	8,0 (0,8)	0,00 (0,00)	0,25 (0,75)
SSF-SU _S -2	15,4 (1,3)	0,00 (0,00)	0,25 (0,75)
SSF-SU _S -KS-1	9,1 (0,7)	0,00 (0,00)	0,25 (0,75)
SSF-SU _S -KS-2	16,9 (1,1)	0,00 (0,00)	0,25 (0,75)
SSF-SU _S -I	15,4 (1,3)	0,00 (0,00)	0,25 (0,75)
SSF-SU _S -KS-I	16,9 (1,1)	0,00 (0,00)	0,25 (0,75)
MMP (M)	2,4 (0,8)	0,00 (0,00)	—
MMP (M)	2,4 (0,8)	—	0,00 (0,00)
MMP ($\bar{m}$)	—	2,13 (0,58)	1,86 (0,66)
MMP (P)	1,0 (0,0)	30,75 (7,08)	26,5 (5,94)
Todos	20	0,00 (0,00)	0,25 (0,75)

Tabela A.6: Resultados obtidos — Yeast.

Algoritmo	M^*	%-KNN	%-NB
SSF- λ -1	2,0 (0,0)	10,71 (8,20)	9,24 (8,44)
SSF- λ -2	3,0 (0,0)	8,81 (5,22)	7,29 (7,23)
SSF- ρ -1	2,0 (0,0)	9,88 (7,76)	5,88 (3,04)
SSF- ρ -2	3,1 (0,3)	5,93 (4,38)	5,43 (4,16)
SSF-R-1	8,0 (0,7)	2,43 (3,23)	1,45 (2,22)
SSF-R-2	13,7 (1,1)	2,43 (3,23)	1,95 (2,39)
ACA-1	2,0 (0,0)	8,88 (7,68)	8,88 (6,25)
ACA-2	4,0 (0,0)	4,45 (3,49)	7,40 (7,16)
SSF-SU-1	8,0 (0,7)	2,43 (3,23)	1,45 (2,22)
SSF-SU-2	13,7 (1,1)	2,43 (3,23)	1,95 (2,39)
SSF-SU _S -1	7,5 (2,2)	2,90 (2,37)	2,43 (2,43)
SSF-SU _S -2	13,6 (3,3)	1,45 (2,22)	2,45 (2,45)
SSF-SU _S -KS-1	8,8 (1,4)	3,88 (6,00)	3,40 (4,31)
SSF-SU _S -KS-2	15,8 (1,5)	1,95 (2,39)	1,95 (2,39)
SSF-SU _S -I	13,6 (3,3)	1,48 (2,26)	2,45 (2,45)
SSF-SU _S -KS-I	15,8 (1,5)	2,45 (2,45)	1,95 (2,39)
MMP (M)	6,4 (2,9)	1,45 (2,22)	—
MMP (M)	5,0 (1,7)	—	1,45 (2,22)
MMP ($\bar{m}$)	—	8,59 (2,99)	7,14 (3,72)
MMP (P)	1,1 (0,3)	53,21 (9,53)	47,34 (9,98)
Todos	20	1,95 (2,39)	2,45 (3,32)

Tabela A.7: Resultados obtidos — Ionosphere.

Algoritmo	M^*	%-KNN	%-NB
SSF- λ -1	2,0 (0,0)	30,50 (6,33)	23,08 (5,19)
SSF- λ -2	3,4 (0,4)	22,78 (9,26)	24,49 (4,21)
SSF- ρ -1	11,5 (1,1)	11,68 (4,33)	13,67 (5,96)
SSF- ρ -2	21,6 (1,8)	11,66 (4,93)	17,63 (5,82)
SSF-R-1	14,2 (1,6)	11,68 (2,38)	14,51 (5,69)
SSF-R-2	25,1 (1,4)	11,39 (5,25)	14,22 (3,91)
ACA-1	3,2 (0,4)	16,81 (5,63)	23,06 (7,87)
ACA-2	6,4 (0,8)	12,83 (5,33)	32,48 (4,49)
SSF-SU-1	14,2 (1,6)	11,68 (2,38)	14,51 (5,69)
SSF-SU-2	24,9 (1,7)	11,96 (5,81)	14,22 (3,91)
SSF-SU _S -1	15,3 (1,1)	10,83 (4,93)	14,82 (6,62)
SSF-SU _S -2	26,8 (1,1)	13,95 (4,48)	14,79 (4,08)
SSF-SU _S -KS-1	15,1 (1,8)	9,670 (5,85)	17,35 (8,22)
SSF-SU _S -KS-2	26,9 (1,7)	12,53 (5,87)	12,82 (4,27)
SSF-SU _S -I	26,8 (1,1)	13,38 (6,11)	15,10 (4,60)
SSF-SU _S -KS-I	26,9 (1,7)	13,66 (7,04)	15,67 (4,30)
MMP (M)	11,2 (6,1)	7,13 (4,48)	—
MMP (M)	13,7 (12,6)	—	13,67 (2,74)
MMP ($\bar{m}$)	—	13,55 (4,50)	22,34 (4,05)
MMP (P)	2,0 (0,6)	28,22 (6,75)	—
MMP (P)	15,6 (8,7)	—	29,93 (5,95)
Todos	34	13,67 (6,71)	16,79 (3,83)

Tabela A.8: Resultados obtidos — Pima.

Algoritmo	M^*	%-KNN	%-NB
SSF- λ -1	2,0 (0,0)	37,37 (4,40)	32,03 (4,23)
SSF- λ -2	3,0 (0,0)	34,90 (6,14)	31,78 (3,66)
SSF- ρ -1	3,0 (0,0)	35,02 (4,27)	34,51 (3,73)
SSF- ρ -2	6,0 (0,0)	36,46 (4,49)	30,74 (6,05)
SSF-R-1	3,5 (0,6)	35,56 (6,08)	29,82 (6,90)
SSF-R-2	6,3 (0,4)	33,46 (5,22)	30,21 (6,10)
ACA-1	2,0 (0,0)	38,02 (3,70)	34,12 (5,54)
ACA-2	4,0 (0,0)	37,25 (5,25)	34,52 (5,54)
SSF-SU-1	3,5 (0,6)	35,56 (6,08)	29,82 (6,90)
SSF-SU-2	6,3 (0,4)	33,46 (5,22)	30,21 (6,10)
SSF-SU _S -1	2,5 (0,5)	36,58 (4,28)	33,08 (5,32)
SSF-SU _S -2	5,0 (1,0)	32,56 (3,80)	24,75 (5,44)
SSF-SU _S -KS-1	3,1 (0,5)	36,86 (6,66)	31,90 (4,14)
SSF-SU _S -KS-2	5,9 (0,8)	31,26 (4,19)	24,87 (3,17)
SSF-SU _S -I	5,0 (1,0)	36,74 (4,20)	30,48 (4,91)
SSF-SU _S -KS-I	5,9 (0,8)	35,43 (4,59)	30,99 (3,34)
MMP (M)	4,7 (2,4)	28,00 (4,51)	—
MMP (M)	5,4 (1,5)	—	23,05 (5,53)
MMP ($\bar{m}$)	—	36,43 (2,57)	30,89 (3,68)
MMP (P)	3,0 (0,6)	43,50 (3,55)	—
MMP (P)	3,1 (1,2)	—	37,64 (3,71)
Todos	8	30,22 (5,13)	24,35 (4,82)

Tabela A.9: Resultados obtidos — Wisconsin.

Algoritmo	M^*	%-KNN	%-NB
SSF- λ -1	2,6 (0,4)	7,32 (1,73)	6,16 (2,45)
SSF- λ -2	5,1 (1,1)	4,39 (2,53)	3,96 (1,74)
SSF- ρ -1	2,0 (0,0)	6,73 (1,49)	6,60 (2,58)
SSF- ρ -2	3,0 (0,0)	7,03 (2,27)	5,57 (2,62)
SSF-R-1	2,4 (1,2)	6,45 (2,28)	6,00 (2,39)
SSF-R-2	3,9 (0,3)	6,00 (1,67)	5,28 (2,11)
ACA-1	2,0 (0,0)	6,73 (1,49)	6,60 (2,58)
ACA-2	3,0 (0,0)	6,89 (1,76)	5,86 (1,75)
SSF-SU-1	2,4 (1,2)	6,45 (2,28)	6,00 (2,39)
SSF-SU-2	3,9 (0,3)	6,00 (1,67)	5,28 (2,11)
SSF-SU _S -1	4,0 (0,0)	6,74 (2,31)	4,39 (2,27)
SSF-SU _S -2	8,0 (0,0)	4,10 (1,29)	3,52 (1,75)
SSF-SU _S -KS-1	4,0 (0,0)	6,00 (3,03)	5,12 (2,70)
SSF-SU _S -KS-2	8,0 (0,0)	3,96 (1,48)	3,52 (1,75)
SSF-SU _S -I	8,0 (0,0)	4,10 (1,29)	3,52 (1,75)
SSF-SU _S -KS-I	8,0 (0,0)	3,96 (1,48)	3,52 (1,75)
MMP (M)	4,6 (0,6)	2,64 (1,71)	—
MMP (M)	4,2 (1,4)	—	2,49 (1,96)
MMP ($\bar{m}$)	—	6,92 (1,78)	6,94 (1,2)
MMP (P)	1,0 (0,0)	21,23 (3,14)	24,01 (2,89)
Todos	9	4,25 (1,68)	3,66 (1,76)

Tabela A.10: Resultados obtidos — Spambase.

Algoritmo	M^*	%-KNN	%-NB
SSF- λ -1	2,0 (0,0)	37,14 (1,02)	53,49 (1,22)
SSF- λ -2	3,0 (0,0)	32,71 (3,66)	51,68 (1,65)
SSF- ρ -1	20,9 (2,1)	11,78 (1,40)	31,90 (5,88)
SSF- ρ -2	36,7 (2,7)	11,02 (2,54)	25,97 (4,54)
SSF-R-1	25,4 (3,2)	12,26 (1,45)	29,99 (3,68)
SSF-R-2	43,5 (3,2)	10,54 (1,29)	22,34 (2,10)
ACA-1	6,0 (0,0)	16,32 (1,40)	39,64 (1,47)
ACA-2	12,0 (0,0)	13,39 (1,03)	34,10 (1,84)
SSF-SU-1	24,3 (3,4)	12,67 (2,11)	30,71 (3,90)
SSF-SU-2	42,2 (3,7)	11,00 (1,78)	22,58 (2,37)
SSF-SU _S -1	25,5 (2,3)	11,61 (1,15)	30,10 (4,02)
SSF-SU _S -2	44,7 (2,2)	9,85 (1,25)	21,06 (1,93)
SSF-SU _S -KS-1	24,6 (1,7)	11,26 (1,49)	28,71 (4,49)
SSF-SU _S -KS-2	44,1 (2,0)	9,69 (1,20)	20,21 (1,50)
SSF-SU _S -I	44,7 (2,2)	9,63 (1,64)	21,15 (1,61)
SSF-SU _S -KS-I	44,1 (2,0)	10,04 (1,03)	21,08 (1,73)
MMP (M)	52,9 (3,2)	8,39 (0,98)	—
MMP (M)	53,5 (3,5)	—	19,86 (1,65)
MMP ($\bar{m}$)	—	16,77 (0,80)	33,33 (1,58)
MMP (P)	1,2 (0,5)	39,32 (0,20)	60,07 (0,54)
Todos	57	9,15 (1,28)	20,28 (1,84)

Tabela A.11: Resultados obtidos — Colon Cancer.

Algoritmo	M^*	%-KNN	%-NB
SSF- λ -1	10,2 (13,0)	31,19 (22,01)	40,24 (12,25)
SSF- λ -2	12,1 (14,2)	36,67 (21,06)	41,43 (14,13)
SSF- ρ -1	2,3 (0,4)	49,76 (14,30)	41,67 (18,91)
SSF- ρ -2	4,6 (0,9)	36,90 (17,63)	32,14 (14,30)
SSF-R-1	2,5 (0,8)	43,81 (22,64)	54,05 (25,38)
SSF-R-2	5,0 (1,6)	43,10 (24,22)	38,81 (21,30)
ACA-1	40,6 (4,5)	24,76 (17,21)	40,71 (20,55)
ACA-2	81,2 (9,0)	24,29 (13,38)	42,38 (17,65)
SSF-SU-1	2,2 (0,4)	38,57 (20,50)	52,14 (18,76)
SSF-SU-2	4,4 (0,8)	32,14 (25,91)	42,62 (26,99)
SSF-SU _S -1	2,0 (0,0)	37,14 (18,17)	45,24 (20,78)
SSF-SU _S -2	4,0 (0,0)	28,81 (11,39)	33,57 (19,47)
SSF-SU _S -KS-1	2,2 (0,4)	35,48 (16,82)	28,81 (19,44)
SSF-SU _S -KS-2	4,4 (0,8)	24,05 (12,97)	29,05 (14,44)
SSF-SU _S -I	4,0 (0,0)	28,81 (15,52)	34,29 (27,03)
SSF-SU _S -KS-I	4,4 (0,8)	27,14 (18,23)	33,57 (14,06)
MMP (M)	1929,7 (11,9)	25,71 (10,64)	—
MMP (M)	1924,0 (10,5)	—	48,1 (19,51)
MMP ($\bar{m}$)	—	26,52 (11,12)	48,1 (19,51)
MMP (P)	1993,3 (17,1)	27,14 (11,82)	—
MMP (P)	1999,0 (0)	—	48,1 (19,51)
Todos	2000	25,71 (10,64)	48,1 (19,51)

Tabela A.12: Resultados obtidos — Leukemia.

Algoritmo	M^*	%-KNN	%-NB
SSF- λ -1	51,6 (9,7)	27,68 (17,74)	15,00 (9,15)
SSF- λ -2	53,3 (10,1)	27,68 (16,55)	15,00 (9,15)
SSF- ρ -1	2,0 (0,0)	26,79 (13,83)	29,29 (20,52)
SSF- ρ -2	4,0 (0,0)	37,50 (12,68)	30,89 (16,94)
SSF-R-1	2,0 (0,0)	28,04 (20,36)	36,07 (10,65)
SSF-R-2	4,0 (0,0)	37,68 (9,46)	36,07 (15,36)
ACA-1	84,0 (0,0)	18,04 (12,69)	16,61 (12,22)
ACA-2	168,0 (0,0)	19,29 (13,87)	8,04 (8,65)
SSF-SU-1	2,0 (0,0)	28,04 (18,25)	38,93 (12,00)
SSF-SU-2	4,0 (0,0)	37,68 (11,42)	36,07 (16,63)
SSF-SU _S -1	2,0 (0,0)	32,14 (11,09)	35,89 (14,34)
SSF-SU _S -2	4,0 (0,0)	9,29 (11,46)	9,46 (10,54)
SSF-SU _S -KS-1	5,7 (4,0)	5,18 (8,47)	3,93 (6,02)
SSF-SU _S -KS-2	11,3 (7,8)	5,36 (8,64)	2,68 (5,37)
SSF-SU _S -I	4,0 (0,0)	14,82 (10,29)	16,61 (12,22)
SSF-SU _S -KS-I	11,3 (7,8)	10,71 (11,09)	3,93 (6,02)
MMP (M)	7022,0 (36,9)	15,18 (14,67)	—
MMP (M)	7011,3 (28,5)	—	1,25 (3,75)
MMP ($\bar{m}$)	—	16,51 (17,44)	1,25 (3,75)
MMP (P)	7127,5 (1,5)	16,61 (17,68)	—
MMP (P)	7128,0 (0,0)	—	1,25 (3,75)
Todos	7129	15,18 (14,67)	1,25 (3,75)

Tabela A.13: Tempos computacionais (em segundos) dos experimentos de classificação. Computador: Opteron 2GHz com 8Gb de RAM.

Algoritmo	Bio1	Bio2	Bio3	Bio4	Bio5	Yeast	Iono	Pima	Wisc	Spam	Colon	Leu
SSF- λ -1	0,30 (0,20)	0,12 (0,00)	0,12 (0,00)	0,12 (0,00)	0,12 (0,00)	0,10 (0,00)	0,30 (0,00)	0,03 (0,00)	0,04 (0,00)	5,55 (0,00)	261,52 (8,32)	19140,80 (5442,16)
SSF- λ -2	0,06 (0,00)	0,05 (0,00)	0,05 (0,00)	0,05 (0,00)	0,05 (0,00)	0,04 (0,00)	0,12 (0,00)	0,02 (0,00)	0,02 (0,00)	1,80 (0,00)	258,69 (14,10)	18903,89 (4615,15)
SSF- ρ -1	0,16 (0,00)	0,13 (0,00)	0,12 (0,00)	0,13 (0,00)	0,12 (0,00)	0,10 (0,00)	0,33 (0,00)	0,04 (0,00)	0,04 (0,00)	6,13 (0,00)	93,87 (1,24)	2137,44 (50,53)
SSF- ρ -2	0,06 (0,00)	0,05 (0,00)	0,05 (0,00)	0,05 (0,00)	0,05 (0,00)	0,04 (0,00)	0,12 (0,00)	0,02 (0,00)	0,02 (0,00)	1,91 (0,00)	93,45 (1,01)	2131,35 (35,69)
SSF-R-1	0,17 (0,00)	0,13 (0,00)	0,14 (0,00)	0,13 (0,00)	0,13 (0,00)	0,10 (0,00)	0,36 (0,00)	0,04 (0,00)	0,04 (0,00)	7,19 (0,10)	112,80 (1,35)	1276,94 (62,88)
SSF-R-2	0,07 (0,00)	0,06 (0,00)	0,06 (0,00)	0,06 (0,00)	0,06 (0,00)	0,05 (0,00)	0,14 (0,00)	0,02 (0,00)	0,02 (0,00)	2,92 (0,10)	112,20 (1,78)	1284,68 (60,75)
ACA-1	0,15 (0,00)	0,13 (0,00)	0,13 (0,00)	0,13 (0,00)	0,13 (0,00)	0,10 (0,00)	0,34 (0,00)	0,04 (0,00)	0,04 (0,00)	7,13 (0,10)	117,12 (5,99)	1211,20 (44,79)
ACA-2	0,06 (0,00)	0,06 (0,00)	0,06 (0,00)	0,06 (0,00)	0,06 (0,00)	0,05 (0,00)	0,14 (0,00)	0,02 (0,00)	0,02 (0,00)	2,88 (0,00)	114,30 (5,22)	1231,95 (44,71)
SSF-SU-1	0,13 (0,00)	0,13 (0,00)	0,13 (0,00)	0,13 (0,00)	0,13 (0,00)	0,10 (0,00)	0,35 (0,00)	0,04 (0,00)	0,04 (0,00)	7,16 (0,00)	113,26 (3,27)	1136 (18,15)
SSF-SU-2	0,06 (0,00)	0,06 (0,00)	0,06 (0,00)	0,06 (0,00)	0,06 (0,00)	0,05 (0,00)	0,13 (0,00)	0,02 (0,00)	0,02 (0,00)	2,80 (0,00)	115,98 (6,36)	1140,68 (16,02)
SSF-SU _S -1	0,13 (0,00)	0,13 (0,00)	0,13 (0,00)	0,13 (0,00)	0,13 (0,00)	0,10 (0,00)	0,34 (0,00)	0,04 (0,00)	0,04 (0,00)	6,95 (0,10)	115,77 (5,40)	1295,11 (53,66)
SSF-SU _S -2	0,06 (0,00)	0,06 (0,00)	0,06 (0,00)	0,06 (0,00)	0,06 (0,00)	0,05 (0,00)	0,13 (0,00)	0,02 (0,00)	0,02 (0,00)	2,79 (0,00)	114,39 (4,40)	1284,32 (33,92)
SSF-SU _S -KS-1	0,16 (0,00)	0,13 (0,00)	0,13 (0,00)	0,14 (0,00)	0,13 (0,00)	0,10 (0,00)	0,34 (0,00)	0,04 (0,00)	0,05 (0,00)	7,51 (0,46)	171,36 (115,14)	4077,40 (355,56)
SSF-SU _S -KS-2	0,06 (0,00)	0,06 (0,00)	0,06 (0,00)	0,06 (0,00)	0,06 (0,00)	0,05 (0,00)	0,14 (0,00)	0,02 (0,00)	0,02 (0,00)	3,15 (0,17)	171,64 (114,86)	3964,30 (249,46)
SSF-SU _S -I	0,07 (0,00)	0,06 (0,00)	0,07 (0,00)	0,06 (0,00)	0,06 (0,00)	0,05 (0,00)	0,13 (0,00)	0,03 (0,00)	0,02 (0,00)	2,88 (0,00)	113,23 (1,56)	1285,42 (39,81)
SSF-SU _S -KS-I	0,07 (0,00)	0,06 (0,00)	0,06 (0,00)	0,07 (0,00)	0,07 (0,00)	0,05 (0,00)	0,14 (0,00)	0,03 (0,00)	0,02 (0,00)	3,25 (0,17)	173,56 (117,21)	4088,31 (279,82)
MMP-KNN	0,21 (0,00)	0,09 (0,00)	0,09 (0,00)	0,09 (0,00)	0,09 (0,00)	0,06 (0,00)	0,17 (0,00)	0,11 (0,00)	0,10 (0,00)	32,62 (1,35)	159,28 (17,81)	3965,47 (876,01)
MMP-NB	0,16 (0,00)	0,09 (0,00)	0,09 (0,00)	0,09 (0,00)	0,08 (0,00)	0,06 (0,00)	0,13 (0,00)	0,06 (0,00)	0,06 (0,00)	4,11 (0,00)	159,13 (17,74)	3557,92 (478,10)

Algoritmo	Bio1	Bio2	Bio3	Bio4	Bio5	Yeast	Iono	Pima	Wisc	Spam	Colon	Leu
SSF- λ -1	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲	▲▲	◇	▲▲	▼	▲▲▲	▲▲
SSF- λ -2	▲▲▲	▲▲▲	▲▲▲	▲	▲▲▲	▲▲	▲▲	▼	▲	▼	▲▲▲	▲▲
SSF- ρ -1	▲▲▲	▲▲	▲▲▲	▲▲	▲▲	▲▲	▲▲▲	▼	▲▲	◇	▲▲▲	◇
SSF- ρ -2	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲	▲	▼	▲▲	▲	▲▲▲	◇
SSF-R-1	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲▲	▲▲▲	◇	▲▲	◇	▲▲▲	▼
SSF-R-2	▲▲▲	▲▲	▲▲▲	▲▲	▲▲	▲▲▲	▲▲▲	▼	▲▲	▲	▲▲▲	▼
ACA-1	▲▲	▲▲	▲▲	▲▲	▲▲▲	▲▲	▲▲	▼	▲▲	▼	▲▲▲	▲▲
ACA-2	▲▲	▲▲	▲▲	▲▲	▲▲▲	▲▲	▼	▼	▲▲	◇	▲▲▲	▲▲
SSF-SU-1	▲▲	▲▲	▲▲	▲▲	▲▲▲	▲▲▲	▲▲▲	◇	▲▲	◇	▲▲▲	▼
SSF-SU-2	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲▲	▲▲▲	▼	▲▲	▲	▲▲▲	▼
SSF-SU _S -1	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲▲	▲▲▲	▼	▲▲	◇	▲▲▲	▼
SSF-SU _S -2	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲	▲▲▲	▲	▲▲▲	▲▲
SSF-SU _S -KS-1	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲	▲▲	▼	▲▲	▲▲	▲▲▲	▲▲
SSF-SU _S -KS-2	▲▲▲	▲▲▲	▲▲▲	▲	▲▲▲	▲▲▲	▲▲▲	▲	▲▲▲	▲▲▲	▲▲▲	▲▲
SSF-SU _S -I	▲▲▲	▲▲▲	▲▲▲	▲	▲▲▲	▲▲▲	▲▲▲	▼	▲▲▲	▲	▲▲▲	▲▲
SSF-SU _S -KS-I	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▼	▲▲▲	▲	▲▲▲	▲▲
MMP	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲

Tabela A.14: Avaliação multicritério — erro obtido pelo classificador NB.

Algoritmo	Bio1	Bio2	Bio3	Bio4	Bio5	Yeast	Iono	Pima	Wisc	Spam	Colon	Leu
SSF- λ -1	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲	◇	▼	▲▲	◇	◇	◇
SSF- λ -2	▲▲▲	▲▲▲	▲▲▲	▲	▲▲▲	▲▲	▲▲	▼	▲	◇	▼	◇
SSF- ρ -1	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▼	▲▲	▲▲	▼	◇
SSF- ρ -2	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▼	▲▲	▲	▼	▼
SSF-R-1	▲▲▲	▲▲	▲▲▲	▲▲	▲▲	▲▲	▲▲▲	▼	▲▲	▲▲	▼	◇
SSF-R-2	▲▲▲	▲▲	▲▲▲	▲▲	▲▲	▲	▲▲▲	▼	▲▲	▲	▼	▼
ACA-1	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲	▼	▲▲	▲▲	▲▲▲	▲▲
ACA-2	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲▲	▼	▲▲	▲▲	▲▲▲	▲▲
SSF-SU-1	▲▲▲	▲▲	▲▲	▲▲	▲▲	▲▲	▲▲▲	▼	▲▲	▲▲	▼	◇
SSF-SU-2	▲▲▲	▲▲	▲▲	▲▲	▲▲	▲	▲▲▲	▼	▲▲	▲	◇	▼
SSF-SU _S -1	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▼	▲▲	▲▲	▼	◇
SSF-SU _S -2	▲▲▲	▲	▲▲▲	▲	▲▲▲	▲▲▲	▲	▼	▲▲▲	▲	▲▲	▲▲▲
SSF-SU _S -KS-1	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▲▲	▲▲▲	▼	▲▲	▲▲	◇	▲▲▲
SSF-SU _S -KS-2	▲▲▲	▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲	▲▲▲	▲	▲▲▲	▲▲▲
SSF-SU _S -I	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▼	▲▲▲	▲	▲▲	▲▲▲
SSF-SU _S -KS-I	▲▲▲	▲	▲▲▲	▲▲▲	▲▲▲	▲	▲▲▲	▼	▲▲▲	▲	▲▲	▲▲▲
MMP	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲▲

Tabela A.15: Avaliação multicritério — erro obtido pelo classificador KNN.

Tabelas dos Resultados dos Experimentos de Agrupamento de Dados

Este apêndice contém as tabelas utilizadas na avaliação dos experimentos de SA em problemas de agrupamento. A Tabela B.1 apresenta o número de atributos selecionados por cada algoritmo de SA. As Tabelas B.2 e B.3 apresentam, respectivamente, os valores dos critérios externos (ARI e CJ) obtidos após a execução do algoritmo *k-means* nos subconjuntos selecionados por cada algoritmo de SA. Finalmente, a Tabela B.4 apresenta os tempos computacionais (em segundos) dispendidos na SA em cada base de dados.

Algoritmo	10_250	20_250	12_200	Bio1	Bio2	Bio3	Bio4	Bio5	Yeast	1000_1000
MMP (M)	6	14	11	17	14	4	6	9	8	4
MMP (P)	1	1	2	8	1	1	19	4	1	1
SSF- λ -1	2	2	2	3	10	9	4	8	2	2
SSF- λ -2	4	3	4	6	20	18	8	16	3	4
SSF- ρ -1	4	5	2	6	4	8	3	5	2	4
SSF- ρ -2	8	10	4	12	8	16	6	10	3	8
ACA-1	2	2	2	2	2	2	2	2	2	2
ACA-2	4	4	4	3	3	3	3	3	4	4
SSF-R-1	4	7	2	8	7	10	3	6	9	2
SSF-R-2	8	14	4	12	11	15	6	11	15	3
SSF-SU-1	4	6	2	8	7	10	2	6	9	2
SSF-SU-2	8	11	4	12	11	15	3	11	15	3

Tabela B.1: Sumário do número de atributos selecionados nos problemas de agrupamento.

Tabela B.2: Sumário dos valores de **ARI** obtidos.

Algoritmo	10_250	20_250	12_200	Bio1	Bio2	Bio3	Bio4	Bio5	Yeast	1000_1000
Todos	1,00	0,95	1,00	0,53	0,80	1,00	0,82	0,55	0,97	0,78
MMP (M)	1,00	0,98	1,00	0,82	0,97	1,00	0,90	0,81	0,98	0,93
MMP ($\bar{m}$)	0,57	0,53	0,56	0,64	0,78	0,92	0,75	0,69	0,76	0,78
MMP (P)	0,02	0,02	0,26	0,49	0,53	0,55	0,52	0,48	0,09	0,28
SSF- λ -1	0,49	0,11	0,43	0,81	0,50	1,00	0,77	0,54	0,62	0,56
SSF- λ -2	0,31	0,17	0,45	0,59	0,80	1,00	0,79	0,55	0,65	0,90
SSF- ρ -1	1,00	1,00	1,00	0,53	0,91	1,00	0,82	0,54	0,75	0,78
SSF- ρ -2	1,00	1,00	0,52	0,53	0,81	1,00	0,80	0,54	0,66	0,78
ACA-1	0,49	1,00	1,00	1,00	0,62	0,78	0,80	0,79	0,60	0,78
ACA-2	1,00	0,99	0,99	1,00	0,54	0,81	0,92	0,54	0,71	1,00
SSF-R-1	0,24	0,97	1,00	1,00	0,91	0,81	0,93	0,53	0,96	0,76
SSF-R-2	0,25	0,56	0,52	0,82	0,92	0,78	0,80	0,56	0,97	0,67
SSF-SU-1	0,24	1,00	1,00	1,00	0,91	0,81	0,87	0,53	0,96	0,76
SSF-SU-2	0,25	1,00	0,52	0,82	0,92	0,78	0,79	0,56	0,97	0,67

Tabela B.3: Sumário dos valores de **CJ** obtidos.

Algoritmo	10_250	20_250	12_200	Bio1	Bio2	Bio3	Bio4	Bio5	Yeast	1000_1000
Todos	1,00	0,92	1,00	0,47	0,72	1,00	0,74	0,48	0,96	0,64
MMP (M)	1,00	0,97	1,00	0,75	0,94	1,00	0,85	0,73	0,97	0,88
MMP ($\bar{m}$)	0,56	0,53	0,54	0,56	0,70	0,89	0,67	0,61	0,76	0,64
MMP (P)	0,13	0,13	0,29	0,43	0,45	0,46	0,46	0,42	0,25	0,16
SSF- λ -1	0,42	0,17	0,42	0,74	0,44	1,00	0,68	0,47	0,60	0,39
SSF- λ -2	0,29	0,21	0,42	0,51	0,72	1,00	0,70	0,48	0,63	0,82
SSF- ρ -1	1,00	1,00	1,00	0,47	0,86	1,00	0,73	0,47	0,72	0,64
SSF- ρ -2	1,00	1,00	0,49	0,46	0,73	1,00	0,72	0,47	0,64	0,64
ACA-1	0,44	1,00	1,00	1,00	0,54	0,69	0,72	0,71	0,59	0,64
ACA-2	1,00	0,98	0,98	1,00	0,47	0,73	0,88	0,47	0,69	1,00
SSF-R-1	0,25	0,95	1,00	1,00	0,85	0,73	0,89	0,46	0,95	0,61
SSF-R-2	0,26	0,49	0,49	0,75	0,88	0,70	0,72	0,49	0,96	0,51
SSF-SU-1	0,25	1,00	1,00	1,00	0,85	0,73	0,80	0,46	0,95	0,61
SSF-SU-2	0,26	1,00	0,49	0,75	0,88	0,70	0,71	0,49	0,96	0,51

Tabela B.4: Tempos computacionais (em segundos) dos experimentos de agrupamento de dados. Computador: Opteron 2GHz com 8Gb de RAM.

Algoritmo	10_250	20_250	12_200	Bio1	Bio2	Bio3	Bio4	Bio5	Yeast	1000_1000
MMP	0,25	0,51	0,13	0,45	0,52	0,36	0,41	0,47	0,20	3834,10
SSF- λ -1	0,18	0,08	0,02	0,09	0,08	0,08	0,08	0,08	0,05	3275,55
SSF- λ -2	0,08	0,03	0,01	0,06	0,03	0,03	0,04	0,03	0,02	943,64
SSF- ρ -1	0,23	0,24	0,03	0,11	0,12	0,12	0,14	0,10	0,06	3574,77
SSF- ρ -2	0,03	0,13	0,02	0,08	0,04	0,09	0,04	0,03	0,02	1024,12
ACA-1	0,22	0,16	0,02	0,14	0,12	0,11	0,14	0,09	0,06	2718,35
ACA-2	0,06	0,07	0,02	0,08	0,06	0,04	0,06	0,04	0,03	821,09
SSF-R-1	0,11	0,12	0,03	0,12	0,13	0,10	0,12	0,09	0,06	3027,73
SSF-R-2	0,07	0,06	0,01	0,08	0,09	0,04	0,04	0,04	0,03	984,61
SSF-SU-1	0,22	0,33	0,05	0,12	0,12	0,10	0,15	0,15	0,11	3044,77
SSF-SU-2	0,05	0,14	0,02	0,07	0,05	0,06	0,05	0,04	0,03	974,95