

LEILA MARIA VRIESMANN

**SELEÇÃO DINÂMICA DE SUBCONJUNTO
DE CLASSIFICADORES: ABORDAGEM
BASEADA EM ACURÁCIA LOCAL**

Tese apresentada ao Programa de Pós-Graduação em
Informática da Pontifícia Universidade Católica do
Paraná como requisito parcial para a obtenção do
título de Doutor em Informática.

**CURITIBA
2012**

LEILA MARIA VRIESMANN

**SELEÇÃO DINÂMICA DE SUBCONJUNTO
DE CLASSIFICADORES: ABORDAGEM
BASEADA EM ACURÁCIA LOCAL**

Tese apresentada ao Programa de Pós-Graduação em
Informática da Pontifícia Universidade Católica do
Paraná como requisito parcial para a obtenção do
título de Doutor em Informática.

Área de concentração: *Ciência da Computação*

Orientador: Alceu de Souza Britto Junior

Co-orientador: Luiz Eduardo Soares de Oliveira

**CURITIBA
2012**

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central

Vriesmann, Leila Maria
V982s Seleção dinâmica de subconjunto de classificadores : abordagem baseada
2012 em acurácia local / Leila Maria Vriesmann ; orientador, Alceu de Souza Britto
Junior ; co-orientador, Luiz Eduardo Soares de Oliveira. – 2012.
xxvi, 108 f. : il. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Paraná, Curitiba,
2012
Bibliografia: f. [103]-108

1. Reconhecimento de padrões. 2. Aprendizado do computador. 3.
Algoritmos. 4. Informática. I. Britto Júnior, Alceu de Souza, 1966-. II. Oliveira,
Luiz Eduardo Soares de. III. Pontifícia Universidade Católica do Paraná.
Programa de Pós-Graduação em Informática. IV. Título.
CDD 20. ed. – 006.4

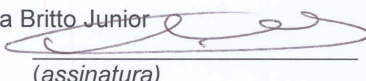
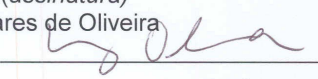
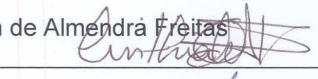

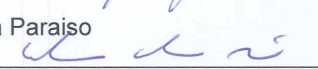
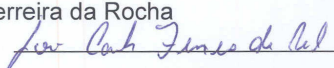



Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática

ATA DE DEFESA DE TESE DE DOUTORADO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

ÁREA DE CONCENTRAÇÃO: CIÊNCIA DA COMPUTAÇÃO
DEFESA DE TESE DE DOUTORADO Nº 011/2012

Ao 01 dia de Junho de 2012 realizou-se a sessão pública da Defesa da Tese de Doutorado “**Seleção Dinâmica de Subconjunto de Classificadores: Abordagem Baseada em Acurácia Local**” apresentada pela aluna **Leila Maria Vriesmann** como requisito parcial para a obtenção do título de Doutor em Informática, perante uma Banca Examinadora composta pelos seguintes membros:

Prof. Dr. Alceu de Souza Britto Junior PUCPR (Orientador)	 (assinatura)	<u>APROV.</u> (aprov/reprov.)
Prof. Dr. Luiz Eduardo Soares de Oliveira UFPR (co-orientador)		<u>APROV</u>
Prof ^a . Dr ^a . Cinthia Obladen de Almendra Freitas PUCPR		<u>APROVADO</u>
Prof. Dr. Alessandro Lameiras Koerich PUCPR		<u>APROV.</u>
Prof. Dr. Emerson Cabrera Paraiso PUCPR		<u>APROV.</u>
Prof. Dr. José Carlos Ferreira da Rocha UEPG		<u>Aprovado</u>
Prof. Dr. Flávio Bortolozzi CESUMAR		<u>APROVADO</u>





Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática

Conforme as normas regimentais do PPGIa e da PUCPR, o trabalho apresentado foi considerado APROVADA (aprovado/reprovado), segundo avaliação da maioria dos membros desta Banca Examinadora. Este resultado está condicionado ao cumprimento integral das solicitações da Banca Examinadora registradas no Livro de Defesas do programa.

Prof. Dr. Mauro Sérgio Pereira Fonseca
Diretor do Programa de Pós-Graduação em Informática



*Dedico à minha avó e madrinha
Martha, ao meu avô e padrinho
Bernardo (in memoriam), à minha
mãe Bernadete e ao meu pai Lolke.*

Agradecimentos

A Deus, por tudo que tem me propiciado.

Ao professor Alceu (orientador) e ao professor Luiz Eduardo (co-orientador), pela ajuda com ideias, ensinamentos, pela luta para minha permanência no Programa de Pós-Graduação em Informática (PPGIA) na Pontifícia Universidade Católica do Paraná (PUC-PR) e, mais que isso, por acreditarem no meu potencial para o trabalho. Mais que orientadores, são amigos.

Aos membros da Banca Examinadora Flávio Bortolozzi, José Carlos Ferreira da Rocha, Alessandro Lameiras Koerich, Emerson Cabrera Paraiso, Cinthia Obladen de Almendra Freitas, Luiz Eduardo Soares de Oliveira e Alceu de Souza Brito Júnior, pelas valorosas sugestões para esta tese.

Aos diretores Mauro Sérgio Pereira Fonseca e Fabrício Enembreck, à secretária Cheila Cristina Farias e outras pessoas do PPGIA que me ajudaram em diferentes procedimentos para o curso de Doutorado. Aos professores Jacques Facon, Manoel Camillo de Oliveira Penna Neto, Hugo Vieira Neto (UTFPR), Júlio Cesar Nievola e todos os professores que ministraram disciplinas ou participaram de comissões relacionadas ao meu Doutorado.

Aos colegas e amigos do PPGIA Mariza Miola Dosciatti, Edenilson José da Silva, Carla Machado da Trindade, José Adilson Lopes da Silva, André Pinz Borges, Osmar Betazzi Dordal, Helyane Bronoski Borges, Richardson Ribeiro, Pedro Luiz de Paula Filho, Marcos Aurelio Pchek Laureano, Tânia Lúcia Monteiro, Marcio Luiz Ferreira Miguel, Ederson Marcos Sgarbi, Glauco Carlos Silva, Andreia Marini e tantos mais que partilharam comigo as mesmas disciplinas (PUC-PR, UFPR ou UTFPR), os mesmos laboratórios ou a mesma sala de doutorandos nesses anos. Às mestrandas da UTFPR, Luciana Vilas Boas Wiecheteck e Ana Maria Pereira, pelas conversas após a disciplina onde adquirimos os conhecimentos básicos em LATEX. Ao Yandre Maldonado e Gomes da Gosta (doutorando na UFPR), à Priscila Ribas da Costa (mestranda na PUC-PR) e ao Albert Hung-Ren Ko, por contribuições em materiais de pesquisa. Ao professor Robert Sabourin, pela ajuda em artigos.

Aos colegas de trabalho (e amigos) Alaine Margarete Guimarães, Alceu de Souza Brito Júnior, José Carlos Ferreira da Rocha, João Umberto Furquim de Souza (*in memoriam*), Márcio Augusto de Souza, Luciano José Senger, Jorim Souza Das Virgens Filho, Maria Salete Marcon Gomes Vaz, Dierone César Foltran Júnior, Tatiana Montes Celinski, Diolete Marcante Lati Cerutti, Rosane Falate, Frederico Guilherme de Paula Ferreira Ielo, Marcelo Ferrasa, Victor George Celinski e demais professores do Departamento de Informática (DEINFO) da Universidade Estadual de Ponta Grossa (UEPG), pelo incentivo. À secretária Marinês do Rocio Ferreira, sempre disposta a ajudar nos procedimentos do DEINFO.

Aos amigos adquiridos durante a vida acadêmica: Fernando Henrique Pereira Leal, Daniel Martineschen e Silvio Alexandre Porto, pelo companheirismo.

Ao meu pai Lolke, à minha mãe Bernadete, aos meus avós Martha e Bernardo (*in memoriam*), à Lúcia, ao Lourenço, ao João Luiz, à Ana, à Lidvina e ao Pedro, pela dedicação e por terem me acompanhado em todos esses anos de estudo. À Melissa, à Ana Clara e ao Bernardo, pelos sorrisos propiciados. À Margarida, ao José Mário, à Charlene e ao Mário, pela torcida. A todos que estiveram presentes, incentivando e rezando pelo meu sucesso acadêmico.

À CAPES e à PUC-PR, pela bolsa de estudo concedida. À UEPG, pela concessão de afastamento das atividades pelo período de um ano, o qual foi dedicado exclusivamente à conclusão dos estudos no Doutorado.

Àqueles que compreenderam tantas ausências minhas, causadas pelas minhas atividades de estudo ou de trabalho. A todos que me auxiliaram de alguma maneira, direta ou indiretamente, para que eu chegasse aqui. Muito obrigada!

O poder nasce do querer. Sempre que o homem aplicar a veemência e perseverante energia de sua alma a um fim, vencerá os obstáculos e, se não atingir o alvo, pelo menos fará coisas admiráveis.

José de Alencar

Porque melhor é a sabedoria do que os rubis; e tudo o que mais se deseja não se pode comparar com ela.

Provérbios 8:11

Resumo

A classificação pode ser considerada a principal tarefa na área de reconhecimento de padrões. Sua função é atribuir uma classe a um padrão de teste desconhecido. É sabido que métodos de classificação baseados em conjunto de classificadores frequentemente apresentam melhores taxas de reconhecimento que métodos baseados em classificadores individuais. Neste contexto, o presente trabalho aborda a seleção dinâmica de classificadores para a composição do conjunto a ser utilizado para o reconhecimento de um padrão desconhecido. Diferente da seleção estática, na abordagem dinâmica assume-se a existência de um oráculo que determina os classificadores com maior competência para a classificação de cada padrão de teste apresentado. O objetivo principal deste trabalho é propor métodos para a seleção dinâmica de subconjunto de classificadores baseando-se em informações de vizinhança, com foco em acurácia local. Para este propósito, estudos preliminares foram realizados com base em um método de seleção dinâmica de subconjunto de classificadores que usa o conceito de oráculo disponível na literatura (KNORA). Estes estudos consideraram: a avaliação de diferentes métricas para determinação da vizinhança de cada padrão de teste; o uso de informações adicionais oriundas de técnicas de agrupamento e o uso de informações geradas através da ordenação dos classificadores disponíveis no conjunto inicial. Tais investigações permitiram a proposta de três novas abordagens para a seleção dinâmica de subconjunto de classificadores. A primeira baseou-se na fusão de informações disponíveis nos estágios do método KNORA original. A segunda, denominada DESCOLA, utilizou a acurácia local geral e a acurácia de classe local como informações complementares em um algoritmo de duas fases. Finalmente, a terceira, denominada DESLCA, apresentou apenas a fase de acurácia de classe local, sendo caracterizada como uma derivação do DESCOLA. Um protocolo experimental robusto que utilizou oito bases de dados, sendo duas de grande dimensão, demonstrou que os três métodos propostos foram promissores. Os resultados foram comparados com o método KNORA original usado nos estudos preliminares, e com outras abordagens de combinação de classificadores. Observou-se que nos experimentos realizados, as melhores taxas de reconhecimento sempre foram obtidas com o DESCOLA ou o DESLCA.

Palavras-chave: sistemas baseados em múltiplos classificadores, oráculos mais próximos, acurácia local.

Abstract

The classification may be considered the most important task in the pattern recognition area. It assigns a class to an unknown test pattern. Classification methods based on ensemble of classifiers usually present a better performance than individual classifiers. In this context, the present work deal with the dynamic selection of classifiers to compose the ensemble to be used to recognize an unknown pattern. Different from the static selection, in the dynamic approach, one assumes the existence of an oracle that determines the classifiers with competence to classify each testing sample. The main objective of the present work is to propose methods for dynamic selection of ensemble of classifiers by considering neighborhood information with focus on local accuracy. For this purpose, preliminary studies based on a method of dynamic selection of ensembles of classifiers that uses the concepts of oracle (KNORA) available in the literature were done. These studies consider: the evaluation of different distance metrics to determine the neighborhood of each test pattern; the use of additional information obtained through cluster analysis; and the use of information obtained through ranking the classifiers of the initial pool. These investigations support the proposal of three new methods. The first method is based on the fusion of information available in the stages of the original KNORA method. The second one, named DESCOLA, uses the overall local accuracy and the local class accuracy as complementary information in a two-step algorithm. Finally, the third one, named DESLCA, was derived from DESCOLA by considering just the local class accuracy. A robust experimental protocol based on eight datasets (two large-scale datasets) has shown that the three proposed methods are very promising. All the obtained results were compared against the original KNORA method and some common used strategies to combine classifiers available in the literature. The best results were always obtained with DESCOLA or DESLCA.

Keywords: multiple classifier systems, nearest oracles, local accuracy.

Sumário

Resumo	xi
Abstract	xiii
Lista de Figuras	xvii
Lista de Tabelas	xviii
Lista de Abreviações	xxiii
1 Introdução	1
1.1 Descrição do Problema e Proposta	2
1.2 Objetivos	4
1.3 Contribuições	4
1.4 Estrutura do Trabalho	5
2 Estado da Arte	7
2.1 Conjunto de Classificadores e Classificadores Individuais	7
2.2 Criação de Conjuntos de Classificadores	8
2.2.1 Topologia	9
2.2.2 Componentes do Sistema	9
2.2.3 Classificação	14
2.3 Seleção de Subconjunto de Classificadores	15
2.3.1 Método KNORA	16
2.3.2 Estratégia Dinâmica de Superprodução e Seleção	19
2.3.3 Seleção Baseada em Acurácia e Diversidade	22
2.3.4 Método GDES	23
2.3.5 Oráculo Aleatório Linear	25
2.3.6 Modelo Probabilístico de Competência de Classificador	25
2.4 Discussão	27
2.5 Conclusão	29
3 Métodos Propostos	31
3.1 Abordagens Preliminares	31
3.1.1 Adaptação de Novas Métricas de Distância ao Método KNORA	32
3.1.2 Método de Seleção baseada em Oráculos Mais Próximos com Informações Adicionais de Agrupamento	33

3.1.3	Uso de Informações sobre Proporção de Classes Reconhecidas	34
3.2	Fusão do Método KNORA com o Método k -NN	36
3.3	Método de Seleção baseada em Acurácia Local Geral e de Classe	39
3.4	Conclusão	47
4	Resultados e Discussão	49
4.1	Bases de Dados e Conjunto Inicial de Classificadores	49
4.1.1	Cálculo da Vizinhaça para Experimentos com a Base <i>NIST SD19</i>	51
4.2	Parâmetros de Comparação	53
4.3	Testes Estatísticos	55
4.4	Abordagens Preliminares	56
4.4.1	Adaptação de Novas Métricas de Distância ao Método KNORA	56
4.4.2	Método de Seleção baseada em Oráculos Mais Próximos com Informa- ções Adicionais de Agrupamento	67
4.4.3	Uso de Informações sobre Proporção de Classes Reconhecidas	72
4.5	Fusão do Método KNORA com o Método k -NN	74
4.6	Método de Seleção baseada em Acurácia Local Geral e de Classe	90
4.7	Conclusão	95
5	Conclusão	99
	Referências Bibliográficas	102

Lista de Figuras

2.1	Topologia em cascata (baseada em [Ranawana & Palade, 2006])	9
2.2	Topologia em paralelo (baseada em [Ranawana & Palade, 2006])	10
2.3	Topologia hierárquica (baseada em [Ranawana & Palade, 2006])	10
2.4	Representação do <i>bagging</i> (baseada em [Bittencourt, 2005])	11
2.5	Representação do <i>boosting</i> (baseada em [Bittencourt, 2005])	12
2.6	Representação de subespaços aleatórios (ilustração baseada em [Maia, 2008]) .	12
2.7	Representação do <i>stacking</i> (ilustração de [Maia, 2008])	13
2.8	Representação da seleção estática e da seleção dinâmica de subconjunto de clas- sificadores (adaptada de [Ko <i>et al.</i> , 2008])	16
2.9	KNORA-ELIMINATE (baseado em [Ko <i>et al.</i> , 2008])	18
2.10	KNORA-UNION (baseado em [Ko <i>et al.</i> , 2008])	18
2.11	Visão geral do processo OCS (baseada em [Santos, 2008])	20
2.12	Visão geral do processo DOCS (baseada em [Santos, 2008])	20
3.1	Representação do método KNORA CU	37
3.2	Representação do método KNN+KNORA CF	37
3.3	Representação do método KNN CU	38
3.4	Representação do método KNORA+KNN CF	38
3.5	Representação do método KNN+KNORA UF	39
3.6	Representação do método DESCOLA	40
3.7	Representação do método DESCOLA-ELIMINATE	42
3.8	Representação do método DESCOLA-UNION	43
3.9	Representação do método DESCOLA-UNION-U	44
4.1	Resultados obtidos com os k vizinhos mais próximos e resultados obtidos com a metodologia apresentada de delimitação dos vizinhos na base de dados <i>NIST</i> <i>SD19</i> , para o método k -NN e para o método KNORA no conjunto inicial de classificadores criados por meio de subespaços aleatórios	52
4.2	Taxas de reconhecimento na estratégia KNORA(ELIMINATE) CU	80
4.3	Taxas de reconhecimento na estratégia KNORA(UNION) CU	81
4.4	Taxas de reconhecimento na estratégia KNN+KNORA(ELIMINATE) CF	81
4.5	Taxas de reconhecimento na estratégia KNN+KNORA(UNION) CF	82
4.6	Taxas de reconhecimento no KNN (após KNORA-ELIMINATE) CU	82
4.7	Taxas de reconhecimento no KNN (após KNORA-UNION) CU	83
4.8	Taxa de reconhecimento utilizando a estratégia KE+KNN CF	83
4.9	Taxa de reconhecimento utilizando a estratégia KU+KNN CF	84
4.10	Taxas de reconhecimento para KNORA-UNION, KNORA-ELIMINATE, KNN+KNORA(UNION) UF e KNN+KNORA(ELIMINATE) UF	84

Lista de Tabelas

4.1	Características das bases de dados	50
4.2	Melhores taxas de reconhecimento (TR) e valores de desvio padrão (σ) obtidos para comparação nos experimentos com conjunto inicial de classificadores criado com subespaços aleatórios	53
4.3	Melhores taxas de reconhecimento (TR) e valores de desvio padrão (σ) obtidos para comparação nos experimentos com conjunto inicial de classificadores criado com <i>bagging</i>	54
4.4	Melhores taxas de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando a distância Euclidiana e a distância de Canberra no método KNORA aplicado no conjunto inicial de classificadores formados por subespaços aleatórios	57
4.5	Melhores taxas de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando a distância Euclidiana e a distância de Canberra no método KNORA aplicado no conjunto inicial de classificadores formados por <i>bagging</i>	57
4.6	Taxa de reconhecimento (em %) para cada k variando de 1 a 30 utilizando a distância Euclidiana e a distância de Canberra no KNORA para o conjunto inicial de classificadores criados por subespaços aleatórios da base de dados <i>NIST SD19</i>	58
4.7	Valores de z obtidos no teste para diferenças entre proporções para KE e KU utilizando a distância Euclidiana e para KE e KU utilizando a distância de Canberra, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados <i>NIST SD19</i>	59
4.8	Valores de z obtidos no teste para diferenças entre proporções para KE e KU utilizando a distância Euclidiana e para KE e KU utilizando a distância de Canberra, aplicado no conjunto inicial de classificadores formados por <i>bagging</i> da base de dados <i>NIST SD19</i>	59
4.9	Melhores taxas de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando a distância Euclidiana e a distância de Cosine no método KNORA aplicado no conjunto inicial de classificadores formados por subespaços aleatórios	60
4.10	Melhores taxas de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando a distância Euclidiana e a distância de Cosine no método KNORA aplicado no conjunto inicial de classificadores formados por <i>bagging</i>	60
4.11	Taxa de reconhecimento (em %) para cada k variando de 1 a 30 utilizando a distância Euclidiana e a distância de Cosine no KNORA para o conjunto inicial de classificadores criados por subespaços aleatórios da base de dados <i>NIST SD19</i>	61

4.12	Valores de z obtidos no teste para diferenças entre proporções para KE e KU utilizando a distância Euclidiana e para KE e KU utilizando a distância de Cosine, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados <i>NIST SD19</i>	62
4.13	Valores de z obtidos no teste para diferenças entre proporções para KE e KU utilizando a distância Euclidiana e para KE e KU utilizando a distância de Cosine, aplicado no conjunto inicial de classificadores formados por <i>bagging</i> da base de dados <i>NIST SD19</i>	62
4.14	Melhores taxas de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando a distância Euclidiana e a distância de Pearson no método KNORA aplicado no conjunto inicial de classificadores formados por subespaços aleatórios	63
4.15	Melhores taxas de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando a distância Euclidiana e a distância de Pearson no método KNORA aplicado no conjunto inicial de classificadores formados por <i>bagging</i>	63
4.16	Taxa de reconhecimento (em %) para cada k variando de 1 a 30 utilizando a distância Euclidiana e a distância de Pearson no KNORA para o conjunto inicial de classificadores criados por subespaços aleatórios da base de dados <i>NIST SD19</i>	64
4.17	Valores de z obtidos no teste para diferenças entre proporções para KE e KU utilizando a distância Euclidiana e para KE e KU utilizando a distância de Pearson, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados <i>NIST SD19</i>	65
4.18	Valores de z obtidos no teste para diferenças entre proporções para KE e KU utilizando a distância Euclidiana e para KE e KU utilizando a distância de Pearson, aplicado no conjunto inicial de classificadores formados por <i>bagging</i> da base de dados <i>NIST SD19</i>	65
4.19	Melhores médias da taxa de reconhecimento (em %) e respectivos valores de desvio padrão obtidos utilizando o algoritmo KLUNORA no conjunto inicial de classificadores gerados com subespaços aleatórios e <i>1-NN</i>	68
4.20	Melhores médias de taxa de reconhecimento (em %) e respectivos valores de desvio padrão obtidos utilizando o algoritmo KLUNORA no conjunto inicial de classificadores gerados com <i>bagging</i> e <i>1-NN</i>	69
4.21	Frequência absoluta de instâncias de validação nos agrupamentos do método <i>K</i> -médias, categorizados por classes, para a base de dados <i>NIST SD19</i>	70
4.22	Taxa de reconhecimento (em %) para k variando de 1 a 30 utilizando os métodos KNORA-ELIMINATE, KNORA-UNION, KLUNORA-ELIMINATE e KLUNORA-UNION no conjunto inicial de classificadores criados com subespaços aleatórios da base de dados <i>NIST SD19</i>	71
4.23	Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA e KLUNORA, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados <i>NIST SD19</i>	71
4.24	Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA e KLUNORA, aplicado no conjunto inicial de classificadores formados por <i>bagging</i> da base de dados <i>NIST SD19</i>	72

4.25	Melhores médias de taxa de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando probabilidades de classe por classificador como peso no método de oráculos mais próximos no conjunto inicial de classificadores formados por subespaços aleatórios	73
4.26	Melhores médias de taxa de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando probabilidades de classe por classificador como peso no método de oráculos mais próximos no conjunto inicial de classificadores formados por <i>bagging</i>	73
4.27	Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA e KNORA-P, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados <i>NIST SD19</i>	74
4.28	Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA e KNORA-P, aplicado no conjunto inicial de classificadores formados por <i>bagging</i> da base de dados <i>NIST SD19</i>	74
4.29	Melhores médias de taxa de reconhecimento (TR) e valores de desvio padrão correspondentes (σ) obtidos pelos esquemas condicionais de fusão do KNORA com o k -NN nos conjuntos iniciais de classificadores criados utilizando subespaços aleatórios	76
4.30	Melhores médias de taxa de reconhecimento (TR) e valores de desvio padrão correspondentes (σ) obtidos pelos esquemas incondicionais de fusão do KNORA com o k -NN nos conjuntos iniciais de classificadores criados utilizando subespaços aleatórios	76
4.31	Melhores médias de taxa de reconhecimento (TR) e valores de desvio padrão correspondentes (σ) obtidos pelos esquemas condicionais de fusão do KNORA com o k -NN nos conjuntos iniciais de classificadores criados utilizando <i>bagging</i>	77
4.32	Melhores médias de taxa de reconhecimento (TR) e valores de desvio padrão correspondentes (σ) obtidos pelos esquemas incondicionais de fusão do KNORA com o k -NN nos conjuntos iniciais de classificadores criados utilizando <i>bagging</i>	77
4.33	Métodos que mostraram diferenças utilizando correção de Dunn-Sidak e de Bonferroni-Dunn no teste de múltiplas comparações com os métodos MAJ, KNORA e de fusão do k -NN com o KNORA, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios	78
4.34	Métodos que mostraram diferenças utilizando correção de Bonferroni-Dunn no teste de múltiplas comparações com os métodos MAJ, KNORA e de fusão do k -NN com o KNORA, aplicado no conjunto inicial de classificadores formados por <i>bagging</i>	78
4.35	Melhores taxas de reconhecimento em % com seus valores de (k) correspondentes para os esquemas condicionais da fusão do KNORA com o k -NN, aplicados no conjunto inicial de classificadores criados com subespaços aleatórios na base de dados <i>NIST SD19</i>	79
4.36	Melhores taxas de reconhecimento em % com seus valores de (k) correspondentes para os esquemas incondicionais da fusão do KNORA com o k -NN, aplicados no conjunto inicial de classificadores criados com subespaços aleatórios na base de dados <i>NIST SD19</i>	79

4.37	Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA e de fusão do k -NN com o KNORA, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados <i>NIST SD19</i>	86
4.38	Distinções no teste para diferenças entre proporções com os métodos MAJ, KNORA e de fusão do k -NN com o KNORA, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados <i>NIST SD19</i>	87
4.39	Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA e de fusão do k -NN com o KNORA, aplicado no conjunto inicial de classificadores formados por <i>bagging</i> da base de dados <i>NIST SD19</i>	88
4.40	Distinções no teste para diferenças entre proporções com os métodos MAJ, KNORA e de fusão do k -NN com o KNORA, aplicado no conjunto inicial de classificadores formados por <i>bagging</i> da base de dados <i>NIST SD19</i>	89
4.41	Melhor média de taxa de reconhecimento (TR) e valores de desvio padrão correspondentes (σ) utilizando subespaços aleatórios para a criação do conjunto inicial de classificadores	91
4.42	Melhor média de taxa de reconhecimento (TR) e desvios de padrão correspondentes (σ) utilizando <i>bagging</i> para a criação do conjunto inicial de classificadores	92
4.43	Métodos que mostraram diferenças utilizando correção de Dunn-Sidak no teste de múltiplas comparações com os métodos MAJ, KNORA, DESCOLA e DESLCA, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios	93
4.44	Métodos que mostraram diferenças utilizando correção de Dunn-Sidak e de Bonferroni-Dunn no teste de múltiplas comparações com os métodos MAJ, KNORA, DESCOLA e DESLCA, aplicado no conjunto inicial de classificadores formados por <i>bagging</i>	93
4.45	Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA, DESCOLA e DESLCA, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados <i>NIST SD19</i>	94
4.46	Distinções no teste para diferenças entre proporções com os métodos MAJ, KNORA, DESCOLA e DESLCA, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados <i>NIST SD19</i>	94
4.47	Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA, DESCOLA e DESLCA, aplicado no conjunto inicial de classificadores formados por <i>bagging</i> da base de dados <i>NIST SD19</i>	94
4.48	Distinções no teste para diferenças entre proporções com os métodos MAJ, KNORA, DESCOLA e DESLCA, aplicado no conjunto inicial de classificadores formados por <i>bagging</i> da base de dados <i>NIST SD19</i>	94

Lista de Abreviações

<i>k</i> -NN	<i>k</i> vizinhos mais próximos (do inglês, <i>k-Nearest Neighbors</i>)
ASNN	Seleção adaptativa de redes neurais (do inglês, <i>Adaptive Selection of Neural Networks</i>)
CF	Fusão condicional (do inglês, <i>Conditional Fusion</i>)
CU	Uso condicional (do inglês, <i>Conditional Use</i>)
DCS	Seleção dinâmica de classificadores (do inglês, <i>Dynamic Classifier Selection</i>)
DCS-LA	Seleção dinâmica de classificador baseada em acurácia local (do inglês, <i>Dynamic Classifier Selection based on Local Accuracy</i>)
DCS-LCA	Seleção dinâmica de classificador – acurácia de classe local (do inglês, <i>Dynamic Classifier Selection - Local Class Accuracy</i>)
DCS-MC	Seleção dinâmica de classificador – mais competente (do inglês, <i>Dynamic Classifier Selection - Most Competent</i>)
DCS-MCB	Seleção dinâmica de classificador baseada em comportamento de múltiplos classificadores (do inglês, <i>Dynamic Classifier Selection based on Multiple Classifier Behaviour</i>)
DCSDT	Seleção dinâmica de classificador utilizando decisão teórica (do inglês, <i>Dynamic Classifier Selection using Decision-Theoretic</i>)
DE	DESCOLA-ELIMINATE
DES-CS	Seleção dinâmica de classificador – competência com suportes à classe (do inglês, <i>Dynamic Ensemble Selection - competence with weighted Class Supports</i>)
DES-CV	Seleção dinâmica de classificador – competência com voto majoritário ponderado (do inglês, <i>Dynamic Ensemble Selection - Competence with weighted majority Voting</i>)
DESCOLA	Seleção dinâmica de subconjunto utilizando acurácia local geral e acurácia de classe local (do inglês, <i>Dynamic Ensemble Selection using Class and Overall Local Accuracy</i>)

DESLCA	Seleção dinâmica de subconjunto utilizando acurácia de classe local (do inglês, <i>Dynamic Ensemble Selection using Local Class Accuracy</i>)
DOCS	Estratégia dinâmica de superprodução e seleção (do inglês, <i>Dynamic Overproduce-and-Choose Strategy</i>)
DT	Árvore de decisão (do inglês, <i>Decision Tree</i>)
DU	DESCOLA-UNION
DU-U	DESCOLA-UNION-U
FuzzyMLP	<i>Perceptron</i> multicamada com método <i>Fuzzy</i> (do inglês, <i>Fuzzy Multi-Layer Perceptron</i>)
GAES	Seleção de subconjunto de classificadores baseada no método GMDH (do inglês, <i>Classifier Ensemble Selection based on GMDH</i>)
GDES	Seleção dinâmica de subconjunto de classificadores baseada no método GMDH (do inglês, <i>Dynamic Classifier Ensemble Selection based on GMDH</i>)
Genetic	Algoritmo genético para combinar redes neurais
GMDH	Método de agrupamento da teoria de entrega de dados (do inglês, <i>Group Method of Data Handing</i>)
KE	KNORA-ELIMINATE
KLUNORA	Método de seleção baseada em k oráculos mais próximos com informações adicionais de agrupamento
KNN CU	Método de uso condicional dos resultados do k -NN embutido após a execução do algoritmo KNORA (do inglês, <i>K-Nearest Neighbors with Conditional Use</i>)
KNN+KNORA CF	Método de fusão condicional dos resultados do k -NN embutido com os resultados do esquema KNORA após a execução do algoritmo k -NN (do inglês, <i>K-Nearest Neighbors plus K-Nearest-ORAcles with Conditional Fusion</i>)
KNN+KNORA UF	Método de fusão incondicional dos resultados do k -NN embutido com os resultados do KNORA (do inglês, <i>K-Nearest Neighbors plus K-Nearest-ORAcles with Unconditional Fusion</i>)
KNORA	k oráculos mais próximos (do inglês, <i>K-Nearest-ORAcles</i>)
KNORA CU	Método de uso condicional do KNORA após a execução do algoritmo k -NN (do inglês, <i>K-Nearest-ORAcles with Conditional Use</i>)
KNORA+KNN CF	Método de fusão condicional dos resultados do k -NN embutido com os resultados do KNORA após a execução do algoritmo KNORA (do inglês, <i>K-Nearest-ORAcles plus K-Nearest Neighbors with Conditional Fusion</i>)

KU	KNORA-UNION
LA	Acurácia local (do inglês, <i>Local Accuracy</i>)
LCA	Acurácia de classe local (do inglês, <i>Local Class Accuracy</i>)
LMD	Base de dados de músicas latinas (do inglês, <i>Latin Music Database</i>)
MLP	<i>Perceptron</i> multicamada (do inglês, <i>Multi-Layer Perceptron</i>)
MLR	Regressão linear multi resposta (do inglês, <i>Multi-response Linear Regression</i>)
OCS	Estratégia de superprodução e seleção (do inglês, <i>Overproduce-and-Choose Strategy</i>)
OLA	Acurácia local geral (do inglês, <i>Overall Local Accuracy</i>)
SA	Subespaços Aleatórios
SDSC	Seleção Dinâmica de Subconjunto de Classificadores
SOCS	Estratégia estática de superprodução e seleção (do inglês, <i>Static Overproduce-and-Choose Strategy</i>)
Stacking MLR	<i>Stacking</i> com meta classificador MLR
StackingC	<i>Stacking</i> com confidências (do inglês, <i>Stacking with Confidences</i>)

Capítulo 1

Introdução

A classificação é uma das tarefas tratadas na área de reconhecimento de padrões. Sua função é atribuir uma classe¹, dentre várias possíveis, a uma instância de teste² (representada por um vetor de características não visto anteriormente). Para mapear uma classe, deve-se ter um modelo apropriado. Algoritmos de aprendizado de máquina podem ser utilizados para criar um modelo para classificação (também chamado de classificador). Essa criação ocorre de maneira supervisionada, com instâncias de treinamento rotuladas, ou de maneira não supervisionada, com instâncias cujas classes são desconhecidas ou não são levadas em consideração no processo de treinamento.

Gerar um classificador envolve uma série de decisões que afetam seu desempenho. Após optar pelo uso do aprendizado supervisionado ou não supervisionado, escolhe-se uma técnica, dentre diversas, que parece mais promissora ao domínio do problema, baseando-se em pesquisas, estudos e experimentos. Cada técnica tem características e habilidades próprias que a torna mais, ou menos, adequada, para um determinado tipo de aplicação.

Escolhida uma técnica, se a mesma for paramétrica, ainda existem os parâmetros do modelo que devem ser configurados. É comum a execução de testes com diferentes valores para esses parâmetros para a seleção do melhor classificador (aquele cujas configurações levaram a um melhor desempenho de modo global). Essa abordagem pode ser inconveniente para domínios com uma grande variabilidade nas instâncias, para domínios com poucas instâncias diferentes, porém significantes, e para domínios com frequente variação na distribuição. Informações valiosas acabam sendo descartadas.

Um único classificador pode obter bom desempenho em determinados padrões de entrada, mas falhar em outros. Para esses padrões, se fosse selecionado um classificador mais adequado, esses erros poderiam ser amenizados. Isso devido à maneira na qual o aprendizado ocorre na técnica utilizada para a classificação. Na tentativa de melhorar a taxa de reconhecimento, surge o conceito de subconjunto³ de classificadores. Em um subconjunto, classificadores

¹Também é comum o emprego da palavra “rótulo” ou “saída” para designar uma classe.

²Instância de teste também pode ser nomeada instância de entrada, instância a ser classificada, instância a ser reconhecida, instância desconhecida, padrão de entrada, padrão de teste, exemplo de entrada, exemplo de teste ou exemplo a ser reconhecido.

³Subconjunto de classificadores (*classifier ensemble*) é conhecido, na língua portuguesa, como agrupamento de classificadores. No entanto, para não confundir o termo com a técnica de agrupamento (*clustering*) decidiu-se, nesse trabalho, nomeá-lo “subconjunto de classificadores”. Outros termos utilizados são: sistemas multiclassificadores (*Multiple Classifier Systems*), sistemas de multiclassificação, sistemas baseados em múltiplos classificadores, classificador múltiplo, multiestratégia de aprendizado ou comitê (*committees*), dependendo do contexto.

fracos fornecem suas decisões individuais, que são posteriormente combinadas para se definir a classe final.

Um subconjunto de classificadores é retirado de um conjunto inicial⁴ de classificadores candidatos. Os classificadores candidatos são ditos “fracos” porque reconhecem apenas uma parte das instâncias do problema. Classificadores fracos distintos podem classificar corretamente distintas instâncias do domínio de aplicação.

A seleção de classificadores para compor o subconjunto ocorre de maneira estática ou dinâmica. Segundo [Santos *et al.*, 2008], na maneira estática, as regiões de competência dos classificadores são definidas na fase de treinamento, enquanto na dinâmica, essas regiões são estipuladas na fase de classificação. Então, no caso estático, define-se um único subconjunto de classificadores, o qual classificará todas as instâncias de teste. Já no caso dinâmico, para cada nova instância são escolhidos os classificadores que podem ter maiores chances de acertar. Então, esses classificadores são agrupados em um subconjunto, que será responsável pela classificação. A seleção dinâmica pode prover classificadores mais apropriados para cada instância se comparada à seleção estática, uma vez que não está fortemente vinculada às configurações da base de dados utilizada na fase de treinamento.

Selecionar dinamicamente subconjuntos de classificadores é uma tarefa que exige técnicas adequadas para considerar os classificadores mais habilitados a um sucesso na classificação. Na seleção de classificador assume-se a existência de um oráculo que encontra o elemento que possui melhor competência para a instância em questão [Kuncheva & Rodrigues, 2007]. Então, para compor um subconjunto, o oráculo deve ser capaz de optar pelos classificadores com maiores chances de acertar.

O presente trabalho busca encontrar critérios que contribuam para a modelagem do oráculo em subconjuntos de classificadores selecionados de maneira dinâmica. Essa tarefa exige aprofundamento em diferentes aspectos do problema, os quais estão descritos brevemente na Seção 1.1.

1.1 Descrição do Problema e Proposta

A construção de um classificador que obtenha um bom desempenho exige uma base de dados significativa. Mas isso nem sempre é realidade. Uma base de dados que seja representativa, com todas as instâncias possíveis, é complicada de se obter no mundo real. Além disso, é comum o uso de bases de dados pequenas, impossibilitando um aprendizado mais robusto. Uma solução para tal problema é a criação de um conjunto de classificadores fracos.

Os domínios de aplicação de sistemas baseados em múltiplos classificadores frequentemente possuem ao menos um de seus elementos no conjunto inicial que classifica corretamente uma determinada instância de teste. Isso ocorre porque diferentes classificadores reconhecem diferentes pontos do domínio de aplicação. No entanto, muitas vezes não são selecionados os classificadores adequados para compor os subconjuntos, e isso impede que sejam atingidas taxas de reconhecimento mais altas.

⁴A palavra *pool* também é utilizada para designar todos os classificadores candidatos a serem selecionados. Na língua portuguesa, o termo significa “piscina” mas, no caso de sistemas baseados em múltiplos classificadores, é comum o seu uso na língua inglesa. No presente trabalho, o termo “conjunto inicial de classificadores” normalmente será utilizado para designar um *pool*.

O desenvolvimento de uma técnica que selecione dinamicamente classificadores deveria levar em consideração a capacidade que cada classificador tem de reconhecer⁵ a instância de teste. O questionamento que surge aqui é: como saber se um determinado classificador está habilitado a reconhecer uma instância de teste?

Em um sistema baseado em múltiplos classificadores, os classificadores fracos podem ter sido gerados de forma heterogênea (com diferentes algoritmos de aprendizado de máquina), de forma homogênea (com o mesmo algoritmo de aprendizado de máquina, porém alterando-se parâmetros de entrada desses algoritmos), ou de forma mista (combinando a forma homogênea com a heterogênea). Esse trabalho concentrar-se-á em classificadores gerados de forma homogênea, embora não seja objetivo limitar o problema a esse tipo de classificadores. Assim, outras perguntas podem ser feitas:

- Quais características poderiam ser importantes na escolha do classificador?
- Considerar a capacidade que cada classificador tem em reconhecer cada classe auxiliaria o processo?
- Quais informações adicionais ou de contexto poderiam ser utilizadas durante a seleção dos classificadores para uma dada instância de teste?

Um método de seleção dinâmica de subconjunto de classificadores (SDSC) é o KNORA [Ko *et al.*, 2008]. O método KNORA analisa a capacidade que os classificadores têm de reconhecer as instâncias de uma base de validação, instâncias essas consideradas parecidas com a instância de teste. Os valores de distância calculados entre os atributos da instância de teste e os atributos de cada uma das instâncias de validação são que determinam quais são as k instâncias “vizinhas”. Um dos esquemas do KNORA, denominado KNORA-ELIMINATE, seleciona apenas os classificadores que reconhecem o maior número de vizinhos entre os k disponíveis. Já um outro esquema, denominado KNORA-UNION, seleciona todos os classificadores que possuem ao menos um reconhecimento na vizinhança. No trabalho de [Ko *et al.*, 2008], as taxas de reconhecimento obtidas com o KNORA mostraram-se superiores às outras técnicas avaliadas. Tanto é que vários novos trabalhos na literatura que abordam o tema de seleção de subconjunto, como [Xiao & He, 2009] [Woloszynski & Kurzynski, 2010] [Woloszynski & Kurzynski, 2011], comparam seus resultados com os resultados do método KNORA.

Abordando todos esses questionamentos e o trabalho citado, um tópico que parece promissor diz respeito à análise de comportamento dos classificadores na vizinhança. O KNORA utilizou a habilidade que cada classificador fraco tem em reconhecer as instâncias consideradas vizinhas. Porém, antes de propor uma nova abordagem do comportamento local dos classificadores, faz-se necessário analisar o efeito no desempenho que pequenas adaptações no método KNORA podem causar.

O cálculo da distância delimita os vizinhos, que por sua vez influenciam na escolha dos classificadores. Assim sendo, como primeira adaptação seria útil testar diferentes medidas de distância no método KNORA. Mais adaptações que poderiam ser feitas englobam o uso de informações adicionais no método. A primeira informação adicional diz respeito à frequência das classes em agrupamentos. Assim, os votos dos classificadores seriam influenciados pelas

⁵Reconhecer quer dizer acertar a classe.

classes presentes no agrupamento ao qual a instância a ser reconhecida pertence. Outra informação adicional seria a aptidão dos classificadores no reconhecimento de cada classe (taxa de acerto em cada uma das classes). Um classificador que é apto para uma determinada classe deve ter seu voto com um peso maior para essa classe. O KNORA localiza seus vizinhos e apenas os utiliza para verificar quais classificadores os reconhecem. No entanto, esses vizinhos também possuem uma classe, que pode ser aproveitada no processo de classificação. Um novo método poderia então utilizar de maneira complementar essas duas informações: classes dos vizinhos com classes dos classificadores selecionados.

Por fim, uma nova abordagem do comportamento dos classificadores pode ser proposta. Essa abordagem diz respeito à acurácia local, que pode ser tanto geral (somente a taxa de reconhecimento) como pode tratar especificamente das classes reconhecidas.

Sintetizando, esse trabalho trata o problema da seleção dinâmica de subconjunto de classificadores que sejam adequados para cada instância e, para tanto, propõe métodos inspirados no estudo da vizinhança e na acurácia local. Maiores detalhes sobre a proposta podem ser encontrados no Capítulo 3.

1.2 Objetivos

O objetivo principal do presente trabalho é propor uma nova abordagem para SDSC que escolha, a partir de um conjunto inicial de classificadores fracos, um subconjunto de classificadores adequados para classificar uma instância de teste. A seleção deverá basear-se em informações contextuais oriundas da vizinhança da instância de teste em uma base de validação pré-estabelecida.

Assim sendo, os objetivos específicos são:

- a) entender conceitos relacionados a sistemas baseados em múltiplos classificadores;
- b) avaliar o impacto de diferentes adaptações em um método da literatura que utiliza seleção dinâmica de subconjunto de classificadores;
- c) avaliar o uso de informações adicionais durante a seleção de subconjunto de classificadores no sentido de aproximar o desempenho do método proposto àquele atingido pelo seu oráculo.

1.3 Contribuições

O estudo inicial que serviu de base para a proposta de um novo método de SDSC apresentou contribuições importantes, uma vez que respondeu a diferentes questões que ainda estavam em aberto em relação ao comportamento do KNORA, método de seleção dinâmica de subconjuntos de classificadores criado pelo mesmo grupo de pesquisa no qual se insere este trabalho. As contribuições principais deste estudo inicial são listadas na sequência [Vriesmann *et al.*, 2012]:

- Foram implementadas e avaliadas diferentes métricas de similaridade no KNORA. Tais métricas são utilizadas na definição ou escolha da vizinhança da instância de teste na base de validação. Os experimentos demonstraram que pequenas alterações na vizinhança selecionada não resultaram em impacto significativo nos resultados finais de classificação.

- Foi implementada e avaliada uma estratégia que considerou informação adicional gerada a partir do agrupamento da base de validação onde a vizinhança da instância de teste é selecionada.
- Foi implementada e avaliada uma estratégia que considerou o uso da acurácia global por classe dos classificadores do conjunto inicial (taxa de acerto em cada uma das classes) calculada na base de validação, como informação adicional no processo de seleção daqueles elementos que participariam do subconjunto a ser utilizado para classificação de uma determinada instância de teste.

Tais avaliações permitiram um maior entendimento do comportamento do oráculo (que deve ser capaz de escolher os classificadores com melhor competência para classificar uma determinada instância) e de como seria possível contribuir com uma nova abordagem. Além destas contribuições, destacam-se:

- a) implementação e avaliação de diferentes estratégias que combinam informações do k -NN embutido no KNORA com os resultados finais deste método [Vriesmann *et al.*, 2010];
- b) implementação e avaliação de proposta de uma nova abordagem de SDSC que combina a acurácia local, geral e por classe, dos classificadores, obtidas na vizinhança da base de validação;
- c) implementação e avaliação de proposta de SDSC que considera apenas a acurácia de classe local.

As contribuições científicas apontadas anteriormente podem viabilizar contribuições tecnológicas e de inovações no que diz respeito a possível criação de produtos voltados ao reconhecimento de padrões, tais como OCR, classificação de gêneros musicais, expressões faciais, dentre outros.

1.4 Estrutura do Trabalho

O presente documento está organizado em 5 capítulos.

No Capítulo 2 é feita uma revisão sobre sistemas baseados em múltiplos classificadores, sobre a criação de conjuntos e sobre o processo de seleção de classificadores. Além disso, são citados alguns trabalhos relacionados. Seu objetivo, portanto, é propiciar um entendimento da área e mostrar o que tem sido feito, mais especificamente em relação à SDSC.

Após isso pode-se observar a proposta desse trabalho para seleção dinâmica de subconjuntos de classificadores no Capítulo 3. Os resultados e a discussão são apresentados no Capítulo 4. Finalmente, o Capítulo 5 conclui o trabalho.

Capítulo 2

Estado da Arte

Um dos principais problemas em aprendizado de máquina, segundo [Dietterich, 2000], é o estudo de métodos que propiciem a construção de bons conjuntos de classificadores. A razão para o interesse crescente é que muitos experimentos que utilizam conjunto de classificadores apresentam um melhor desempenho em termos de taxas de reconhecimento do que aqueles que usam um único classificador.

Mas por que conjuntos de classificadores frequentemente saem-se melhores do que classificadores individuais? De acordo com [Hansen & Salamon, 1990], existe uma condição necessária e suficiente para que conjuntos de classificadores tenham melhor desempenho que seus elementos individuais: os classificadores do conjunto devem ser acurados (com um nível de precisão superior a 50%) e diversificados. Como normalmente essa condição está presente nos conjuntos, são observados incrementos no reconhecimento. A Seção 2.1 trata desse assunto mais detalhadamente.

A garantia de acurácia e diversidade nos classificadores nos conjuntos está fortemente ligada a duas fases: a fase de construção dos classificadores que poderão participar dos conjuntos e a fase de seleção dos subconjuntos. O processo de construção (Seção 2.2) envolve diferentes decisões, como a organização dos componentes, as técnicas a serem utilizadas, a maneira como o aprendizado ocorrerá, etc. Porém, por melhor que seja a construção dos classificadores, é a fase de seleção dos subconjuntos (Seção 2.3) que vai definir quais que efetivamente participarão da classificação. A seleção é justamente o assunto desse trabalho.

Quando a seleção do subconjunto de classificadores ocorre no momento da classificação é chamada de seleção dinâmica. Assim, visa encontrar o subconjunto mais adequado para a instância a ser reconhecido.

Os trabalhos citados nesse capítulo têm como objetivos apresentar como a seleção dinâmica de subconjunto de classificadores têm sido realizada, quais são as propostas e quais são os prós e contras de cada abordagem. Uma discussão geral pode ser encontrada na Seção 2.4. Já a Seção 2.5 finaliza o capítulo, citando os principais pontos que foram abordados.

2.1 Conjunto de Classificadores e Classificadores Individuais

Conjuntos de classificadores podem obter sucesso, segundo [Dietterich, 2000], quando dois fatos acontecem: a taxa de erro de cada classificador individual é inferior a 50% e os erros individuais não estão correlacionados. Isso também é abordado como dilema da diversidade-acurácia [Kuncheva *et al.*, 2002, Kuncheva & Whitaker, 2003, Oliveira, 2008].

Um classificador é acurado se, em novos padrões a serem classificados, sua taxa de erro é menor do que a obtida aleatoriamente [Dietterich, 2000]. Não haveria motivos para utilizar um classificador se aleatoriamente obtém-se maior precisão. Quanto à diversidade de classificadores, pode-se afirmar que ocorre quando os erros não estão relacionados, ou seja, os erros são diferentes nas novas instâncias. Quanto mais parecidos forem os erros, significa que mais parecidos são os classificadores. Sendo idênticos, o processo de classificação não obteria ganhos.

Alguns estudos que utilizaram conjuntos de classificadores serão apresentados nessa seção com o objetivo de salientar que podem obter melhores resultados que classificadores individuais. Apesar dos trabalhos utilizarem a seleção estática de subconjunto, ou seja, todos os classificadores participam da classificação da instância de teste, pode ser notado que mesmo um número pequeno de classificadores, desde que adequados, pode atingir resultados bons se comparados com os resultados obtidos pelos classificadores individuais.

Em [Giacinto & Roli, 1997b] utilizou-se votação majoritária em 182 classificadores base, obtidos por meio de variações de parâmetros de 3 tipos de redes neurais, de *perceptron* multicamada, de funções de base radial, de rede neural probabilística, de classificador Gaussiano e de classificador *k*-NN (*k-Nearest Neighbor*) para classificar imagens de uma área agrícola da cidade de Feltwell, no Reino Unido. O problema tinha 5 classes, e os resultados mostraram que mesmo poucos classificadores (3, por exemplo) tem melhor taxa de reconhecimento que classificadores individuais. O fato de utilizar diferentes técnicas de classificação já auxilia na diversidade do conjunto, uma vez que diferentes técnicas têm habilidades diferenciadas. Quanto à acurácia, é conhecido que as técnicas utilizadas, desde que treinadas, podem atingir bons resultados.

Em [Fauvel *et al.*, 2006], 3 tipos de *bagging*, 3 tipos de *boosting* e 1 tipo de *random forests* compostas de árvores de decisão CART [Breiman *et al.*, 1984] foram usados para classificar terrenos montanhosos do Colorado. Para o estágio de classificação final, cada árvore forneceu um voto para uma entre 10 classes (água, campina, e 8 tipos de árvores). A classe com o maior número de votos era a vencedora. Seu trabalho mostrou-se interessante para bases de dados grandes, e os resultados de 6 dos 7 métodos de conjuntos de classificadores utilizados obtiveram melhoras em relação ao simples uso de CART. Nesse caso, apenas um tipo de algoritmo foi utilizado (CART). Porém, o que proporcionou diversidade no sistema foram os métodos de criação de conjuntos: *bagging*, *boosting* e *random forests*.

De acordo com as exposições, pode-se ver que, além da questão de sistemas baseados em múltiplos classificadores obterem melhores resultados que classificadores individuais, não existe a necessidade de utilizar todos os classificadores disponíveis. Ou seja, poucos classificadores, desde que adequados, podem obter bom desempenho [Giacinto & Roli, 1997b].

2.2 Criação de Conjuntos de Classificadores

A criação de um sistema baseado em múltiplos classificadores¹ pode ser dividida em 3 partes: a organização de seus componentes, os componentes do sistema e a classificação.

A organização de componentes (também chamada de topologia) pode ser feita de maneira modular, sendo que cada classificador fica responsável por uma parte do sistema, ou em modo de conjunto, quando cada classificador é capaz de responder pela mesma tarefa que

¹Essa seção tomou como base alguns itens do trabalho de [Maia, 2008].

os outros. Na organização modular os componentes normalmente são conectados de forma serial (cascata), enquanto na organização em modo de conjunto todos os classificadores têm uma ligação paralela ou redundante. Maiores detalhes podem ser encontrados na Seção 2.2.1.

O bom desempenho do sistema baseado em múltiplos classificadores depende dos elementos do conjunto, ou seja, dos componentes do sistema (Seção 2.2.2). Considerando que os classificadores individualmente tenham uma precisão maior que 50%, e sendo esses classificadores independentes uns dos outros, tem-se uma boa seleção para criação de um conjunto de classificadores [Ranawana & Palade, 2006].

Finalmente, a classificação (Seção 2.2.3) diz respeito à maneira como é obtida a saída (classe final) para cada instância de teste em um sistema baseado em múltiplos classificadores.

2.2.1 Topologia

A topologia de sistemas baseados em múltiplos classificadores, de acordo com [Lu, 1996], é dividida em três categorias: cascata, paralelo e hierárquico. Para a topologia em cascata (Figura 2.1), o resultado da classificação é gerado após passar por uma fila de classificadores. O primeiro classificador recebe o conjunto de dados de entrada, e gera uma saída que será a entrada do segundo classificador, e assim ocorre até chegar ao último classificador que tem a saída com o resultado da classificação. Uma vantagem dessa topologia é que a tarefa é dividida, ficando cada classificador responsável por uma parte do sistema. Em contrapartida, têm como desvantagem o fato dos últimos classificadores acumularem os erros cometidos pelos primeiros classificadores.

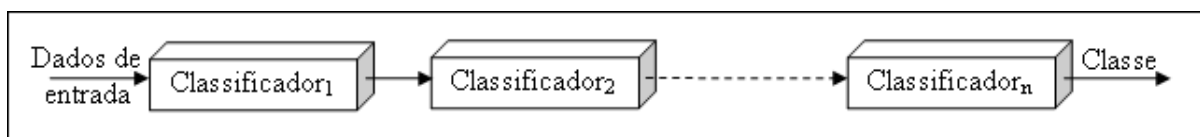


Figura 2.1: Topologia em cascata (baseada em [Ranawana & Palade, 2006])

Na topologia em paralelo (Figura 2.2), a saída dos classificadores é direcionada para um único lugar. Todos os classificadores são treinados para responder pelo mesmo problema. Os erros cometidos por um classificador não são acumulados por outro, como no caso da topologia em cascata. No entanto, deve-se ter uma atenção especial na metodologia da combinação (Seção 2.2.3). Se for selecionada uma estratégia de combinação apropriada, o sistema pode atingir um ótimo desempenho. Contudo, uma seleção inapropriada influenciaria acentuadamente no baixo desempenho do classificador, e afetaria todo o sistema.

A topologia hierárquica (Figura 2.3) combina os classificadores paralelos e os classificadores em cascata para otimizar o desempenho. O uso de tal metodologia pode reduzir a desvantagem encontrada na integração em cascata. Os sistemas hierárquicos podem também ser usados para introduzir a checagem de erros, anulando a influência de classificadores de baixo desempenho [Ranawana & Palade, 2006].

2.2.2 Componentes do Sistema

O conjunto de classificadores pode ser homogêneo ou heterogêneo. No caso de sistemas baseados em múltiplos classificadores homogêneos, são utilizadas as mesmas técnicas

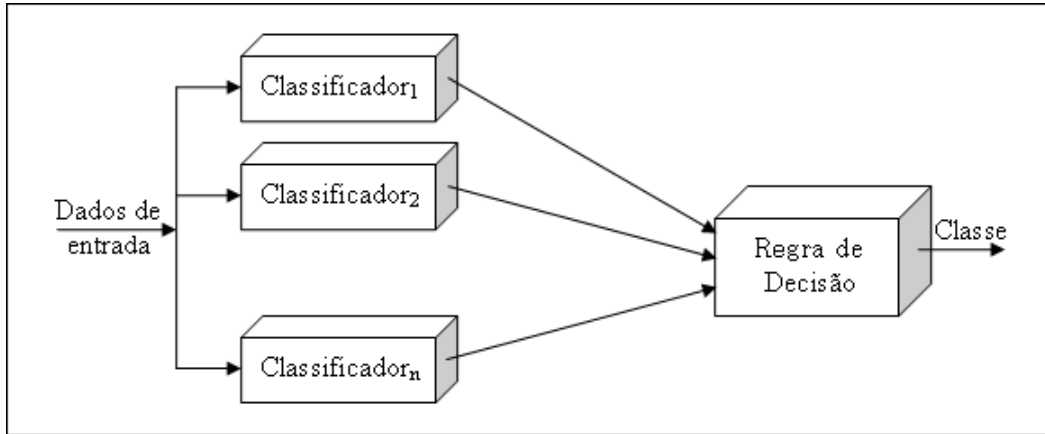


Figura 2.2: Topologia em paralelo (baseada em [Ranawana & Palade, 2006])

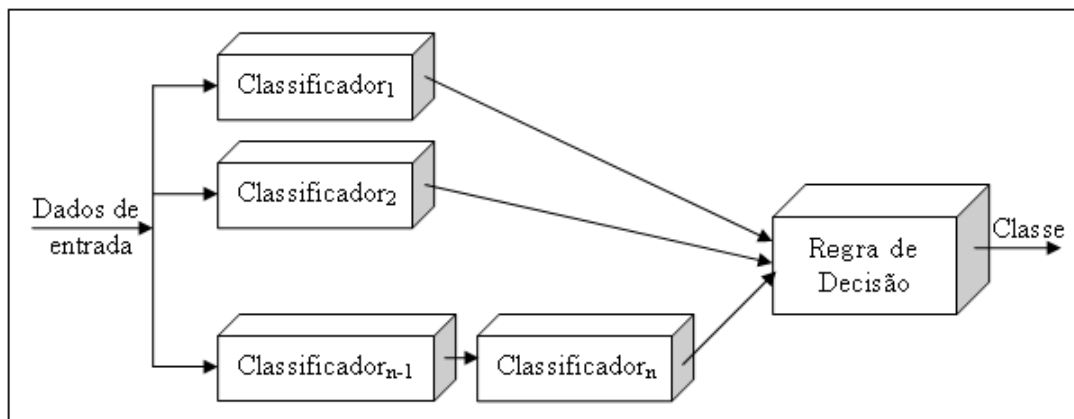


Figura 2.3: Topologia hierárquica (baseada em [Ranawana & Palade, 2006])

(algoritmos base²) para gerar todos os classificadores, sendo variados alguns parâmetros do modelo, como o vetor de características. Já para os sistemas baseados em múltiplos classificadores heterogêneos, utilizam-se diferentes algoritmos base para gerar os classificadores. Métodos utilizados para a forma homogênea e métodos utilizados para a forma heterogênea de criação de classificadores para conjuntos são citados na sequência.

Sistemas Baseados em Múltiplos Classificadores Homogêneos

Os sistemas baseados em múltiplos classificadores homogêneos utilizam os mesmos algoritmos de aprendizado de máquina, porém têm o conjunto de dados ou parâmetros de entrada alterados. Para geração dos classificadores homogêneos, merecem destaque algumas técnicas, as quais são apresentadas na sequência:

- *Bagging*: O método *bagging* (*Bootstrap AGGreatING*) foi proposto por [Breiman, 1996], baseado na amostragem *bootstrap* [Efron & Tibshirani, 1993]. A partir de um conjunto de dados, são gerados subconjuntos de treinamento diferentes. Com esses subconjuntos são

²Um algoritmo base é o algoritmo utilizado para treinar cada um dos elementos do sistema baseado em múltiplos classificadores. Também é denominado algoritmo fraco (*weak classifier*).

criados classificadores diferentes, como representado na Figura 2.4. Cada subconjunto contém o mesmo número de instâncias que o conjunto de dados de entrada. A diversidade entre os conjuntos de dados se encontra no fato de que algumas instâncias do conjunto de dados de entrada podem aparecer mais de uma vez em um subconjunto, ou até não serem selecionadas.

O algoritmo *bagging* geralmente é utilizado para aumentar o desempenho de algoritmos de aprendizado de máquina instáveis. Os algoritmos de aprendizado de máquina instáveis são aqueles que para qualquer variação do conjunto de dados da entrada se têm uma grande variação na saída [Breiman, 1996].

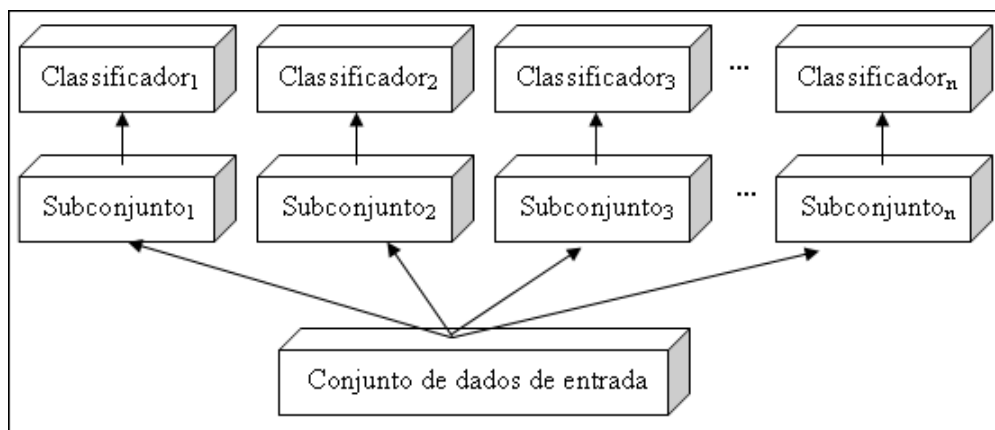


Figura 2.4: Representação do *bagging* (baseada em [Bittencourt, 2005])

- *Boosting*: O *boosting* utiliza o mesmo algoritmo de aprendizado de máquina, com diferentes distribuições do conjunto de treinamento, para combinar as saídas dos classificadores. A alteração na distribuição dos dados do conjunto de treinamento é baseada na análise de erros cometidos pelo classificador anterior. Assim, o *boosting* tenta melhorar o desempenho de cada novo classificador. O método *boosting* pode ser observado na Figura 2.5.

Teoricamente, a técnica de *boosting* pode ser utilizada em algoritmos ditos “fracos”, para corrigir erros.

- Subespaços aleatórios (*random subspace*): A técnica de combinação deste método (Figura 2.6) foi proposta por [Ho, 1998]. Subespaços aleatórios utiliza amostragens dos dados de treinamento configurada por um espaço aleatório de características (atributos). Então são construídos classificadores nos subespaços, agregando-os posteriormente na regra de decisão final.

Na Figura 2.6, das n características originais da base de dados, foram selecionadas aleatoriamente m características para construir cada um dos classificadores (cada subespaço). Ao todo, foram criados z classificadores para fazer parte do conjunto inicial.

Subespaços aleatórios pode ser utilizado tanto para construir classificadores como para agregar classificadores. Uma maneira de resolver um problema com a quantidade de objetos de treinamento relativamente pequena comparada com a dimensionalidade dos dados é construindo classificadores por esse método, pois o treinamento do tamanho da

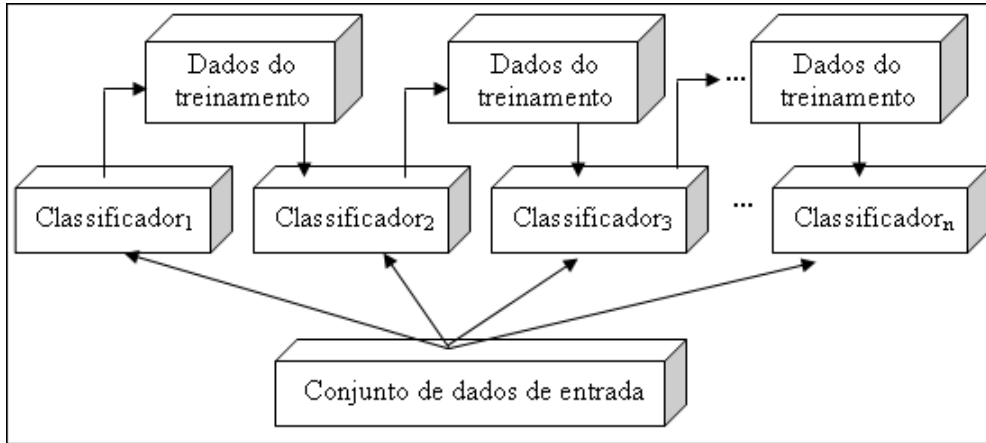


Figura 2.5: Representação do *boosting* (baseada em [Bittencourt, 2005])

amostragem melhora em subespaços. Também quando o dado tem muitas características redundantes, pode-se obter classificadores melhores em um subespaço aleatório do que no espaço de características original [Skurichina & Duin, 2001].

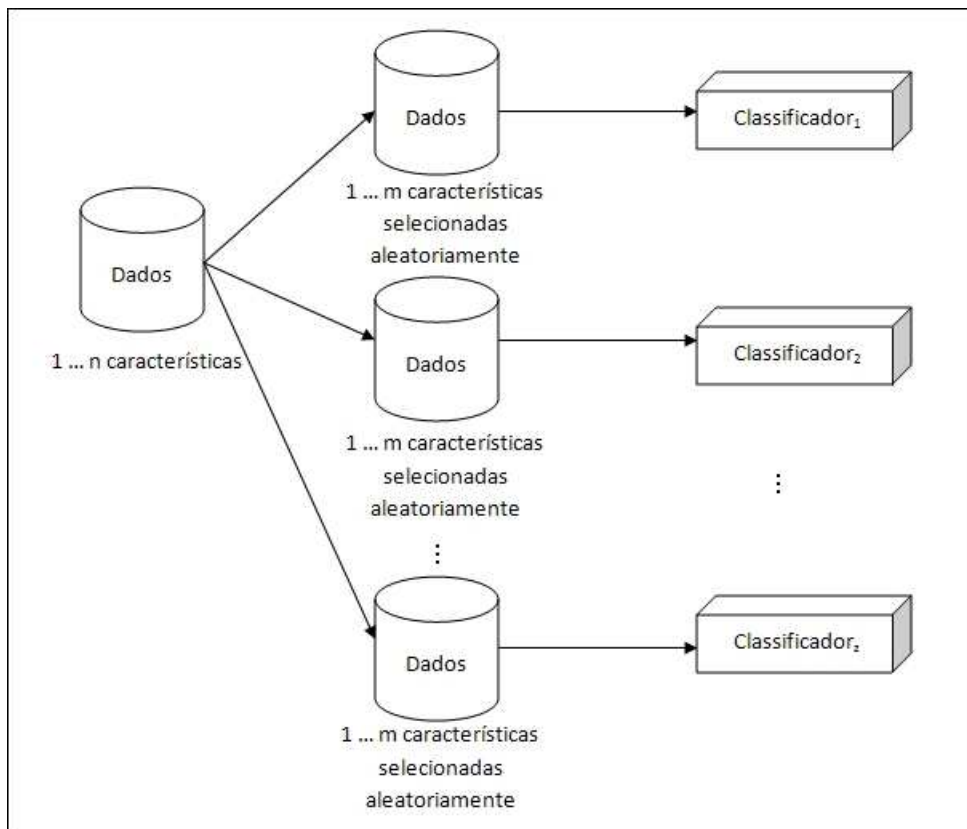


Figura 2.6: Representação de subespaços aleatórios (ilustração baseada em [Maia, 2008])

Sistemas Baseados em Múltiplos Classificadores Heterogêneos

Os sistemas baseados em múltiplos classificadores heterogêneos utilizam algoritmos de aprendizados diferentes para gerar uma combinação. Alguns métodos que se destacam são listados a seguir:

- *Stacking*: Também conhecido em inglês como *stacked generalization* [Wolpert, 1992] (generalização empilhada), considera que um classificador é mais confiável que outro. O método *stacking* tenta aprender qual é o classificador mais confiável e qual a melhor maneira para combinar as suas saídas [Witten & Frank, 2005]. Pode-se observar na Figura 2.7 que o *stacking* combina a saída de vários classificadores, que são formados por diferentes algoritmos de aprendizado de máquina, e estas saídas são combinadas por um meta classificador, o qual gera as saídas com um melhor desempenho.

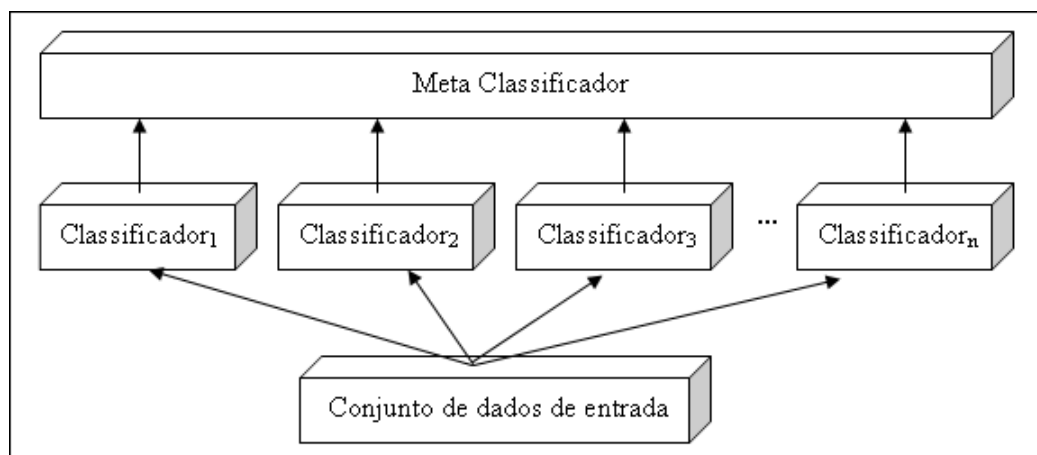


Figura 2.7: Representação do *stacking* (ilustração de [Maia, 2008])

- *StackingC*: É uma variação do algoritmo *stacking*. O método *stackingC* (*stacking with Confidences* - generalização empilhada com confidências) [Seewald, 2002] remove os atributos que não são relevantes e reduz a dimensionalidade da base de dados antes de utilizá-la como entrada para o meta classificador. Foi proposto motivado pelo fraco desempenho em problemas com mais de 2 classes do método *stacking MLR* [Ting & Witten, 1999], o qual é um tipo de *stacking* que utiliza como meta classificador a técnica MLR (*Multi-response Linear Regression* - regressão linear multi resposta) para fornecer um modelo distinto para cada classe. No algoritmo *stackingC*, durante o treinamento e o teste, utiliza-se somente a distribuição parcial de probabilidade da classe que está associada a cada modelo linear.

Quando a base de dados tem atributos irrelevantes e mais que duas classes, segundo [Seewald, 2003], o *stackingC* tende a ser melhor que o *stacking* em termos de precisão e de tempo de treinamento.

De acordo com [Paradedda, 2007], tanto o algoritmo *stacking* como o algoritmo *stackingC* têm dificuldades na definição do meta classificador mais apropriado, de quais e de quantos classificadores base devem ser utilizados.

2.2.3 Classificação

A saída (classe final) de um conjunto pode ser obtida por três métodos distintos: baseado em seleção, baseado em combinação e métodos híbridos. No método baseado em seleção, considera-se que cada classificador tem conhecimento em diferentes áreas. A seleção dinâmica de classificador (DCS, de *Dynamic Classifier Selection*) [Woods *et al.*, 1997] é um dos métodos baseados em seleção. A DCS pode utilizar uma seleção *A priori* ou *A posteriori* para fazer uma análise local, que determinará qual classificador será usado para uma entrada específica.

Para os métodos baseados em combinação, assume-se que todos os classificadores têm o mesmo conhecimento em todas as áreas. De acordo [Paradedda, 2007], esses métodos podem ser classificados em 4 tipos, os quais são apresentados na sequência:

- Métodos de combinação linear: As saídas fornecidas pelos classificadores são combinadas linearmente. Como exemplos de métodos de combinação linear têm-se a soma [Kittler & Alkoot, 2003], a média [Kuncheva, 2002b], a mediana [Kuncheva, 2002b], etc.
- Métodos de combinação não-linear: As saídas dos classificadores são combinadas de forma não linear, como o que ocorre no uso do voto majoritário [Stefano *et al.*, 2002] e na combinação baseada em ordenação (*ranking*).
- Métodos baseados em estatísticas: Utilizam-se probabilidades ou combinações estatísticas nas saídas geradas pelos classificadores. O método de combinação bayesiana [Davis *et al.*, 2004] e a técnica Dempster-Shafer [Mitchell, 1997] são alguns exemplos.
- Métodos baseados em inteligência computacional: Técnicas de inteligência computacional são usadas para combinar as saídas fornecidas pelos classificadores. Nessa classificação, incluem-se, dentre outros, FuzzyMLP (*Fuzzy Multi-Layer Perceptron*) [Canuto, 2001], redes neurais do tipo MLP (*Multi-Layer Perceptron - perceptron* multicamada) [Xiang *et al.*, 2005] e algoritmos genéticos [Kuncheva, 2004a].

Além desses, também têm-se os métodos híbridos, que utilizam os métodos baseados em seleção e em combinação em conjunto para classificar cada instância de teste [Woods *et al.*, 1997]. Em [Paradedda, 2007] são citados como sistemas híbridos: DCS-LA (*Dynamic Classifier Selection based on Local Accuracy* - seleção dinâmica de classificador baseada em acurácia local) [Giacinto & Roli, 2001], DCSDT (*Dynamic Classifier Selection using Decision-Theoretic* - seleção dinâmica de classificador utilizando decisão teórica) [Kuncheva, 2002a] e DCS-MCB (*Dynamic Classifier Selection based on Multiple Classifier Behaviour* - seleção dinâmica de classificador baseada em comportamento de múltiplos classificadores) [Giacinto & Roli, 2001].

Segundo [Ranawana & Palade, 2006], as estratégias de combinação podem ser classificadas em independente de dados, em implicitamente dependente de dados e em explicitamente dependente de dados. Na estratégia independente de dados, somente as saídas de cada um dos classificadores é utilizada. Não é adicionada mais nenhuma outra informação para fazer a classificação. Muitos dos sistemas de voto, como média, máximo e maioria de votos, são exemplos dessa estratégia. Já na estratégia implicitamente dependente de dados, os combinadores são treinados para maximizar o desempenho global nos dados. São exemplos a média

ponderada [Hashem, 1997] e a composição difusa (*fuzzy composition*) [Gader *et al.*, 1996]. Finalmente, na estratégia explicitamente dependente de dados, a seleção de um classificador ou a seleção de um método de combinação é dependente do subespaço ao qual a instância de teste pertence. Dentro dessa categoria, tem-se a seleção dinâmica de classificadores (DCS, de *Dynamic Classifier Selection*), onde é escolhido apenas um classificador para classificar uma instância.

Assim, é possível fazer combinações entre classificadores pertencentes a um mesmo conjunto (ou subconjunto). A questão da seleção desses subconjuntos é apresentada na Seção 2.3.

2.3 Seleção de Subconjunto de Classificadores

A seleção de subconjunto de classificadores ocorre ou de maneira estática ou de maneira dinâmica, conforme já citado anteriormente. Na maneira estática todas as instâncias de teste terão o mesmo subconjunto de classificadores participando do processo de predição. Esses classificadores são escolhidos em um momento anterior à fase de classificação das instâncias de teste.

Entretanto, como diferentes instâncias de teste são normalmente associados a diferentes dificuldades de classificação, assume-se que melhores resultados podem ser obtidos se forem utilizados diferentes subconjuntos para cada um [Ko *et al.*, 2008]. É isso que é feito com a seleção dinâmica de subconjuntos: os classificadores são selecionados baseando-se em características ou regiões de decisão da instância de teste. Em outras palavras, na seleção dinâmica, cada instância a ser reconhecida terá um subconjunto de classificadores que parece mais apropriado, e a seleção é realizada no momento da classificação.

Na Figura 2.8 pode ser visualizada uma representação da seleção estática (lado esquerdo da figura) e da seleção dinâmica (lado direito da figura) de subconjunto de classificadores. Na figura, no caso da seleção estática, os classificadores 1, 5 e 6 serão utilizados para todas as instâncias de teste. Já onde está sendo representada a seleção dinâmica de subconjunto, os classificadores serão selecionados de acordo com características da instância de teste X . Então, diferentes instâncias poderão ter diferentes classificadores no subconjunto.

Quando se trata de escolher o melhor subconjunto possível de classificadores, surge o conceito de oráculo. Em [Kuncheva & Rodrigues, 2007] o termo é definido como sendo a descoberta do classificador que possui melhor competência para a instância em questão. Selecionando o subconjunto de classificadores mais competentes, maiores seriam as chances de sucesso na taxa de reconhecimento.

Os próximos tópicos apresentam trabalhos que utilizam a seleção dinâmica de subconjunto de classificadores. Na Seção 2.3.1 é exposto um método que escolhe os classificadores baseando-se nos acertos obtidos nos vizinhos da instância de teste. Na Seção 2.3.2 é utilizada uma estratégia que tem uma fase de superprodução de classificadores e uma fase de seleção de dois níveis. A Seção 2.3.3 apresenta uma metodologia para a seleção baseada em acurácia e em diversidade. A Seção 2.3.4 relata a aplicação do método GMDH [Ivakhnenko, 1970] para seleção de subconjunto de classificadores de forma estática e, depois, de forma dinâmica. A Seção 2.3.5 trata de um oráculo aleatório linear e a Seção 2.3.6 expõe uma metodologia de seleção dinâmica de classificador e de seleção dinâmica de subconjunto que utiliza um modelo probabilístico para avaliar a competência de cada classificador.

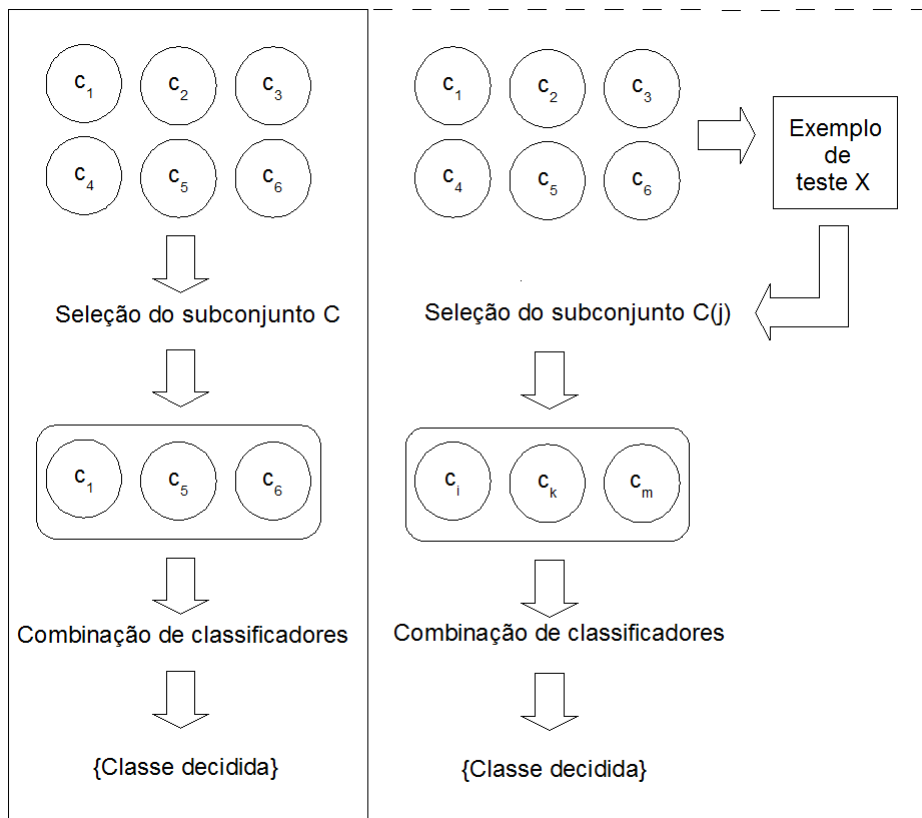


Figura 2.8: Representação da seleção estática e da seleção dinâmica de subconjunto de classificadores (adaptada de [Ko *et al.*, 2008])

2.3.1 Método KNORA

Alguns métodos populares de seleção dinâmica de classificadores³ são a seleção *A Priori*, a seleção *A Posteriori*, a OLA (*Overall Local Accuracy* - acurácia local geral) e a LCA (*Local Class Accuracy* - acurácia de classe local) [Didaci & Giacinto, 2004, Didaci *et al.*, 2005, Giacinto & Roli, 1999, Woods *et al.*, 1997], os quais consideram os vizinhos das instâncias de teste. Em [Ko *et al.*, 2008], faz-se uma comparação entre esses métodos com um novo método, chamado KNORA (*K-Nearest-ORAcles* - k oráculos mais próximos), que diferencia-se dos demais por usar diretamente suas propriedades de escolher um subconjunto de classificadores que melhor classificam seus k vizinhos mais próximos nas instâncias de validação. Então, para participar do processo de classificação de cada instância da base de dados de teste, o KNORA forma um subconjunto com os classificadores que acertam as classes desses k vizinhos.

O método KNORA foi proposto para a seleção dinâmica de subconjunto de classificadores. Supõe que todos os classificadores podem responder pelo mesmo problema, ou seja, utiliza a topologia em paralelo (Figura 2.2, página 10). A saída, então, é direcionada para um único lugar.

Os classificadores foram formados de maneira homogênea (Seção 2.2.2, página 10), ou seja, utilizando os mesmos algoritmos de aprendizado de máquina, porém com alguns parâ-

³A seleção dinâmica de classificadores visa escolher o melhor classificador para cada instância a ser reconhecida. De acordo com as características da instância, é selecionado um único classificador.

metros alterados. Particularmente, em um dos experimentos tratados, os classificadores foram gerados por meio de subespaços aleatórios⁴ sobre a técnica k -NN.

A estratégia de combinação é explicitamente dependente dos dados (Seção 2.2.3, página 15), usando posteriormente combinadores simples.

Quatro diferentes esquemas utilizando o KNORA foram propostos:

- KNORA-ELIMINATE: Dados k vizinhos x_j , $1 \leq j \leq k$, de uma instância de teste X , e sendo $C(j)$, $1 \leq j \leq k$, o subconjunto de classificadores que corretamente classificam todos os seus k vizinhos mais próximos, então cada classificador $c_i \in C(j)$ deve dar um voto no subconjunto de X . No caso de nenhum classificador poder classificar corretamente todos os k vizinhos mais próximos da instância de teste, procure o classificador que mais acertou vizinhos no subconjunto k . Então, somente utilize os classificadores que acertaram o mesmo número de vizinhos que o classificador escolhido. No lado esquerdo da Figura 2.9 pode ser visualizado um hexágono representando a instância de teste X . As instâncias de validação correspondem aos círculos, sendo que os 5 círculos hachurados são as $k=5$ instâncias de validação mais próximas da instância de teste. No lado direito está sendo representada a interseção dos classificadores corretos.
- KNORA-UNION: Dados k vizinhos x_j , $1 \leq j \leq k$ de uma instância de teste X , e suponha que o j -ésimo vizinho mais próximo foi reconhecido pelo subconjunto de classificadores $C(j)$, $1 \leq j \leq k$, então cada classificador $c_i \in C(j)$ deve dar um voto no subconjunto de X . Todos os k vizinhos mais próximos são considerados. Um classificador pode ter mais do que um voto se ele classifica corretamente mais do que um vizinho. Mais vizinhos um classificador classifica corretamente, mais votos esse classificador terá para a instância de teste. No lado esquerdo da Figura 2.10, a instância de teste X está representado como um hexágono e as instâncias de validação como círculos. Os 5 círculos hachurados são as $k=5$ instâncias de validação mais próximas da instância de teste. No lado direito está sendo representada a união dos classificadores corretos.
- KNORA-ELIMINATE-W: O mesmo que KNORA-ELIMINATE, mas cada voto tem o peso da distância entre a instância vizinha x_j e a instância de teste X .
- KNORA-UNION-W: O mesmo que KNORA-UNION, mas cada voto tem o peso da distância entre a instância vizinha x_j e a instância de teste X .

Alguns dos experimentos para avaliar o KNORA foram executados em base de dados (*NIST SD19*) de números manuscritos com 10 classes. Os dados foram gerados pelo método de subespaços aleatórios baseados em k -NN [Cover & Hart, 1967] com $k = 1$. Cada classificador possuiu um subconjunto de 32 características de um total de 132. Foram utilizados 3 subconjuntos da base de dados:

- a) base de dados de treinamento (5.000 amostras), para gerar 100 k -NNs, com $k = 1$;
- b) base de dados de validação (10.000 amostras), usada para selecionar os vizinhos de cada instância de teste e verificar quais classificadores os reconhecem;

⁴O método de subespaços aleatórios (Figura 2.6, página 12) cria diversos classificadores utilizando diferentes subconjuntos de características para classificadores de treinamento. Como o problema é representado em diferentes subespaços, diferentes classificadores desenvolvem diferentes limites para a classificação.

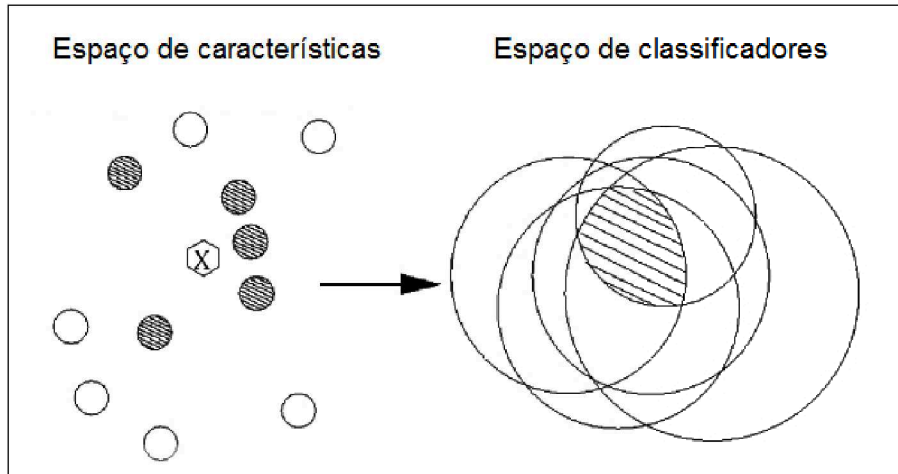


Figura 2.9: KNORA-ELIMINATE (baseado em [Ko *et al.*, 2008])

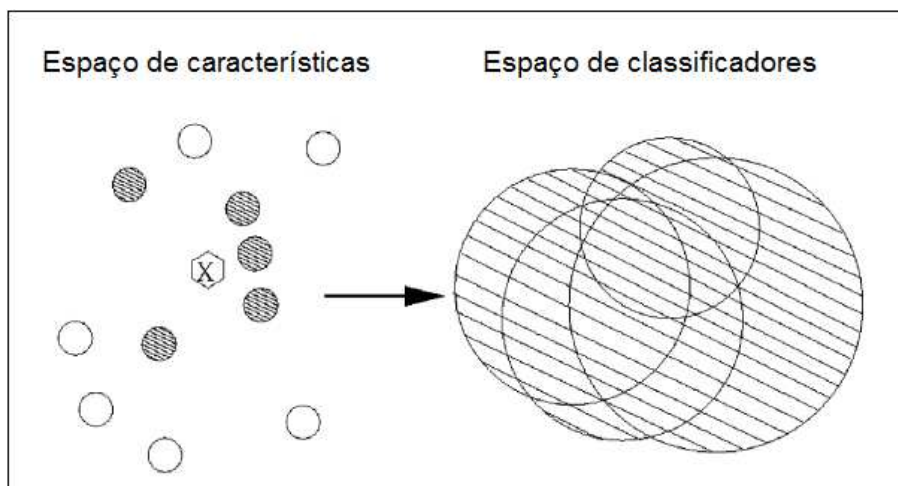


Figura 2.10: KNORA-UNION (baseado em [Ko *et al.*, 2008])

- c) base de dados de teste (60.089 amostras), usada para avaliar a precisão do subconjunto de classificadores.

Na base de dados *NIST SD19*, o KNORA-ELIMINATE e o KNORA-ELIMINATE-W apresentaram os melhores resultados e mostraram um desempenho semelhante mesmo com a variação do valor de k , exceto quando $k=1$. Já o KNORA-UNION e o KNORA-UNION-W não apresentaram resultados tão bons quanto *LCA* e *A Posteriori*. Com relação ao número de amostras, todos os esquemas do KNORA apresentaram um desempenho levemente melhor quanto maior o tamanho das amostras, enquanto os demais métodos mostraram instabilidade. O método que apresentou melhores resultados foi o KNORA-ELIMINATE (97,52% de taxa de acerto), em especial usando $k=7$ e com 76 classificadores, mas ainda encontra-se longe do oráculo (99,95% de taxa de acerto). O oráculo é um processo comumente utilizado para comparação de desempenhos (taxa de reconhecimento) de conjuntos de classificadores. Conforme definido anteriormente, o oráculo atribui a classe correta para uma instância de teste se ao menos um classificador individual do conjunto acerta a instância.

KNORA-UNION e KNORA-UNION-W não mostraram um desempenho tão bom quanto KNORA-ELIMINATE e KNORA-ELIMINATE-W, e perderam desempenho com o aumento de k . Todos os esquemas do KNORA foram visivelmente melhores que os outros métodos para amostras de validação menores. Porém, considerando o tamanho do subconjunto de classificadores, os resultados foram ainda melhores quando estes eram grandes.

Um ponto interessante do KNORA é o fato de utilizar a vizinhança no espaço de características como fator de escolha dos classificadores. Supõe-se que os classificadores que acertam os vizinhos podem acertar também a instância a ser reconhecida. No entanto, ainda não se chegou ao oráculo. Um fato que pode estar contribuindo para isso é que, selecionando os classificadores que acertam as instâncias vizinhas, existe o risco de selecionar classificadores muito parecidos (se não for feita uma avaliação prévia de diversidade), especialmente no KNORA-ELIMINATE. Considerando que subconjuntos de classificadores devem ser acurados e diversificados, o último item não seria cumprido e poderia acontecer de justamente os classificadores selecionados serem aqueles que cometem o mesmo erro na instância de teste.

Sabe-se que originalmente a seleção de subconjunto de classificadores ocorria de forma estática. Então, a diversidade era algo que poderia ser utilizada, objetivando ter um único subconjunto que atendesse a todas as instâncias de teste. Porém, em se tratando de seleção dinâmica, não seria melhor privilegiar mais a acurácia do que a diversidade? Mas, e se no conjunto inicial de classificadores existirem muitos classificadores (parecidos) que classificam corretamente certos vizinhos, mas que cometem o mesmo erro? Uma direção que poderia ser tomada seria avaliar a diversidade no conjunto inicial, descartando classificadores parecidos, para evitar que tenham maior força na votação.

2.3.2 Estratégia Dinâmica de Superprodução e Seleção

Em [Santos *et al.*, 2008] encontra-se um método de seleção dinâmica de subconjunto de classificadores que baseia-se na seleção estática de subconjunto. No método para a seleção estática tratado, chamado de estratégia de superprodução e seleção, assume-se que encontrar o subconjunto de classificadores mais relevante é melhor do que utilizar todo o conjunto inicial de classificadores.

A estratégia de superprodução e seleção (OCS, de *Overproduce-and-Choose Strategy*; ou SOCS, de *Static Overproduce-and-Choose Strategy*) clássica é dividida em duas fases (Figura 2.11): a fase de superprodução, que gera o conjunto inicial de classificadores candidatos C ; e a fase de seleção, que testa diferentes combinações para achar um subconjunto ótimo C_j^* retirado estaticamente do conjunto inicial. O C_j^* é então usado para classificar todas as instâncias de teste.

Uma OCS dinâmica (DOCS) foi proposta para solucionar o problema da escolha fixa do melhor subconjunto de classificadores. Seu esquema pode ser visualizado na Figura 2.12. Inicialmente, um conjunto inicial de classificadores C é criado na fase de superprodução. Foram utilizados 3 conjuntos iniciais de classificadores gerados pelo método de subespaços aleatórios e pelo *bagging*.

Depois, passa-se por uma fase de seleção, que é dividida em dois níveis. No primeiro nível, uma população é escolhida usando um conjunto de dados de otimização e outra população é encontrada usando um conjunto de dados de validação. O processo de otimização é executado por algoritmos genéticos com funções tanto com um objetivo quanto multi-objetivo, guiadas por 5 funções objetivos: 4 medidas de diversidade e a taxa de erro. É gerada uma população de

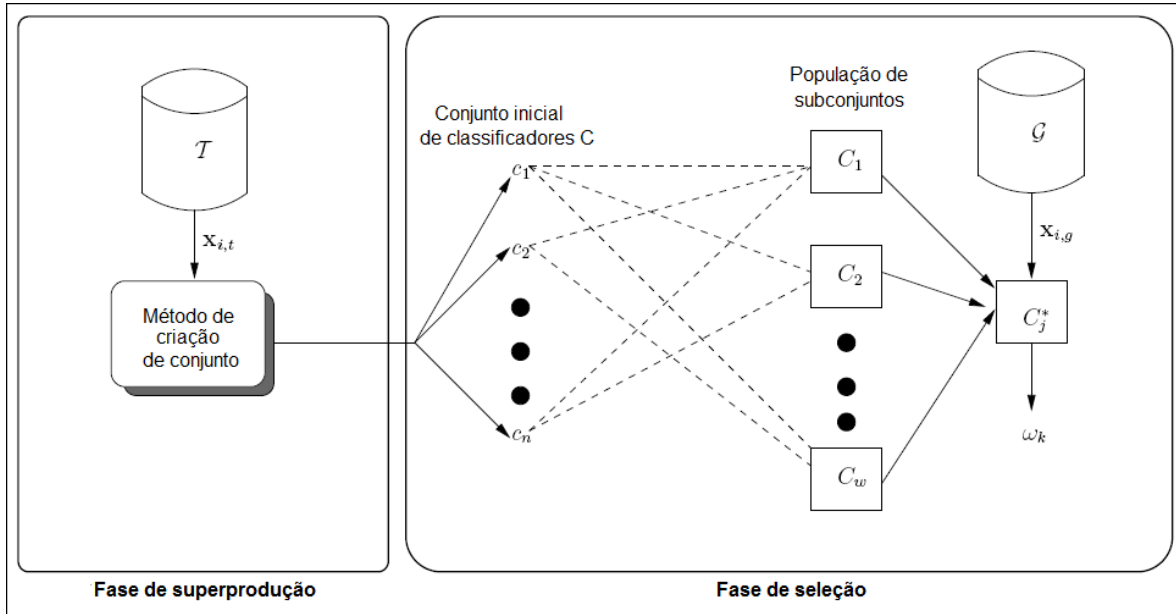


Figura 2.11: Visão geral do processo OCS (baseada em [Santos, 2008])

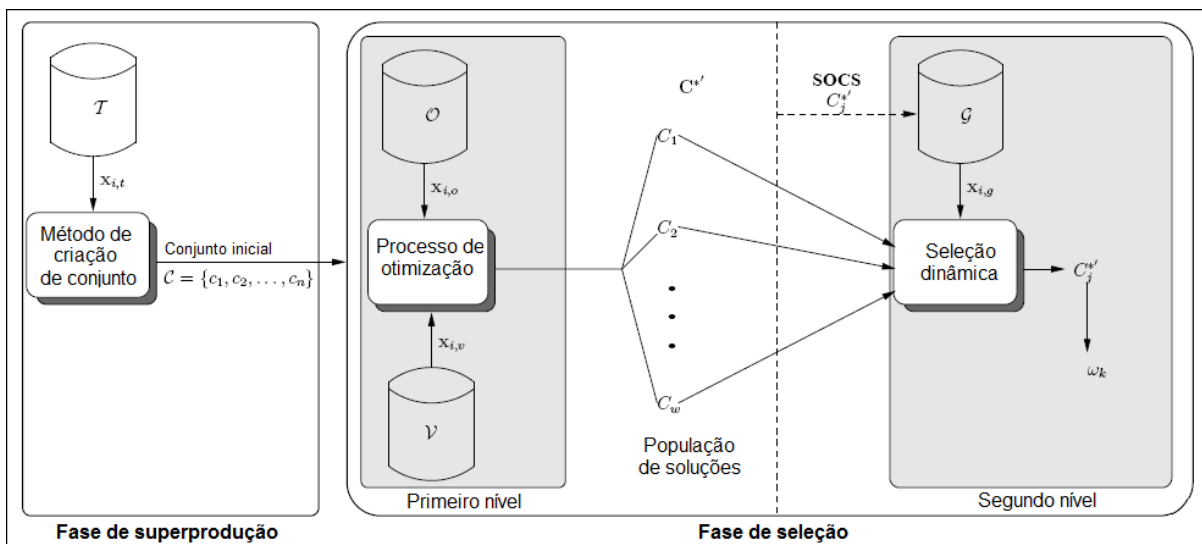


Figura 2.12: Visão geral do processo DOCS (baseada em [Santos, 2008])

conjuntos candidatos $C^{*'}$. No segundo nível, os conjuntos em $C^{*'}$ são considerados na seleção dinâmica para identificar a solução $C_j^{*'}$ mais adequada. O processo de seleção dinâmica do subconjunto de classificadores na DOCS envolve várias condições, as quais estão expostas no Algoritmo 2.1. O $C_j^{*'}$ é então combinado pelo voto da maioria.

Nota-se que o nível de seleção dinâmica utiliza medidas de confiança baseadas no consenso dos subconjuntos candidatos. Foram usadas três estratégias baseadas em confiança: ambiguidade, margem (*margin*) e força (*strength*) em relação à classe mais próxima.

Os seguintes métodos de classificação foram treinados a fim de comparar seus resultados:

- fusão do conjunto inicial de classificadores C pelo voto majoritário;

Algoritmo 2.1: DOCS (baseado em [Santos, 2008])

```

1 Gere um conjunto inicial de classificadores  $C$ .;
2 Utilize um algoritmo de busca para gerar uma população de subconjuntos
  candidatos  $C^{*}$ .;
3 para cada instância de teste  $x_{i,g}$  faça
4   se todos os subconjuntos candidatos concordam a respeito da classe então
5     | atribua a classe a  $x_{i,g}$ ;
6   senão
7     | calcule a confiança das soluções de  $C^{*}$  no nível de seleção dinâmica;
8     | se um subconjunto candidato vencedor é identificado então
9       | selecione o subconjunto candidato  $C_j^{*}$  mais competente para
10      | classificar  $x_{i,g}$ ;
11     | senão
12       | se uma classe com maioria dos votos entre todos os subconjuntos
13       | candidatos com competência igual é identificada então
14         | atribua a classe com maioria dos votos a  $x_{i,g}$ .;
15       | senão
16         | selecione o segundo subconjunto candidato com maior
17         | competência;
18         | se uma classe com maioria de votos entre todos os subconjuntos
19         | candidatos com a primeira e a segunda maior competência é
20         | identificada então
21           | atribua a classe com maioria de votos à instância  $x_{i,g}$ ;
22         | senão
23           | selecione aleatoriamente um subconjunto candidato para
24           | classificar  $x_{i,g}$ .;
25         | fim
26       | fim
27     | fim
28   | fim
29 fim

```

- b) seleção do melhor classificador individual do conjunto inicial C ;
- c) k -NN e árvore de decisão DT treinados com todas as características disponíveis;
- d) DOCS utilizando a melhor estratégia de seleção dinâmica;
- e) oráculo para cada conjunto inicial de classificadores candidatos.

A análise dos resultados mostrou que para os conjuntos de árvore de decisão gerados usando *bagging* e para os conjuntos de árvore de decisão gerados por subespaços aleatórios, DOCS obteve um desempenho melhor que os outros métodos em quase todas as bases de dados, exceto uma. Como segundo melhor método ficou o seu uso com conjuntos de k -NN gerados por subespaços aleatórios.

Os experimentos em 8 problemas reais de classificação mostraram que DOCS se saiu melhor do que OCS estática, do que um método clássico de seleção dinâmica de classificador, e do que a combinação dos votos de todos os classificadores do conjunto inicial.

Comparando os resultados de DOCS com SOCS, com taxa de erros igual a zero, DOCS foi melhor que SOCS e em duas situações a diferença entre ambos os métodos não foi significativa. Analisando a curva de erros obtida com os experimentos não é possível afirmar, entretanto, que DOCS é mais efetivo que SOCS quando aumentado a taxa de rejeição.

Como pontos positivos da DOCS podem-se citar:

- a) incrementos na taxa de reconhecimento (com conjuntos de árvore de decisão gerados usando *bagging* e conjuntos de árvore de decisão gerados por subespaços aleatórios);
- b) a ideia de se utilizar um algoritmo de aprendizado de máquina para formar subconjuntos candidatos à classificação;
- c) o uso de medidas de consenso entre subconjuntos candidatos, o que leva à seleção da solução com maior nível de confiança e permite um aumento do “grau de certeza” da classificação.

A principal desvantagem é a qualidade da população de subconjuntos candidatos obtida no nível de otimização. O oráculo da população de subconjuntos é pior em relação ao oráculo do conjunto inicial de classificadores. Quanto mais próximas as taxas de reconhecimento desses oráculos fossem, melhor seria, pois o subconjunto de classificadores será selecionado dessa população. Isso afeta o desempenho do sistema. Uma das sugestões de trabalho futuro, que é também sugerido em [Santos, 2008], seria o investimento em estratégias para aumentar a qualidade dos subconjuntos candidatos.

2.3.3 Seleção Baseada em Acurácia e Diversidade

Um método de seleção dinâmica de classificadores para construir subconjuntos utilizando precisão (acurácia) e diversidade é tratado em [Santana *et al.*, 2006]. Os classificadores que formam o subconjunto são escolhidos na fase de teste. São apresentadas duas versões diferentes do procedimento de seleção dinâmica. Na primeira, o algoritmo de agrupamento K -médias é utilizado para agrupar padrões de um subconjunto de validação. Para cada agrupamento produzido, colocam-se os classificadores em ordem decrescente de precisão (*rank*) e em

ordem crescente de diversidade. Para colocar os classificadores em ordem crescente de diversidade qualquer medida de diversidade entre pares pode ser usada. Nesse trabalho, os autores utilizam a medida falha dupla (*double fault*). Depois, cada instância de teste é atribuída ao agrupamento com o centroide mais próximo, medido pela distância Euclidiana e são escolhidos os N classificadores mais precisos e, dos N classificadores mais precisos são escolhidos os J classificadores mais diversos ($J < N$) para serem membros do subconjunto. Finalmente, a instância é classificada utilizando um método baseado em combinação.

Na segunda versão de [Santana *et al.*, 2006], um classificador k -NN é construído no conjunto de validação, encontrando os k vizinhos mais próximos da instância de teste. Com base nos k vizinhos, colocam-se os classificadores em ordem decrescente de precisão e em ordem crescente de diversidade. Para colocar os classificadores em ordem crescente de diversidade qualquer medida de diversidade entre pares pode ser usada. Depois, são escolhidos os N classificadores mais precisos dos k vizinhos e, dos N classificadores mais precisos são selecionados os J classificadores mais diversos ($J < N$) para serem membros do subconjunto. Finalmente, a instância é classificada utilizando um método baseado em combinação. Resumindo, a diferença entre as duas versões é que o primeiro usa um método de agrupamento e o segundo usa um classificador k -NN.

Para avaliação de desempenho das versões de [Santana *et al.*, 2006] foram feitas comparações empíricas, onde 4 diferentes métodos de conjuntos foram analisadas: as duas versões propostas (com método de combinação baseado em soma de votos), um método de seleção estática de classificadores (que usa também diversidade e acurácia) e o método de seleção dinâmica de classificadores original (método k -NN para definir os vizinhos da instância de teste, escolhendo o classificador com a melhor precisão nos vizinhos). O conjunto inicial de classificadores teve dois diferentes tamanhos, de 10 e de 15 classificadores. Os classificadores base utilizaram métodos que, no geral, apresentavam diferentes *bias* de aprendizado (*perceptron* multicamadas, *JRIP* (*IREP* otimizado), redes de função de base radial, máquina de vetor de suporte e classificadores bayesianos *Naïve*). As duas versões propostas do método atingiram melhoras, sendo que essa melhora foi estatisticamente significativa na maioria das configurações.

Os resultados apresentados permitiram concluir que levar em conta a diversidade e a acurácia é algo positivo para os conjuntos criados. Houve uma união de conceitos promissores: localização no espaço de características, uso da precisão e da diversidade. Buscou-se, então, similaridades com a instância de teste, levando em conta também as considerações de [Dietterich, 2000]. Porém, apesar de haver uma comparação com um método de seleção dinâmica (k -NN para definir os vizinhos da instância de teste, escolhendo o classificador com a melhor precisão nos vizinhos), seria interessante que houvessem mais comparações com métodos de seleção de subconjunto que não levassem em consideração de forma explícita os itens diversidade e acurácia.

2.3.4 Método GDES

Em [Xiao & He, 2008], o método GMDH [Ivakhnenko, 1970] de agrupamento da teoria de entrega de dados é introduzido na combinação de conjunto de classificadores e é proposto o algoritmo GAES para seleção de subconjunto de classificadores. O algoritmo multicamadas em GMDH constrói uma estrutura de redes neurais com alimentação para frente. Em cada camada constroem-se modelos candidatos através da combinação de dois dos modelos previamente selecionados, e em seguida, utiliza-se o critério externo para avaliar e selecionar os

melhores modelos para entrar a próxima camada. O processo é repetido até chegar ao modelo ideal da teoria de complexidade. Aplicado em sistemas baseados em múltiplos classificadores, o método torna-se hábil para selecionar um subconjunto apropriado do conjunto inicial de classificadores de forma adaptativa, determinando os pesos de combinações entre classificadores base, e completando o processo de combinação automaticamente. Experimentos foram feitos em 16 bases de dados da UCI⁵ e 4 da ELENA⁶. Utilizou-se k vizinhos mais próximos (k -NN) como algoritmo de classificação. Em cada base de dados, foram produzidos 10 classificadores base por 1 -NN e construiu-se o conjunto inicial de classificadores baseado em *bagging*. Na avaliação adotou-se validação cruzada com 10 partes e método reservado. O resultado final foi a média das 10 vezes dos experimentos. Os resultados mostraram que GAES é estatisticamente superior aos métodos bayesiano [Kittler *et al.*, 1998], linear [Benediktsson *et al.*, 1997] e ASNN [Lipnickas & Korbicz, 2004] e que alcançou taxas de reconhecimento comparáveis com os métodos de fusão MAJ [Xu *et al.*, 1992] e genético [Cho, 1999].

Em [Xiao & He, 2009], o GAES [Xiao & He, 2008] é estendido para tratar a seleção dinâmica de subconjunto de classificadores baseada no GMDH [Ivakhnenko, 1970], sendo denominado GDES. Para cada instância de teste, o GDES seleciona um subconjunto de classificadores apropriados do conjunto inicial de classificadores, determina os pesos de combinação entre os classificadores base e completa o processo de combinação automaticamente, de acordo com o Algoritmo 2.2. O GDES foi testado em 6 base de dados da UCI, utilizando também 1 -NN como classificador base, e construindo o conjunto inicial baseado no *bagging*. Os resultados mostraram que seu desempenho foi melhor que o método de fusão MAJ [Xu *et al.*, 1992] e foi levemente melhor que o DCS-LCA [Woods *et al.*, 1997] e o KNORA [Ko *et al.*, 2008].

Algoritmo 2.2: GDES (baseado em [Xiao & He, 2009])

```

1 para cada instância de teste  $x_i^* \in D_2$ ,  $i = 1, 2, \dots, n$  faça
2   Encontre os  $k$  mais próximos vizinhos de  $x^i$  do conjunto de treinamento  $D_1$ ,
   os quais formam um novo conjunto de treinamento  $T = t_1, t_2, \dots, t_k$ ;
3   Utilize  $N$  classificadores base  $L_1, L_2, \dots, L_N$  para classificar o novo conjunto
   de treinamento  $T$ , e a instância de teste  $x_i^*$ , e obtenha os resultados
    $X' = (X'_1, X'_2, \dots, X'_N)$  e  $x' = (x'_1, x'_2, \dots, x'_N)$ ;
4   Execute o algoritmo GAES no novo conjunto  $T$ , recompensando  $X'$  e a
   correspondente variável de classe  $Y$  como entradas do algoritmo, e encontre
   o modelo de combinação do classificador final com complexidade ótima;
5   Coloque os resultados de classificação  $x'$  no modelo de combinação ótimo
   para gerar o resultado final de classificação de  $x_i^*$ ;
6 fim

```

Um item que merece atenção é o critério externo para avaliar os modelos candidatos. Certamente, estudos devem ser feitos para selecionar o melhor critério, uma vez que eles auxiliam de forma direta na adaptação. Em caso de não se ter um bom critério, há uma perda de desempenho.

O ponto positivo do GAES é a sua habilidade para selecionar um subconjunto apropriado do conjunto inicial de classificadores de forma adaptativa, determinando os pesos de

⁵ftp:// ftp.ics.edu/pub/machine-learning-data.

⁶ftp:// ftp.dice.ucl.ac.be/pub/neural-nets/ELENA/databases.

combinações entre classificadores base, e completando o processo de combinação automaticamente. A ideia de utilizar um método que possa selecionar o subconjunto apropriado parece ser bastante promissora.

2.3.5 Oráculo Aleatório Linear

Em [Kuncheva & Rodrigues, 2007] foi proposta uma abordagem, denominada oráculo aleatório linear, que combina a seleção e a fusão para o projeto de subconjunto de classificadores. Cada classificador no conjunto possui um subconjunto com dois classificadores e um oráculo, que é uma função aleatória linear. Para cada instância de teste, o oráculo de um classificador decide qual subclassificador usar. A classe escolhida por cada subclassificador então é submetida à regra da combinação dos resultados do conjunto.

A diferença entre um oráculo aleatório linear e outras seleções de classificadores é que o oráculo não supõe aperfeiçoar nenhum critério, ele apenas serve como um divisor de espaços entre duas partes aleatórias. O oráculo aleatório divide o espaço de características em duas partes, dividindo um problema em dois problemas mais fáceis. A classificação de uma instância de teste é feita por apenas um dos dois subclassificadores, que são treinados em bases de dados diferentes, aumentando a diversidade.

Experimentos foram feitos com 35 bases de dados e 20 modelos de conjuntos para testar a eficiência do oráculo aleatório linear. Árvores de decisão foram usadas como classificadores base e todos os métodos de conjunto foram executados com as mesmas divisões de bases de treinamento-teste com e sem oráculo aleatório linear. Os resultados mostram que os métodos de conjunto com oráculo linear aleatório obtiveram um desempenho melhor que os métodos sem, embora também tenham mostrado que utilizar somente esse método é insuficiente.

A abordagem foi mais útil em conjuntos de *bagging* e em conjuntos de subespaços aleatórios. Os resultados foram confirmados em 7 bases de dados de domínio médico do mundo real.

O trabalho combina uma espécie de seleção estática com seleção dinâmica de subconjuntos. Seleção estática porque cada um dos classificadores terá que participar do processo de classificação, e dinâmica porque pode ser escolhido um de seus dois subclassificadores. Assim, diferentes combinações de subclassificadores acabam sendo geradas no momento da classificação.

Um ponto negativo no oráculo aleatório linear é o fato de, para cada classificador, ser obrigatória a escolha de um dos seus subclassificadores. Dessa maneira, cada classificador terá um subclassificador votando. Então, dado que dois subclassificadores pertencem ao mesmo classificador e que ambos não são adequados para a instância de teste, um deles acabará sendo escolhido, e isso influenciará no desempenho.

Outro aspecto negativo é o fato de funcionar apenas como divisor de espaços entre duas partes aleatórias, não supondo aperfeiçoar nenhum critério.

2.3.6 Modelo Probabilístico de Competência de Classificador

Em [Woloszynski & Kurzynski, 2011], a seleção dinâmica de classificador e a seleção dinâmica de subconjunto de classificador foram tratadas utilizando uma função de competência de classificadores baseada em um modelo probabilístico.

A competência é calculada como sendo a probabilidade de reconhecimento de um classificador de referência randomizado cujos suportes às classes são realizações das variáveis aleatórias. Na média, o classificador de referência randomizado produz o mesmo vetor de suportes à classe que o classificador modelado. Os parâmetros de distribuição de probabilidade são escolhidos utilizando vetores de características em uma base de dados de validação.

A função de competência utilizada foi baseada em modelo de função potencial Gaussiana normalizada ou em regressão com $r = 2$, $r = 3$ e $r = 5$. Três sistemas baseados em seleção dinâmica de classificador e em seleção dinâmica de subconjunto foram construídos para utilizar o modelo probabilístico:

- DCS-MC (*Dynamic Classifier Selection - Most Competent*): O conjunto de competência e a função de competência são construídos para cada classificador do conjunto inicial. Então, é selecionado o classificador mais competente do conjunto inicial para classificar a instância de teste.
- DES-CV (*Dynamic Ensemble Selection - Competence with weighted majority Voting*): A principal diferença entre o DCS-MC e o DES-CV é que este último seleciona todos os classificadores com competência maior que a de uma classificação aleatória para compor um subconjunto, ao invés de selecionar apenas um classificador. Então, os classificadores do subconjunto são combinados utilizando a regra do voto majoritário ponderada, onde os pesos equivalem aos valores de competência. Finalmente, a instância de teste é classificada utilizando a regra do máximo.
- DES-CS (*Dynamic Ensemble Selection - competence with weighted Class Supports*): Possui a mesma metodologia que o DES-CV, excetuando o fato de que é utilizado vetor ponderado de suportes à classe ao invés de usar a regra do voto majoritário ponderada. Novamente, a instância de teste é classificada utilizando a regra do máximo.

Para os experimentos, as bases de dados foram obtidas do repositório de aprendizado de máquina UCI [Asuncion & Newman, 2007], da coleção da Ludmila Kuncheva (LKC) [Kuncheva, 2004b] e do projeto ELENA [ELE, ND]. A avaliação ocorreu em sistemas baseados em múltiplos classificadores criados de forma homogênea e em sistemas baseados em múltiplos classificadores criados de forma heterogênea⁷. Os resultados foram comparados com 8 metodologias de classificação em sistemas baseados em múltiplos classificadores e, entre elas, o KNORA-ELIMINATE [Ko *et al.*, 2008]. O sistema baseado em seleção de subconjunto de classificadores DES-CS com modelo de função potencial para generalizar as competências obteve melhor taxa de reconhecimento que as 8 metodologias comparadas para 22 bases de dados. Essa melhoria, apesar de ser relativamente pequena, apresentou significância estatística.

Observou-se que o modelo de função potencial é o mais acurado para generalizar a competência. No entanto, ele deve ser utilizado para instâncias de dimensão pequena e média. O custo computacional em problemas com instâncias de grande dimensão é muito alto. Nesse caso, segundo [Woloszynski & Kurzynski, 2011], seria melhor utilizar a regressão com $r = 2$ para generalizar a competência.

De uma maneira geral, a seleção dinâmica de subconjunto de classificadores e o uso de vetor ponderado de suportes à classe apresentaram-se como melhor escolha entre os sistemas propostos que utilizam a função de competência de classificadores baseada em um modelo probabilístico.

⁷Para definições de sistemas homogêneos e heterogêneos, ver Seção 2.2.2.

2.4 Discussão

As seções anteriores apresentaram conceitos que conduzem para o entendimento do que é, para que serve, por que funciona e como tem sido empregada a seleção dinâmica de subconjunto de classificadores. Esses classificadores são baseados em características ou regiões de decisão da instância de teste e são utilizados para classificá-lo.

Na literatura cita-se que o sucesso no uso de conjuntos depende da acurácia e da diversidade de classificadores. Obviamente, quando trabalha-se com conjuntos, os classificadores não devem ser idênticos, pois senão bastaria utilizá-los de forma individual. Também é necessário que sua precisão seja melhor do que usar algum resultado aleatório, caso contrário não justificaria o uso de um classificador [Dietterich, 2000]. Pode-se notar que realmente conjuntos de classificadores obtêm resultados melhores do que classificadores individuais, como em [Giacinto & Roli, 1997b], [Fauvel *et al.*, 2006], [Ko *et al.*, 2008], [Santana *et al.*, 2006]. Isso provavelmente porque diferentes classificadores acabam modelando diferentes características do problema.

O primeiro passo para trabalhar com conjunto de classificadores é a criação do conjunto inicial. A maioria dos classificadores é construída para a topologia em paralelo, onde a saída dos classificadores é direcionada para um único lugar. Inclusive, todos os trabalhos citados nesse capítulo eram dessa topologia. Quanto aos componentes do sistema, em [Ko *et al.*, 2008], [Santos *et al.*, 2008], [Xiao & He, 2009] e [Kuncheva & Rodrigues, 2007] foram utilizados sistemas baseados em múltiplos classificadores homogêneos, enquanto que em [Santana *et al.*, 2006] foram utilizados sistemas baseados em múltiplos classificadores heterogêneos. Esses cinco estudos trataram de seleção dinâmica de subconjunto dependendo explicitamente dos dados. Predominantemente, a combinação entre os classificadores foi por meio de maioria de votos, sendo que em alguns casos, havia pesos nessas combinações (como no KNORA-ELIMINATE-W e no KNORA-UNION-W).

Procurou-se por contribuições recentes na área de seleção dinâmica de subconjunto de classificadores, as quais exploraram os seguintes tópicos:

- a) formação de subconjunto com classificadores que acertavam vizinhos no espaço de características da instância a ser classificada, em [Ko *et al.*, 2008];
- b) fase de seleção de 2 níveis, onde no primeiro nível ocorre um processo de otimização por meio de algoritmos genéticos para gerar uma população de subconjuntos candidatos, e no segundo nível ocorre a seleção dinâmica utilizando medidas de confiança baseadas no consenso dos subconjuntos candidatos, em [Santos *et al.*, 2008];
- c) uso de agrupamento (ou de vizinhos) da instância de teste para colocar classificadores em ordem decrescente de precisão e em ordem crescente de diversidade e, depois disso, escolha dos N classificadores mais precisos e, dentre esses, seleção dos J mais diversificados para compor o subconjunto, em [Santana *et al.*, 2006];
- d) adaptação do método GMDH [Ivakhnenko, 1970] de agrupamento da teoria de entrega de dados, em [Xiao & He, 2009];
- e) oráculo aleatório linear como divisor de subclassificadores, em [Kuncheva & Rodrigues, 2007];

- f) uso de um modelo probabilístico para cálculo de competência dos classificadores, em [Woloszynski & Kurzynski, 2011].

Algumas considerações que podem ser feitas, baseando-se nos trabalhos expostos, são apresentadas na sequência:

- Considerar os vizinhos no espaço de características tem atingido bons resultados, uma vez que pode-se obter classificadores que são melhor adaptados para uma determinada região do espaço de características. O oráculo não foi atingido em experimentos de [Ko *et al.*, 2008], o que indica que mais estudos podem ser realizados abordando outros aspectos da vizinhança.
- Propor subconjuntos candidatos de classificadores por meio de uma técnica de aprendizado é algo promissor, porém corre-se o risco de haver diferenças de decremento entre o oráculo do conjunto inicial e o oráculo dos subconjuntos candidatos, como aconteceu em [Santos *et al.*, 2008]. Esforços deveriam ser realizados na tentativa de diminuir essas diferenças.

Subconjuntos candidatos de classificadores acabam limitando as escolhas e, por isso, devem ser bem elaborados. Por outro lado, um aspecto positivo do uso de subconjuntos candidatos é que permite o uso de medidas de consenso, propiciando um aumento do “grau de certeza” da classificação.

- Alguns trabalhos relatam explicitamente que classificadores que compõem conjuntos devem ser precisos e diversificados. A diversidade apresentou-se como fator muito importante porque tradicionalmente a seleção de subconjunto era feita de forma estática, então tinha que englobar classificadores que tivessem habilidades diferentes, para poder atender instâncias diferentes no processo de classificação.

No conjunto inicial é característica obrigatória ter a diversidade, senão não haveria porque utilizar conjuntos. Mas, será que na seleção dinâmica de subconjunto precisa haver tantos esforços no item diversidade?

De qualquer forma, mesmo na seleção dinâmica, poderia haver uma metodologia para verificar a existência de classificadores parecidos e que, portanto, correriam o risco de cometerem os mesmos erros, e pior, ainda privilegiar esses erros (por exemplo, quando a classe final é obtida por meio de maioria de votos, e os classificadores “idênticos” são a maioria).

- Localização no espaço de características, uso da acurácia e da diversidade, como em [Santana *et al.*, 2006] é interessante, pois permite selecionar os classificadores mais precisos nos vizinhos e, ao mesmo tempo, restringem classificadores “idênticos”, que podem cometer os mesmos erros.
- Utilizar na seleção dinâmica técnicas baseadas na seleção estática de subconjunto de classificadores, porém adaptadas, como em [Xiao & He, 2009] e em [Santos *et al.*, 2008], é um caminho que tem sido seguido por alguns estudos, e bons índices de reconhecimento têm sido alcançados. Porém, será que métodos criados especificamente para seleção dinâmica não obteriam melhores resultados?

- A questão de se ter um oráculo como divisor de subclassificadores pode contribuir para a área. No entanto, sugerem-se testes onde não seja aleatório (ou seja, tenha algum significado na divisão), onde possam ser utilizados mais que 2 subclassificadores e onde não seja obrigatório o uso de um subclassificador de cada oráculo.
- O uso da competência dos classificadores calculada com um modelo probabilístico, particularmente a função potencial Gaussiana normalizada, permitiu que a seleção dinâmica de subconjunto de classificadores “competentes” apresentasse incrementos na taxa de reconhecimento em conjuntos homogêneos e conjuntos heterogêneos, em relação à outras técnicas de sistemas baseados em múltiplos classificadores. Esse incremento, embora pequeno, apresentou-se significativo. Para problemas de domínio médico e industrial, segundo [Woloszynski & Kurzynski, 2011], onde pequenos erros podem ser cruciais, o uso da técnica pode ser justificada. Porém, a técnica com esse incremento na taxa de reconhecimento foi justamente aquela com maior custo computacional, o que dificulta o seu uso em problemas de grande dimensão. Será que alguma outra metodologia também não poderia ter esse incremento com menos custo computacional?

O uso de seleção dinâmica de subconjunto de classificadores tem obtido melhor taxa de reconhecimento quando comparadas com outras metodologias de classificação. Pode-se verificar que a seleção dinâmica de subconjunto de classificadores é algo promissor, uma vez que tenta encontrar os classificadores mais adaptados para cada instância a ser classificado. No entanto, não foram encontrados experimentos que relatassem que atingiram o oráculo do conjunto inicial de classificadores. Observa-se que é um campo aberto para pesquisas no sentido de propostas de novos métodos, visando selecionar subconjuntos com maiores chances de acertos. Resumindo, pode-se concluir que a proposta de um método para seleção dinâmica de subconjunto de classificadores é de interesse na comunidade científica.

2.5 Conclusão

Nesse capítulo foi realizada uma revisão sobre conjuntos de classificadores e foram citados trabalhos relacionados ao presente estudo, que visa propor um método para seleção dinâmica de subconjuntos com base na modelagem do oráculo. Primeiramente, foram relatadas condições que se fazem necessárias (acurácia e diversidade) para que um conjunto de classificadores funcione bem e o fato de que é possível obter melhorias, se comparado com classificadores individuais, mesmo utilizando poucos elementos no conjunto.

Posteriormente, conceitos sobre a criação de sistemas baseados em múltiplos classificadores, particularmente sobre topologia, sobre composição do sistema e sobre obtenção das saídas; e também sobre a seleção de subconjunto foram citados. Na parte de seleção de subconjunto foi tratada a seleção dinâmica de subconjunto de classificadores, onde trabalhos relacionados foram apresentados e discutidos.

Os itens expostos forneceram uma base teórica sobre como são formados e utilizados conjuntos de classificadores, especialmente no quesito seleção dinâmica, para então permitir o norteamento no tema desse trabalho, cuja proposta está descrita no Capítulo 3.

Capítulo 3

Métodos Propostos

A busca por uma nova abordagem para a seleção dinâmica de subconjunto de classificadores teve início com a implementação e avaliação do método KNORA [Ko *et al.*, 2008]. Com o objetivo de entender o funcionamento do KNORA e responder a questões ainda em aberto foram realizados experimentos considerando adaptações que figuravam como promissoras na tarefa de aproximar o desempenho do KNORA ao de seu oráculo para diferentes aplicações (Seção 3.1). Dentre as adaptações avaliadas, encontram-se descritas neste capítulo a tentativa de utilizar outras medidas de similaridade no KNORA, em substituição à distância Euclidiana (Seção 3.1.1); e a consideração de informações adicionais para a seleção dos conjuntos de classificadores com base no uso de técnica de agrupamento sobre a base de validação (Seção 3.1.2), ou ainda, com base na criação de uma lista ordenada dos classificadores disponíveis no conjunto inicial por ordem de desempenho no reconhecimento de cada classe (Seção 3.1.3).

Estas adaptações no KNORA e os respectivos experimentos realizados (ver Capítulo 4, na Seção 4.4) propiciaram a base para a proposta de três novas abordagens para a seleção dinâmica de subconjunto de classificadores. A primeira pode ser caracterizada como uma adaptação do KNORA e consiste na fusão das classes fornecidas pelo k -NN embutido no método com as classes fornecidas pelos classificadores escolhidos (Seção 3.2). Já a segunda consiste em uma nova abordagem que combina OLA (*Overall Local Accuracy*) e LCA (*Local Class Accuracy*) em duas fases para a seleção de um subconjunto de classificadores (Seção 3.3). Finalmente, a terceira abordagem é uma modificação da segunda e utiliza apenas a fase de LCA (Seção 3.3).

Por fim, os principais pontos desse capítulo são salientados na Seção 3.4.

3.1 Abordagens Preliminares

Em [Ko *et al.*, 2008] foi introduzido, como já citado na Seção 2.3.1 (página 16), o método KNORA (k oráculos mais próximos). Nesse método, os classificadores selecionados para compor o subconjunto são aqueles que reconhecem os vizinhos (na base de dados de validação) da instância de teste. Esse reconhecimento pode englobar todos os vizinhos (ou o maior número possível dentre as k instâncias consideradas parecidas) no caso do KNORA-ELIMINATE, ou pelo menos um vizinho no caso do KNORA-UNION. Apesar desse critério de seleção ter apresentado bons resultados, o limite superior possível de precisão de classificação (oráculo) não foi atingido, ou seja, a proporção de instâncias que são classificadas corretamente por pelo menos

um classificador do conjunto inicial é superior à taxa de reconhecimento obtida. Existe, então, uma margem de precisão que ainda pode ser incrementada.

Assim, primeiramente, será investigada a alteração da métrica de cálculo de distância entre os vizinhos (Seção 3.1.1) com a finalidade de verificar o quanto essa métrica pode influenciar na taxa de reconhecimento de um método de seleção dinâmica de subconjunto de classificadores que seja baseado em vizinhança.

Já em [Santana *et al.*, 2006] (página 22) foram utilizados tanto a vizinhança quanto o agrupamento na seleção de subconjunto baseada em acurácia e diversidade. Porém, como citado na Seção 2.3.3, houve duas versões distintas do método, não existindo o uso simultâneo de informações obtidas pelos dois algoritmos. É justamente do uso de informações obtidas pelo agrupamento juntamente com as obtidas por oráculos mais próximos que tratará a Seção 3.1.2.

Novamente no KNORA, duas estratégias (KNORA-ELIMINATE-W e KNORA-UNION-W) de [Ko *et al.*, 2008] utilizavam o voto ponderado pela distância entre a instância de teste e a instância de validação reconhecida pelo classificador. Isso permitiu incremento na taxa de reconhecimento no caso do KNORA-UNION-W em relação ao KNORA-UNION. Assim observa-se que votos ponderados podem auxiliar no processo de classificação em sistemas baseados em múltiplos classificadores. Então a Seção 3.1.3 proporá o uso da proporção de classe reconhecida por classificador como informação adicional nos subconjuntos selecionados.

As próximas seções tratam dessas 3 adaptações no método KNORA almejando entender o impacto de pequenas mudanças na tentativa de se aproximar do oráculo.

3.1.1 Adaptação de Novas Métricas de Distância ao Método KNORA

O método KNORA [Ko *et al.*, 2008] utiliza a distância Euclidiana para encontrar os k vizinhos, os quais contribuem na seleção dos classificadores que participarão do conjunto. No caso do KNORA-ELIMINATE, os classificadores que acertam todos os k vizinhos (ou o maior número possível de vizinhos dentre os k) têm direito a um voto. Já no KNORA-UNION, para cada vizinho que o classificador acerta, ele tem direito a um voto para a classe da instância a ser reconhecida. Como nota-se, a seleção dos vizinhos é algo fundamental no método.

Diferentes medidas de distância poderiam trazer diferentes vizinhos e, consequentemente, o conjunto de classificadores a ser selecionado poderia ser diferente também. Talvez isso possa contribuir para selecionar um subconjunto mais próximo do oráculo. Assim, aconselha-se que sejam testadas diferentes medidas de distância para selecionar os vizinhos [Vriesmann *et al.*, 2012]. Entre elas, sugere-se a distância de Canberra (Seção 3.1.1), a distância de Cosine (Seção 3.1.1) e a distância de Pearson (Seção 3.1.1), que são medidas comumente tratadas na literatura.

Canberra

A distância de Canberra (d_C) entre dois elementos, A e B, pode ser expressa pela Equação 3.1.

$$d_C(A, B) = \sum_{i=1}^n \frac{|a_i - b_i|}{|a_i| + |b_i|}, \quad (3.1)$$

onde A é uma instância de teste, B é uma instância de validação, n é o número de características da instância, a_i representa a i -ésima característica da instância de teste, e b_i representa a i -ésima característica da instância de validação.

Cosine

A distância de Cosine calcula o ângulo formado entre a instância de teste e seu vizinho, sendo representada como na Equação 3.2.

$$d_{COS}(A, B) = 1 - \frac{\sum_{i=1}^n a_i \cdot b_i}{\sqrt{\sum_{i=1}^n a_i^2 \cdot \sum_{i=1}^n b_i^2}}, \quad (3.2)$$

onde A é uma instância de teste, B é uma instância de validação, n é o número de características da instância, a_i representa a i -ésima característica da instância de teste, e b_i representa a i -ésima característica da instância de validação.

Pearson

A distância de Pearson é derivada do coeficiente de correlação de Pearson e é medida pela Equação 3.3.

$$d_{pcc}(A, B) = 1 - |p|, \quad (3.3)$$

onde $|p|$ é o coeficiente de correlação de Pearson definido pela Equação 3.4.

$$p = \frac{n \cdot \sum_{i=1}^n (a_i \cdot b_i) - (\sum_{i=1}^n a_i) \cdot (\sum_{i=1}^n b_i)}{\sqrt{[n \cdot \sum_{i=1}^n a_i^2 - (\sum_{i=1}^n a_i)^2][n \cdot \sum_{i=1}^n b_i^2 - (\sum_{i=1}^n b_i)^2]}}, \quad (3.4)$$

onde n é o número de características da instância, a_i representa a i -ésima característica da instância de teste, e b_i representa a i -ésima característica da instância de validação.

Os resultados com os experimentos utilizando a distância de Canberra, a distância de Cosine e a distância de Pearson podem ser encontrados no próximo capítulo, na Seção 4.4.1. Já a próxima seção trata do uso de informações de agrupamento em conjunto com o método KNORA.

3.1.2 Método de Seleção baseada em Oráculos Mais Próximos com Informações Adicionais de Agrupamento

A previsão do comportamento dos classificadores em uma instância a ser classificada pode ser realizada baseando-se em instâncias parecidas. Deve-se saber quais classificadores reconhecem cada uma dessas instâncias parecidas. Uma maneira de localizar instâncias parecidas é calcular distâncias entre instâncias, como no algoritmo k -NN embutido no método KNORA. Uma outra maneira é utilizar alguma técnica de agrupamento.

Técnicas de agrupamento (*clustering*) visam formar, de maneira automática, grupos de dados segundo algum critério de semelhança, o qual depende do algoritmo e do problema a ser tratado. K -médias [MacQueen, 1967] é uma técnica de agrupamento por partições simples

que tenta encontrar k grupos diferentes. Normalmente, cada agrupamento é representado por um centroide, constituído da média de seus pontos. O método K -médias é bastante utilizado na literatura, e por esse motivo, foi escolhido para esse trabalho.

Na seleção de subconjunto baseada em acurácia e diversidade, de [Santana *et al.*, 2006], foram utilizados tanto o agrupamento quanto a vizinhança obtida por meio de cálculo de distância. Porém, como citado na Seção 2.3.3, houve duas versões distintas do método: uma com o algoritmo K -médias e outra com o algoritmo k -NN. Não houve, então, o uso simultâneo de informações obtidas pelos dois algoritmos, a que essa seção se propõe.

O método de seleção baseada em k oráculos mais próximos com informações adicionais de agrupamento [Vriesmann *et al.*, 2012], denominado KLUNORA, utiliza, do ponto de vista do k -NN, apenas informações de acurácia local geral dos classificadores, como em [Ko *et al.*, 2008]. Já do algoritmo K -médias são obtidos dados sobre a frequência das classes. Por isso, o parâmetro K do algoritmo K -médias possui valor igual à quantidade de classes do domínio tratado.

O algoritmo KLUNORA tem duas variações: o KLUNORA-ELIMINATE e o KLUNORA-UNION. Assim como no KNORA (Seção 2.3.1), o ELIMINATE e o UNION determinam critérios para a seleção dos classificadores. Primeiramente, são calculados os k vizinhos da instância de teste. Então, no caso do KLUNORA-ELIMINATE, são escolhidos os classificadores que reconhecem toda a vizinhança delimitada (ou o maior número de vizinhos). No caso do KLUNORA-UNION, são selecionados os classificadores que reconhecem pelo menos uma instância dessa vizinhança. Os classificadores compõem um subconjunto, que é responsável pela classificação. Nesse processo, é localizado o agrupamento ao qual a instância de teste pertence. A mesma equação de distância utilizada no k -NN embutido no método servirá também para o cálculo com o centroide. No caso, optou-se pela distância Euclidiana, que é a métrica de distância utilizada no KNORA original de [Ko *et al.*, 2008]. Cada voto fornecido por cada um dos classificadores é ponderado pela frequência relativa da classe no agrupamento ao qual pertence a instância. A classe com o maior valor de votos ponderados será a vencedora.

Os resultados experimentais do KLUNORA são relatados na Seção 4.4.2 do próximo capítulo. A próxima seção, por sua vez, apresenta uma metodologia para o uso das informações sobre a proporção de classes acertadas por cada classificador no método KNORA.

3.1.3 Uso de Informações sobre Proporção de Classes Reconhecidas

Dois estratégias (KNORA-ELIMINATE-W e KNORA-UNION-W) propostas por [Ko *et al.*, 2008] para seleção dinâmica de subconjunto de classificadores baseando-se em k oráculos mais próximos utilizavam o voto ponderado pela distância entre a instância de teste e a instância de validação. Nos experimentos realizados, pôde-se notar alguns incrementos na taxa de reconhecimento no caso do KNORA-UNION-W em relação ao KNORA-UNION. Votos ponderados podem, então, auxiliar no processo de classificação em sistemas baseados em múltiplos classificadores.

O presente estudo utilizará, em subconjuntos dinâmicos criados utilizando o método de oráculos mais próximos, a probabilidade que cada um dos classificadores tem de acertar cada uma das classes como uma tentativa de aproximar-se do oráculo [Vriesmann *et al.*, 2012]. Esse valor de probabilidade será obtido nos dados de validação, e utilizado na base de dados de teste.

Primeiramente serão calculadas as probabilidades gerais que cada classificador tem de acertar cada classe, dado que seu voto foi para aquela classe, por meio dos passos descritos no Algoritmo 3.1. Depois, os resultados serão utilizados no KNORA-ELIMINATE e no KNORA-UNION.

Algoritmo 3.1: Cálculo de probabilidade de reconhecimento das classes

Dados: x = quantidade de classificadores;
 y = quantidade de classes;
 m = quantidade de instâncias de validação;
 R_i = representa uma instância selecionada da base de dados;
 C_j = representa um classificador;
 $classe(C_j, R_i)$ = classe que o classificador C_j fornece à instância R_i ;
 $classe(R_i)$ = classe real da instância R_i ;
 $M_{acerto}[x][y]$ = matriz de acertos contendo x posições de classificadores por y posições de classes;
 $M_{erro}[x][y]$ = matriz de erros contendo x posições de classificadores por y posições de classes;

Resultado: $P[x][y]$ = matriz contendo a proporção de acertos;

```

1 coloque todos os valores de  $M_{acerto}[i][j] = 0$ ;
2 coloque todos os valores de  $M_{erro}[i][j] = 0$ ;
3 para  $i = 1$  até  $m$  faça
4   | selecione a instância  $R_i$ ;
5   | para cada classificador  $C_j$  faça
6   |   | se  $classe(C_j, R_i) \neq classe(R_i)$  então
7   |   |   |  $M_{erro}[j][classe(R_i)] = M_{erro}[j][classe(R_i)] + 1$ ;
8   |   | senão
9   |   |   |  $M_{acerto}[j][classe(R_i)] = M_{acerto}[j][classe(R_i)] + 1$ ;
10  |   | fim
11  | fim
12 fim
13 para cada classificador  $C_i$  faça
14  | para cada classe  $j$  faça
15  |   |  $P[i][j] = \frac{M_{acerto}[i][j]}{M_{acerto}[i][j] + M_{erro}[i][j]}$ ;
16  | fim
17 fim
18 retorne  $P$ ;
```

Assim, as estratégias propostas são:

- KNORA-ELIMINATE-P: Serão escolhidos, para compor um subconjunto, os classificadores que acertam todos os k vizinhos (ou o maior número possível dentre os k vizinhos). Os votos de cada um dos classificadores terão como peso a proporção P calculada de classes para esse classificador.
- KNORA-UNION-P: Serão escolhidos, para compor um subconjunto, os classificadores que acertam pelo menos um vizinho (para cada vizinho que acerta, o classificador pode

fornecer um voto). Os votos de cada um dos classificadores terão como peso a proporção P calculada de classes para esse classificador.

Nas 2 estratégias, o maior valor obtido com a soma de todos os votos determinará qual será a classe vencedora.

Os resultados experimentais podem ser vistos na Seção 4.4.3, que apresenta-se no próximo capítulo. Já na próxima seção, será tratada uma outra abordagem para uso de vizinhança em SDSC, dessa vez utilizando, além dos classificadores selecionados, também as classes fornecidas pelo k -NN embutido no método.

3.2 Fusão do Método KNORA com o Método k -NN

O método KNORA [Ko *et al.*, 2008] utiliza um k -NN embutido para encontrar os vizinhos da instância de teste durante a seleção do subconjunto de classificadores. A medida de distância usada para calcular os vizinhos é a Euclidiana. Os resultados do k -NN poderiam ser considerados informação adicional. Uma questão que surge é: será que o uso de informação adicional vinda do k -NN pode ser interessante para os resultados finais do KNORA?

Para responder a essa questão, diferentes estratégias podem ser avaliadas combinando os resultados do método de seleção dinâmica de conjunto KNORA com os resultados de seu k -NN embutido. Mais precisamente, 5 esquemas diferentes podem ser implementados [Vriesmann *et al.*, 2010] [Vriesmann *et al.*, 2012]:

- **KNORA CU** (*Conditional Use* - Uso Condicional): Execute o método k -NN para a instância de teste e verifique qual é a classe predominante nos vizinhos. Se menos que $Y\%$ de todos os vizinhos da instância de teste em questão pertencem à mesma classe predominante, faça o KNORA (ELIMINATE ou UNION), e use somente a saída do KNORA. Caso contrário, use a saída do k -NN. A Figura 3.1 mostra esse processo. Os pequenos círculos representam instâncias de validação. O hexágono com o valor X é a instância de teste. Os círculos hachurados são os vizinhos, no caso, 5 vizinhos ($k = 5$). Seguindo a seta, têm-se a aplicação do k -NN, e se pelo menos $Y\%$ das instâncias pertencerem à mesma classe, o resultado é essa classe. Caso contrário, seguindo a seta, são selecionados os classificadores (aproveitando os vizinhos já calculados para o k -NN) que serão combinados pelo voto majoritário.
- **KNN+KNORA CF** (*Conditional Fusion* - Fusão Condicional): Execute o método k -NN para a instância de teste e verifique qual é a classe predominante nos vizinhos. Se menos que $Y\%$ de todos os vizinhos da instância de teste em questão pertencem à mesma classe predominante, combine as saídas do k -NN com as saídas do KNORA (ELIMINATE ou UNION). Caso contrário, use a saída do k -NN. A Figura 3.2 mostra esse processo. Os pequenos círculos representam instâncias de validação. O hexágono com o valor X é a instância de teste. Os círculos hachurados são os vizinhos, no caso, 5 vizinhos ($k = 5$). Seguindo a seta, têm-se a aplicação do k -NN, e se pelo menos $Y\%$ das instâncias pertencerem à mesma classe, o resultado é essa classe. Caso contrário, seguindo a seta, são selecionados os classificadores (aproveitando os vizinhos já calculados para o k -NN) que fornecerão seus votos, os quais serão unidos aos votos obtidos pelo k -NN. O resultado é combinação de todos pelo voto majoritário.

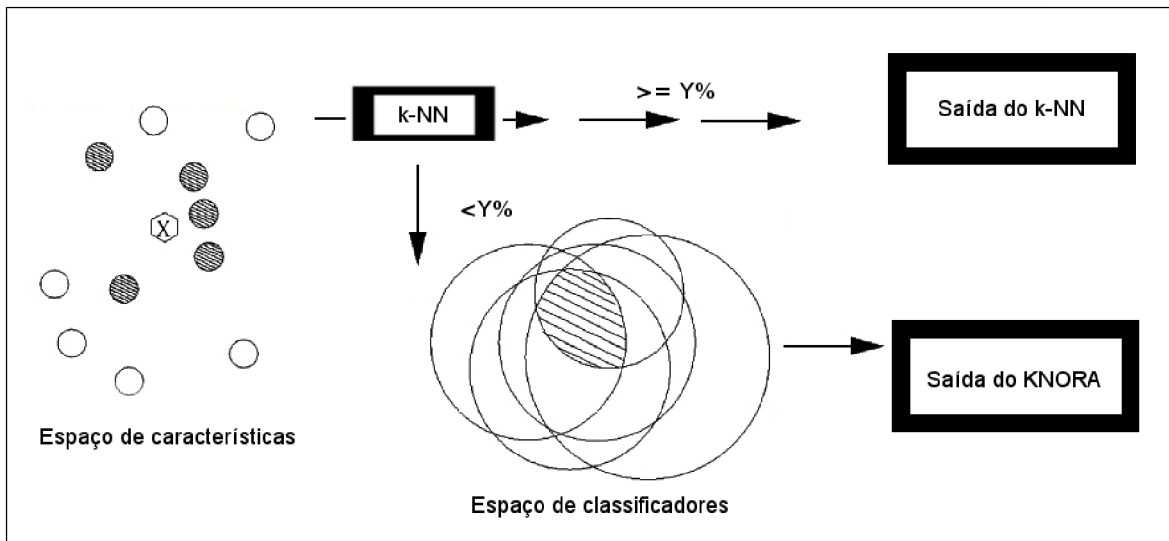


Figura 3.1: Representação do método KNORA CU

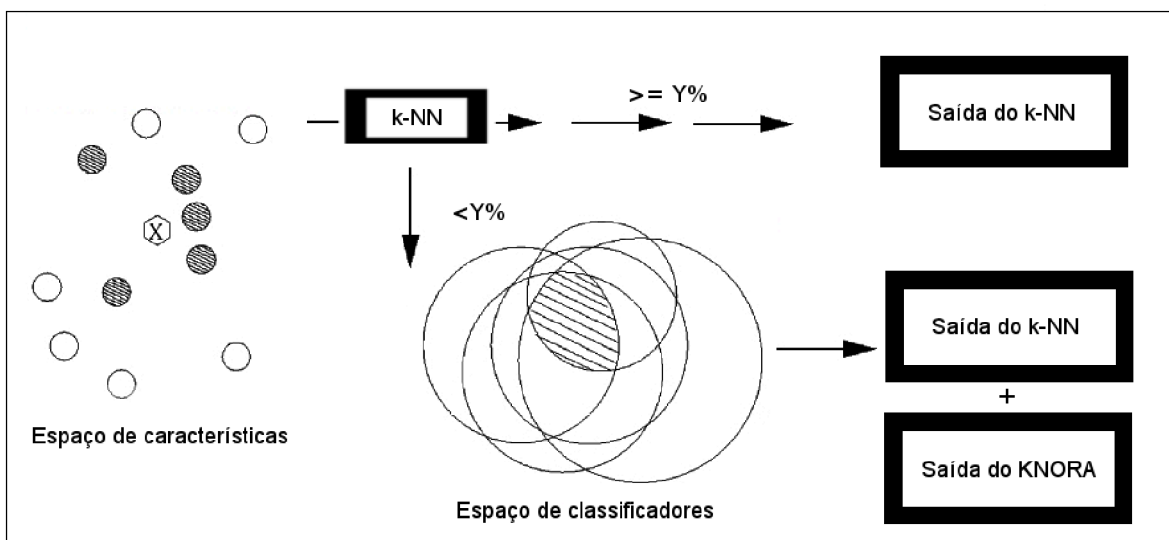


Figura 3.2: Representação do método KNN+KNORA CF

- KNN CU: Execute o método KNORA (ELIMINATE ou UNION) para a instância de teste e verifique qual é a classe vencedora. Se menos que $Y\%$ dos classificadores no conjunto selecionado para a instância de teste em questão fornecem o voto para a classe predominante, use somente a saída do k -NN. Caso contrário, use as saídas do KNORA. A Figura 3.3 mostra esse processo. Os pequenos círculos representam instâncias de validação. O hexágono com o valor X é a instância de teste. Os círculos hachurados são os vizinhos, no caso, 5 vizinhos ($k = 5$). Seguindo a seta, têm-se a seleção dos classificadores através do KNORA, e se pelo menos $Y\%$ dos seus votos pertencerem à mesma classe, o resultado é essa classe. Caso contrário, seguindo a seta, aproveitando os vizinhos já calculados para o KNORA, é selecionada a classe pelo k -NN.

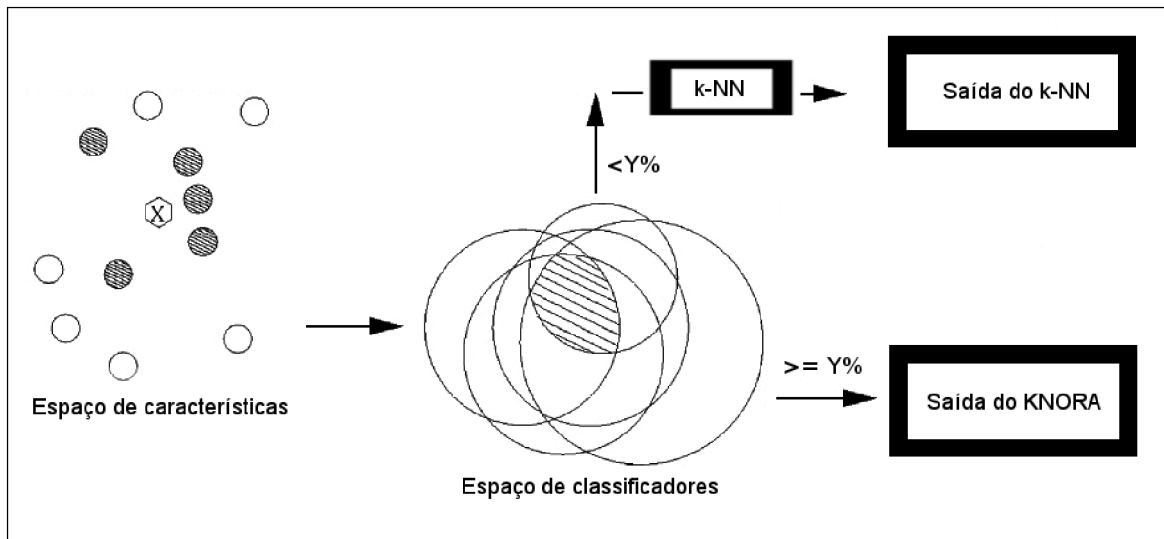


Figura 3.3: Representação do método KNN CU

- KNORA+KNN CF: Execute o KNORA (ELIMINATE ou UNION) para a instância de teste e verifique qual é a classe vencedora. Se menos do $Y\%$ dos classificadores da instância de teste em questão fornecem voto para a classe vencedora, combine as saídas do KNORA (ELIMINATE ou UNION) com as saídas do k -NN. Caso contrário, use as saídas do KNORA. A Figura 3.4 mostra esse processo. Os pequenos círculos representam instâncias de validação. O hexágono com o valor X é a instância de teste. Os círculos hachurados são os vizinhos, no caso, 5 vizinhos ($k = 5$). Seguindo a seta, têm-se a seleção dos classificadores através do KNORA, e se pelo menos $Y\%$ dos seus votos pertencerem à mesma classe, o resultado é essa classe. Caso contrário, seguindo a seta, aproveitando os vizinhos já calculados para o KNORA, são unidos os votos do k -NN aos votos dos classificadores do KNORA.

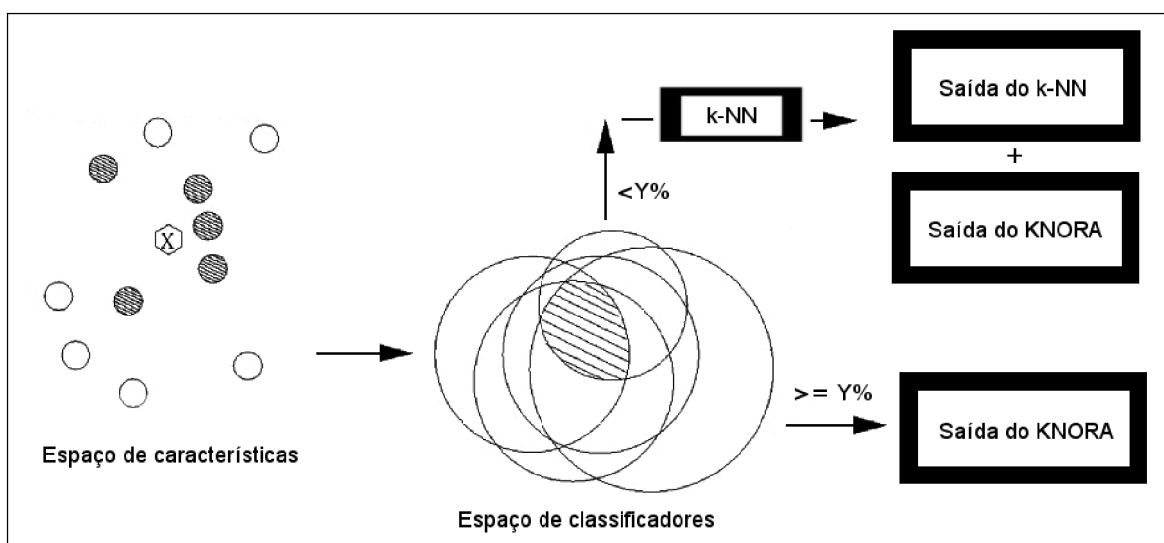


Figura 3.4: Representação do método KNORA+KNN CF

- **KNN+KNORA UF (Unconditional Fusion - Fusão Incondicional):** Para cada instância de teste, combine as saídas do k -NN com as saídas do KNORA (ELIMINATE ou UNION). A Figura 3.5 mostra esse processo. Os pequenos círculos representam instâncias de validação. O hexágono com o valor X é a instância de teste. Os círculos hachurados são os vizinhos, no caso, 5 vizinhos ($k = 5$). Seguindo a seta, têm-se a aplicação do k -NN e a seleção dos classificadores pelo método KNORA, os quais ocorrem no mesmo procedimento. Os votos do k -NN são unidos aos votos dos classificadores.

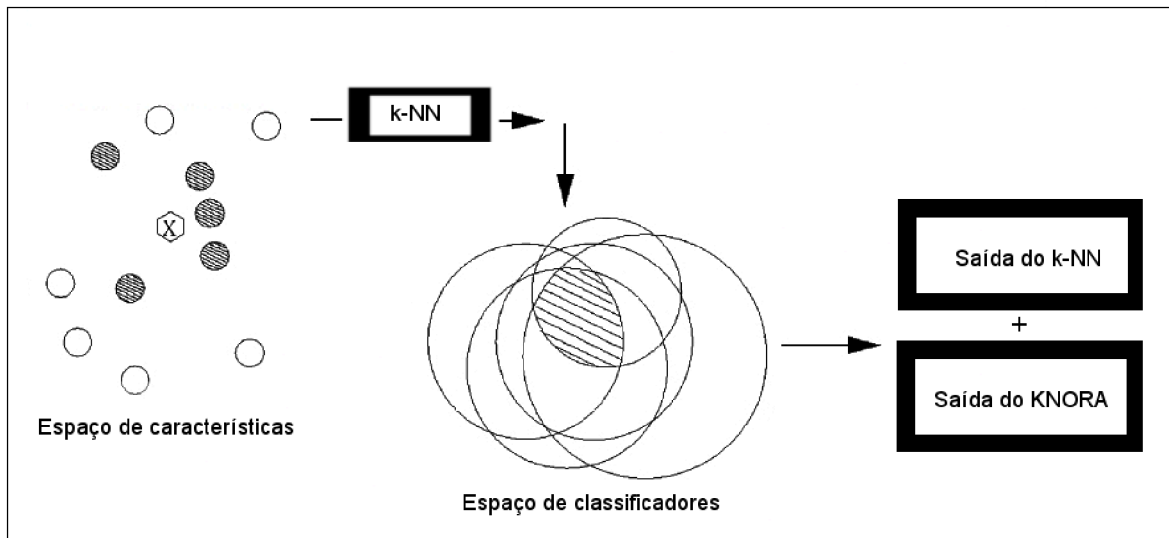


Figura 3.5: Representação do método KNN+KNORA UF

A fusão ou a combinação das saídas é sempre feita pelo esquema de voto majoritário. Os resultados experimentais são tratados no próximo capítulo, na Seção 4.5. Já a próxima seção trata do uso do OLA e do LCA como informações complementares na seleção dinâmica de subconjunto de classificadores.

3.3 Método de Seleção baseada em Acurácia Local Geral e de Classe

A classe atribuída à instância a ser reconhecida pode ser uma informação útil para a decisão de utilizar ou não determinado classificador. Se esse classificador possui uma baixa acurácia dessa classe nas instâncias consideradas parecidas com a instância de teste, a probabilidade de reconhecê-lo também será pequena. Se o classificador possui uma alta acurácia dessa classe, a probabilidade de reconhecê-lo também será alta. A mesma regra vale para um subconjunto de classificadores. De fato, um subconjunto de classificadores com alta acurácia de classe local (LCA, de *Local Class Accuracy*) tem mais chances de reconhecer a instância de teste do que um subconjunto de classificadores que tem baixa LCA, mesmo que tenha alta acurácia local geral (OLA, de *Overall Local Accuracy*). Não é de grande valia um subconjunto ter uma alta taxa de reconhecimento geral na vizinhança, se o reconhecimento ocorreu para classes distintas daquela para a qual será utilizada.

Assim, propõe-se um método onde os valores de OLA e de LCA são utilizados de forma complementar no processo de seleção dinâmica dos classificadores que compõem o subconjunto. O método será nomeado DESCOLA (*Dynamic Ensemble Selection using Class and Overall Local Accuracy*).

O método DESCOLA (Figura 3.6) necessita de uma base de dados de validação onde devem ser conhecidos quais classificadores do conjunto inicial reconhecem quais instâncias. Então, numa primeira fase, para cada instância de teste a ser classificada, é calculada primeiramente a distância Euclidiana (Equação 3.5) para cada uma das instâncias de validação.

$$d_E(A, B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}, \quad (3.5)$$

onde A é uma instância de teste, B é uma instância de validação, n é o número de características da instância, a_i é a i -ésima característica da instância de teste, e b_i é a i -ésima característica da instância de validação.

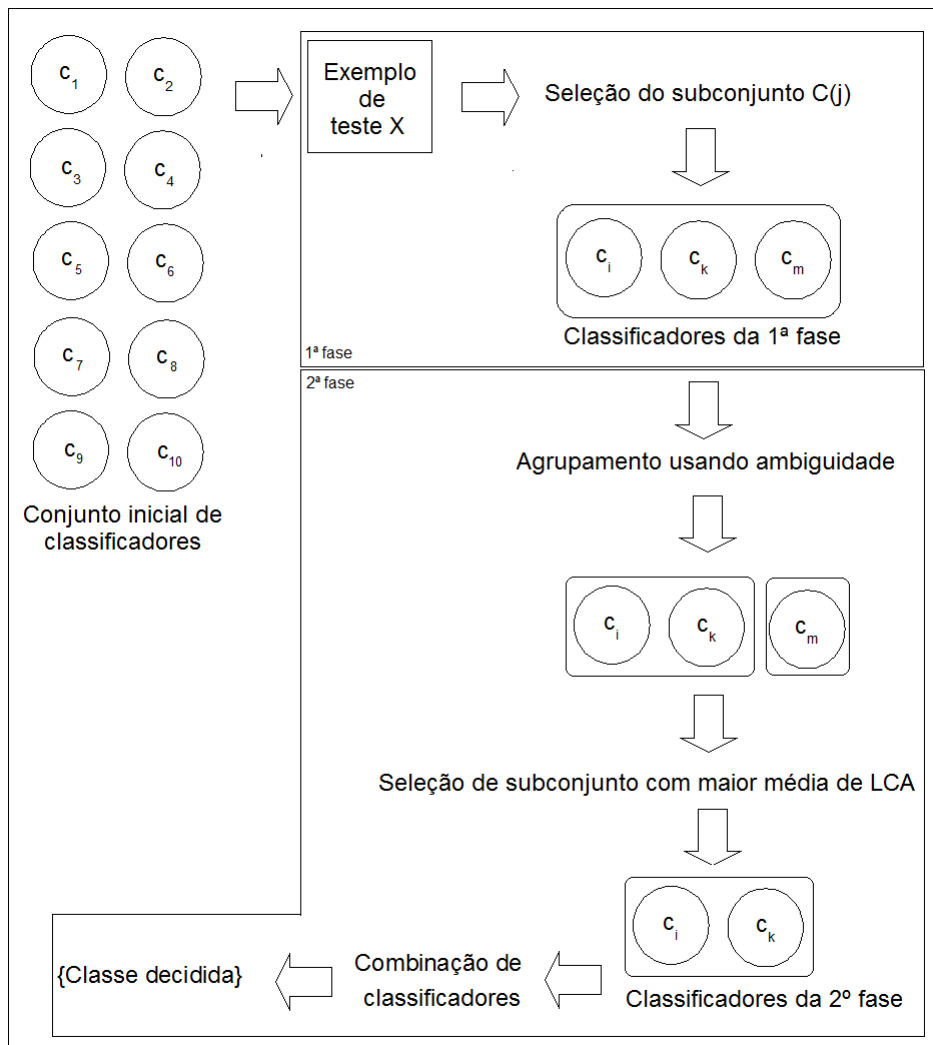


Figura 3.6: Representação do método DESCOLA

Após o cálculo da distância Euclidiana, são selecionadas as k instâncias de validação mais próximas (vizinhas) e, para cada classificador C_j , é calculada a OLA (Equação 3.6).

$$OLA_{j,k}(x^*) = \frac{N_j}{k}, \quad (3.6)$$

onde k é a quantidade total de instâncias na vizinhança, j é o identificador do classificador, e N_j é a quantidade de instâncias vizinhas de x^* que são corretamente classificadas pelo C_j , não importando a classe.

Os valores de OLA de cada classificador do conjunto inicial são utilizados no processo de seleção da primeira fase. De acordo com o processo de seleção, o DESCOLA tem 3 estratégias distintas:

- **DESCOLA-ELIMINATE:** São selecionados apenas os classificadores que possuem o maior valor de OLA. No lado esquerdo da Figura 3.7 esse processo pode ser melhor visualizado. No primeiro desenho da parte superior esquerda da Figura 3.7, os pequenos círculos representam instâncias de validação. O hexágono com o valor X é a instância de teste. Os círculos hachurados são os vizinhos, no caso, 5 vizinhos ($k = 5$). Seguindo a seta, têm-se os classificadores (representados pelas elipses C_i , C_k , C_m , C_n e C_p) que reconhecem pelo menos um dos vizinhos. Seguindo novamente a seta, observam-se os classificadores selecionados (C_i , C_k e C_m), os quais tem o valor de OLA igual a 1, pois reconhecem todos os vizinhos selecionados. Já os classificadores não selecionados (C_n e C_p) acertam a classe de apenas um vizinho e, no caso, possuem valor de $OLA = \frac{1}{5}$. Então, na parte superior direita da Figura 3.7 tem-se o subconjunto de classificadores escolhido na primeira fase para a instância X .
- **DESCOLA-UNION:** São selecionados os classificadores que possuem valor de OLA maior que zero, ou seja, que reconhecem pelo menos um vizinho. No lado esquerdo da Figura 3.8 esse processo pode ser melhor visualizado. No primeiro desenho da parte superior esquerda da Figura 3.8, os pequenos círculos representam instâncias de validação. O hexágono com o valor X é a instância de teste. Os círculos hachurados são os vizinhos, no caso, 5 vizinhos ($k = 5$). Seguindo a seta, têm-se os classificadores (representadas pelas elipses C_i , C_k , C_m , C_n e C_p) que reconhecem pelo menos um dos vizinhos. Seguindo novamente a seta, observam-se os classificadores selecionados (C_i , C_k , C_m , C_n e C_p , ou seja, todos os representados). Os valores de OLA para os classificadores C_i , C_k , C_m , C_n e C_p são, respectivamente, $OLA_{i,5}(x^*) = 1$, $OLA_{k,5}(x^*) = 1$, $OLA_{m,5}(x^*) = 1$, $OLA_{n,5}(x^*) = \frac{1}{5}$ e $OLA_{p,5}(x^*) = \frac{1}{5}$. Supondo que no conjunto inicial há 10 classificadores, 5 classificadores possuem valor de OLA igual a zero, pois não acertam nenhuma das instâncias parecidas com a instância de teste X e, conseqüentemente, não foram selecionados. Então, na parte superior direita da Figura 3.8 tem-se o subconjunto de classificadores escolhido na primeira fase.
- **DESCOLA-UNION-U:** São selecionados os classificadores que possuem valor de OLA maior que zero. Porém, cada classificador será selecionado tantas vezes quanto for a quantidade de vizinhos que reconhece. Assim, cada classificador poderá aparecer mais de uma vez no subconjunto selecionado. Na prática, pode ser utilizado o valor de OLA multiplicado por k como peso para o voto do referido classificador. No lado esquerdo da Figura 3.9 (página 44) esse processo pode ser melhor visualizado. No primeiro desenho

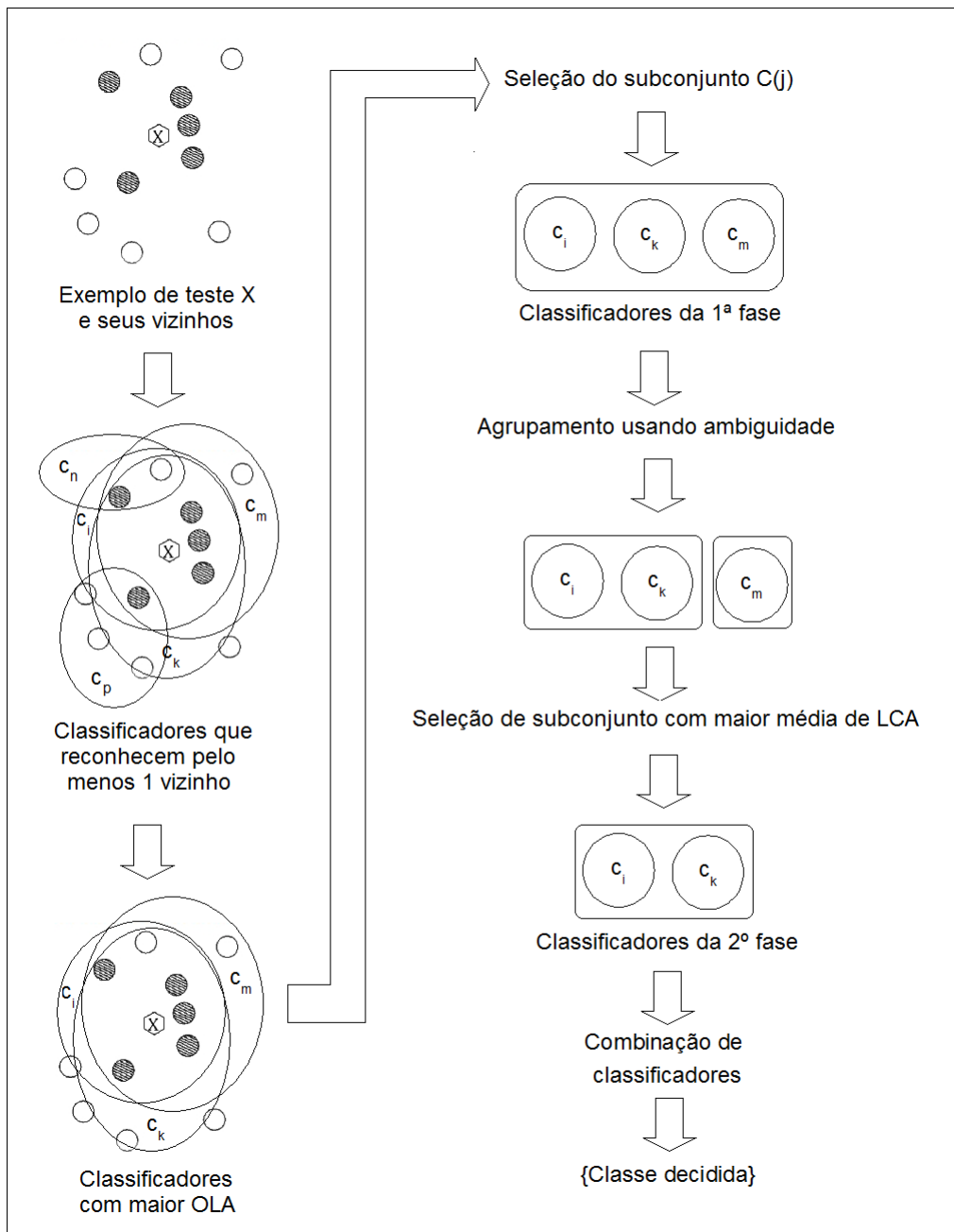


Figura 3.7: Representação do método DESCOLA-ELIMINATE

da parte superior esquerda da Figura 3.9, os pequenos círculos representam instâncias de validação. O hexágono com o valor X é a instância de teste. Os círculos hachurados são os vizinhos, no caso, 5 vizinhos ($k = 5$). Seguindo a seta, têm-se os classificadores (representados pelas elipses C_i , C_k , C_m , C_n e C_p) que reconhecem pelo menos um dos vizinhos. Seguindo novamente a seta, observam-se os classificadores selecionados (C_i , C_k , C_m , C_n e C_p , ou seja, todos os representados). Os valores de OLA para os classificadores C_i , C_k , C_m , C_n e C_p são, respectivamente, $OLA_{i,5}(x^*) = 1$, $OLA_{k,5}(x^*) = 1$, $OLA_{m,5}(x^*) = 1$, $OLA_{n,5}(x^*) = \frac{1}{5}$ e $OLA_{p,5}(x^*) = \frac{1}{5}$. Supondo que no conjunto inicial há

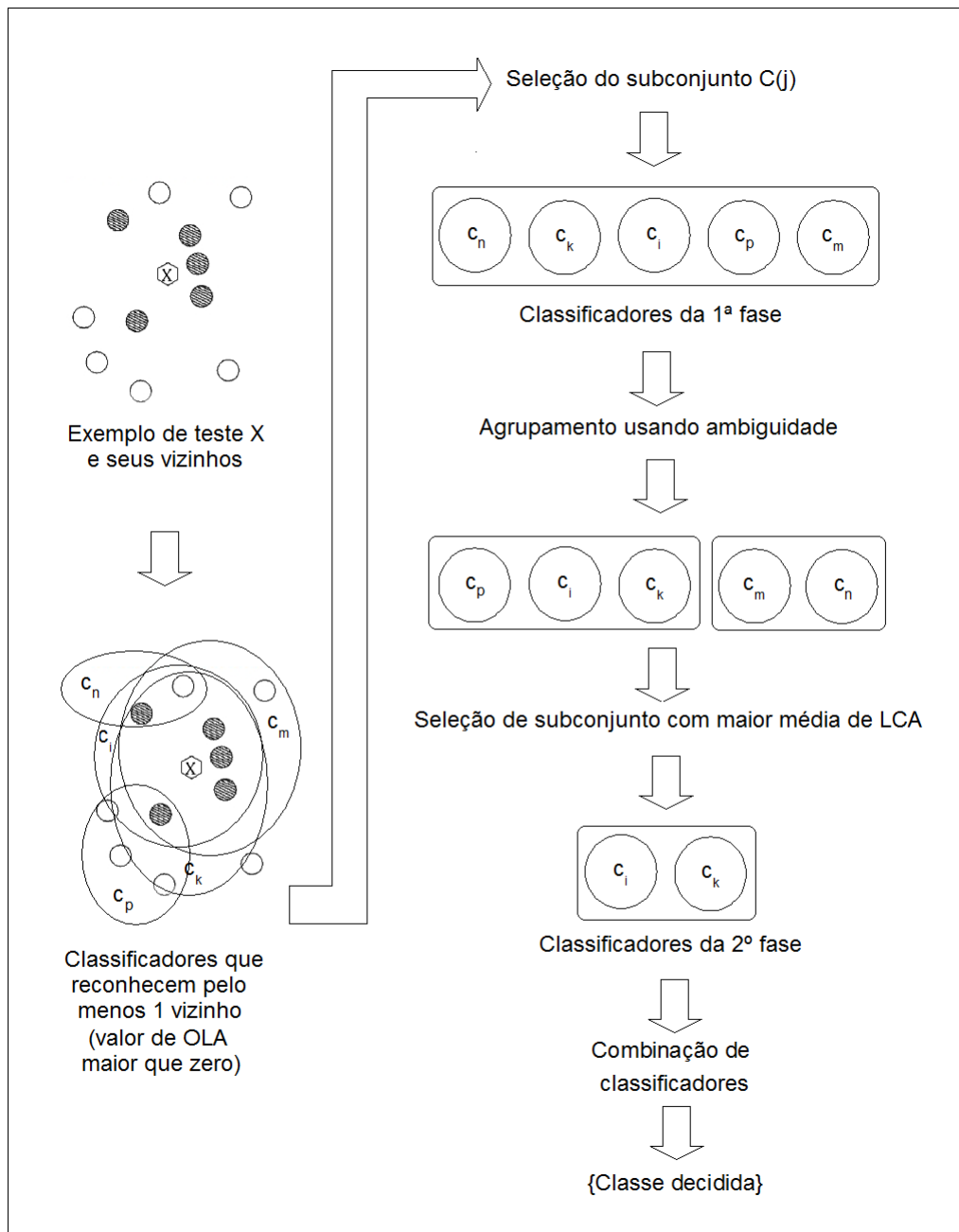


Figura 3.8: Representação do método DESCOLA-UNION

10 classificadores, 5 classificadores possuem valor de OLA igual a zero, pois não acertam nenhuma das instâncias parecidas com a instância de teste X e, conseqüentemente, não foram selecionados. Então, na parte inferior esquerda da Figura 3.9 tem-se o subconjunto de classificadores escolhido na primeira fase. Note que cada classificador aparece no subconjunto tantas vezes quanto foi a quantidade de vizinhos que o mesmo reconheceu. Assim, os classificadores C_n e C_p aparecem uma vez, enquanto que os classificadores C_i , C_k e C_m aparecem 5 vezes.

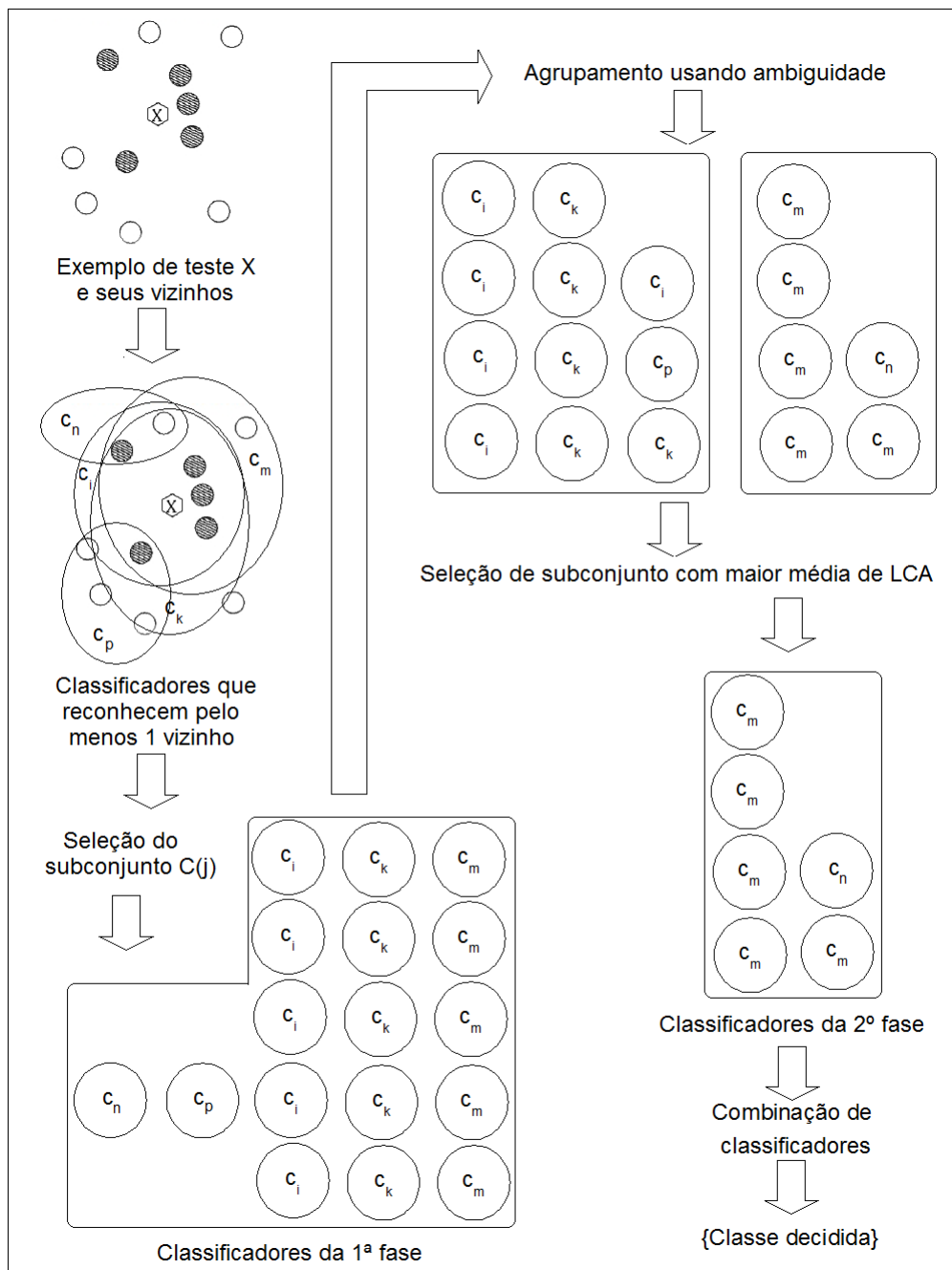


Figura 3.9: Representação do método DESCOLA-UNION-U

Os classificadores preselecionados são utilizados para calcular o valor de LCA em uma segunda fase (Figura 3.6). Para tanto, cada classificador preselecionado atribui uma classe para a instância de teste. Então, esses classificadores são agrupados em subconjuntos de acordo com o valor de ambiguidade [Zenobi & Cunningham, 2001].

Segundo [Tremblay *et al.*, 2004], a ambiguidade é uma medida de diversidade que pode ser utilizada para ajudar na criação de subconjuntos de classificadores com alto desem-

penho. A ambiguidade $\alpha_j(x^*)$ do classificador C_j na instância de teste x^* é representada pela Equação 3.7:

$$\alpha_j(x^*) = \begin{cases} 0 & \text{se } C_j(x^*) = \bar{C}_j(x^*) \\ 1 & \text{caso contrário} \end{cases} \quad (3.7)$$

onde $C_j(x^*)$ é a classe atribuída pelo classificador C_j à instância de teste x^* , e $\bar{C}_j(x^*)$ é a saída do subconjunto ao qual o classificador C_j pertence.

A ambiguidade $\bar{\alpha}(x^*)$ de um subconjunto para a instância de teste x^* será definida pela média da ambiguidade dos classificadores que o compõem, de acordo com a Equação 3.8:

$$\bar{\alpha}(x^*) = \frac{\sum_{i=1}^{N_{\bar{\alpha}}} \alpha_i(x^*)}{N_{\bar{\alpha}}}, \quad (3.8)$$

onde $N_{\bar{\alpha}}$ é a quantidade de classificadores no subconjunto, e $\alpha_i(x^*)$ é o valor de ambiguidade para o classificador i .

Normalmente a ambiguidade é utilizada com o objetivo de mostrar a diversidade dos classificadores de um conjunto, uma vez que a diversidade é uma das características desejáveis, como foi citado na Seção 2.1, na página 7. Quanto maior o valor de $\bar{\alpha}(x^*)$, melhor. No entanto, essa afirmação aparenta ser bastante útil para o conjunto inicial ou para subconjuntos selecionados de maneira estática, uma vez que os classificadores terão que ser capazes de reconhecer instâncias diversificadas. Já quando se trata de subconjuntos selecionados dinamicamente, a finalidade é deixar apenas os classificadores mais específicos possíveis para a instância de teste em questão. Assim, quanto menor for o valor de ambiguidade dos subconjuntos formados, melhor, pois menor será a diversidade. Consequentemente, os classificadores preselecionados na primeira fase são reorganizados em subconjuntos com valor de $\bar{\alpha}(x^*)$ igual a zero. Ou seja, todos os classificadores no mesmo grupo fornecem a mesma classe para a instância de teste.

Mas, não basta agrupar subconjuntos para a mesma classe. Faz-se necessário saber qual subconjunto deverá ter o poder de decisão final na instância de teste. Para tanto, tem que se provar que o subconjunto de classificadores é bom no reconhecimento de sua classe nas instâncias consideradas parecidas com a instância de teste. Neste ponto surge o cálculo do LCA (continuação da segunda fase da Figura 3.6 na página 40).

O cálculo da acurácia de classe local [Woods *et al.*, 1997] [Giacinto & Roli, 1997a], isto é, um tipo de acurácia local *a posteriori*, usa a classe atribuída por cada classificador à instância de teste [Didaci *et al.*, 2005]. Como descrito em [Didaci *et al.*, 2005], suponha que o classificador C_j atribui a classe w_p à instância de teste x^* , ou seja, $C_j(x^*) = w_p$. Então, o LCA do classificador C_j de um subconjunto g pode ser definido como na Equação 3.9.

$$LCA_{j,v}^g(x^*) = \frac{N_{pp}}{\sum_{i=1}^M N_{ip}}, \quad (3.9)$$

onde v é a quantidade de instâncias na vizinhança, j é o identificador do classificador, N_{pp} é o número de instâncias vizinhas de x^* que são corretamente classificadas pelo C_j à classe w_p , e $\sum_{i=1}^M N_{ip}$ é o número total de vizinhos que são atribuídos por C_j à classe w_p .

O valor de LCA de cada classificador é calculado nos v vizinhos mais próximos (na base de dados de validação) da instância de teste. Note que a quantidade de vizinhos utilizados na segunda fase pode ser diferente da quantidade de vizinhos utilizados na primeira fase. Por-

tanto, para diferenciar, k refere-se aos vizinhos da primeira fase, e v refere-se aos vizinhos da segunda fase.

O LCA de cada grupo de classificadores será definido como a média dos valores de LCA de seus elementos, como na Equação 3.10.

$$\overline{LCA}_v^g(x^*) = \frac{\sum_{i=1}^{N_{LCA_v^g}} LCA_{i,v}^g(x^*)}{N_{LCA_v^g}}, \quad (3.10)$$

onde $N_{LCA_v^g}$ é a quantidade de classificadores que tem no subconjunto g para os vizinhos v , e $LCA_{i,v}^g(x^*)$ é o valor de LCA do classificador C_i que tem no subconjunto g para os vizinhos v .

O valor de $LCA_{i,v}^g(x^*)$ na Figura 3.9 apresentará o mesmo classificador mais de uma vez durante o cálculo. Na prática, pode-se ponderar o LCA do classificador pelo valor de OLA obtido na primeira fase de acordo com a Equação 3.11:

$$\overline{LCA}_v^g(x^*) = \frac{\sum_{i=1}^{N_{LCA_v^g}} [LCA_{i,v}^g(x^*) \cdot [OLA_{i,k}(x^*) \cdot k]]}{N_{LCA_v^g}}, \quad (3.11)$$

onde $N_{LCA_v^g}$ é a quantidade de classificadores que tem no subconjunto g para os vizinhos v , $LCA_{i,v}^g(x^*)$ é o valor de LCA do classificador C_i que tem no subconjunto g para os vizinhos v , k é a quantidade total de instâncias na vizinhança na primeira fase, e $OLA_{i,k}(x^*)$ é o valor de OLA para o classificador C_i na primeira fase.

Finalmente, o subconjunto que apresenta o maior valor de média de LCA ($\overline{LCA}_v^g(x^*)$) é escolhido para fornecer a classe final à instância de teste. No caso de empate, ou se a média de LCA em todos os subconjuntos é nula, é selecionado o subconjunto com maior número de classificadores. No caso de empate de ambos os aspectos, um subconjunto é selecionado aleatoriamente.

Além de utilizar a distância Euclidiana para delimitar os vizinhos, o método DESCOLA, em sua primeira fase (OLA), funciona de uma maneira parecida ao método KNORA. Por exemplo, na estratégia do KNORA-ELIMINATE (KE) o subconjunto é composto somente pelos classificadores que acertam todos os vizinhos ou que acertam o maior número de vizinhos dentro do limite estabelecido (k). Esse processo acaba sendo similar à primeira fase do DESCOLA-ELIMINATE, onde são selecionados somente os classificadores que tem o maior valor de OLA na vizinhança. Porém, entre o método KNORA-UNION e a fase OLA do método DESCOLA-UNION, não existe tanta semelhança quanto há entre o KNORA-ELIMINATE e a primeira fase do DESCOLA-ELIMINATE. A semelhança que existe é que, se o classificador acertar pelo menos um vizinho, já é selecionado. Mas, no caso do KNORA-UNION, o voto é ponderado pela quantidade de vizinhos que o classificador acertou. No DESCOLA-UNION, apenas é selecionado o classificador, sem esse conceito de voto ponderado. Tanto o KNORA-UNION quanto a primeira fase do DESCOLA-UNION-U consideram o número de vizinhos reconhecidos pelos classificadores. No entanto, a maneira como essa informação de quantidade de vizinhança é utilizada diferencia-se bastante. No KNORA-UNION, a quantidade de vizinhos reconhecidos vira peso no voto do classificador, enquanto que no DESCOLA-UNION-U essa informação vira peso no cálculo da média de LCA no subconjunto ao qual o classificador pertence.

A primeira grande diferença entre o DESCOLA e o KNORA é que o DESCOLA possui duas fases (OLA e LCA). Outra diferença marcante é que o DESCOLA leva em con-

sideração a classe da instância a ser reconhecido. Além disso, uma medida de diversidade (a ambiguidade) é utilizada para agrupar os classificadores parecidos no DESCOLA. Destes agrupamentos é que será escolhido o subconjunto que aparenta ser mais adequado, de acordo com a média de LCA.

A segunda fase do método DESCOLA possui características específicas relacionadas à classe que o classificador fornece à instância de teste. Portanto, sugere-se também o uso individual da segunda fase (LCA), sem a presença da fase OLA. Para diferenciar-se do procedimento que possui as duas fases, esse método denominou-se DESLCA (*Dynamic Ensemble Selection using Local Class Accuracy*). O método DESLCA utiliza todos os classificadores do conjunto inicial no momento em que são analisadas as classes. Essa é uma de suas desvantagens em relação ao DESCOLA, onde já há uma filtragem de classificadores na primeira fase. Já como vantagem, pode-se considerar o fato de o DESLCA ser independente de uma primeira fase. Assim, o método DESLCA pode ser utilizado individualmente ou então, como uma fase de pós-processamento em subconjuntos de classificadores preselecionados por qualquer método de SDSC.

Os experimentos realizados com o método DESCOLA e com o método DESLCA são tratados no próximo capítulo, na Seção 4.6. A próxima seção trata da conclusão do presente capítulo, salientando resumidamente os principais pontos abordados.

3.4 Conclusão

A metodologia apresentada propõe, primeiramente, algumas adaptações no método KNORA visando verificar o impacto de pequenas mudanças na taxa de reconhecimento:

- Diferentes métricas para cálculo de vizinhos: Ao invés da distância Euclidiana, são utilizadas as distâncias de Canberra, de Cosine e de Pearson para o cálculo da vizinhança da instância de teste.
- Método KLUNORA: O método baseado em k oráculos mais próximos utiliza informações de acurácia local geral dos classificadores para selecionar os classificadores do subconjunto, os quais tem seu voto ponderado pela frequência relativa da sua classe no agrupamento ao qual pertence a instância de teste.
- Informações adicionais de proporção de classes por classificador: A probabilidade que cada um dos classificadores tem de reconhecer cada uma das classes é levada em consideração nos subconjuntos selecionados dinamicamente.

Depois disso, foram propostas três novas abordagens que tratam do uso de vizinhança na seleção dinâmica de subconjunto de classificadores:

- Fusão do k -NN com o KNORA: Para cada instância de teste, além de selecionar os classificadores que acertam os k vizinhos mais próximos, também são utilizadas as classes fornecidas pelos vizinhos.
- Método DESCOLA: Para cada instância de teste, numa primeira fase, selecionam-se os classificadores no valor de OLA obtido nos seus k vizinhos mais próximos na base de dados de validação. Depois, numa segunda fase, os classificadores preselecionados são

agrupados em subconjuntos com ambiguidade zero, e é realizado o cálculo da média da acurácia de classe local de cada subconjunto. Então, o subconjunto com a maior média de acurácia de classe local é escolhido para a classificação final.

- Método DESLCA: Os classificadores do conjunto inicial são agrupados em subconjuntos com ambiguidade zero. Depois, a média da acurácia de classe local é calculada em cada subconjunto. O subconjunto com a maior média é selecionado para a classificação final.

A descrição dos experimentos e os resultados obtidos pelas metodologias propostas podem ser encontrados no próximo capítulo (Capítulo 4).

Capítulo 4

Resultados e Discussão

O presente capítulo trata dos resultados obtidos nos métodos propostos no Capítulo 3. Assim, primeiramente a Seção 4.1 apresentará as bases de dados utilizadas, bem como suas características. Além disso, também explicará a maneira como foram criados os conjuntos iniciais de classificadores.

Já na Seção 4.2 são apresentados valores de taxas de reconhecimento obtidos para essas bases de dados com outras metodologias de classificação e de sistemas baseados em múltiplos classificadores, a fim de se ter parâmetros para comparação com os resultados obtidos pelos métodos propostos. Os testes estatísticos utilizados para comparação são apresentados na Seção 4.3.

A Seção 4.4 trata dos resultados experimentais utilizando adaptações no método KNORA. Resultados com diferentes medidas de distância (Seção 4.4.1) para encontrar os vizinhos, o uso de agrupamento (*clustering*) para auxiliar na classificação (Seção 4.4.2) e o uso de proporção de classes por classificador (Seção 4.4.3) em votos ponderados no método de k oráculos mais próximos podem ser visualizados.

A seguir, na Seção 4.5 podem ser encontrados os experimentos do algoritmo para SDSC que utiliza as classes obtidas pelo procedimento k -NN, o qual está embutido no método de k oráculos mais próximos, juntamente com as informações de reconhecimento dos classificadores. Já a Seção 4.6 trata dos resultados obtidos utilizando o método DESCOLA, que utiliza não apenas informações sobre reconhecimento na vizinhança, mas também informações sobre quais classes os classificadores reconheceram na vizinhança. Também na Seção 4.6 podem ser encontrados os resultados obtidos pelo método DESLCA.

Finalmente, a Seção 4.7 conclui o capítulo.

4.1 Bases de Dados e Conjunto Inicial de Classificadores

Os métodos propostos foram avaliados em 6 problemas de reconhecimento de padrões de pequena escala (*letter*, *liver*, *pima*, *seg*, *wdbc*, *wine*) obtidos no repositório de aprendizado de máquina UCI [Frank & Asuncion, 2010], e também em 2 problemas de reconhecimento de padrão de grande escala (com mais de 100 atributos): a) o reconhecimento de dígitos numéricos manuscritos disponível em *NIST SD19*; e b) a classificação de gêneros musicais usando a base de dados de músicas latinas (*Latin Music Database - LMD*) [Silla Jr. *et al.*, 2008].

As bases de dados *letter*, *liver*, *pima*, *seg*, *wdbc*, *wine* e *NIST SD19* também foram utilizadas em experimentos sobre seleção dinâmica de subconjunto de classificadores

Tabela 4.1: Características das bases de dados

Base de dados	# classes	# total de instâncias	# atributos	Card. de SA	% <i>bag</i>	Validação cruzada	# instâncias de treinamento	# instâncias de validação	# instâncias de teste
letter	26	20000	16	12	66%	10 partes	50% de 9 partes	outras 50% de 9 partes	1 parte
liver	2	345	6	4	66%	10 partes	50% de 9 partes	outras 50% de 9 partes	1 parte
pima	2	768	8	4	66%	10 partes	50% de 9 partes	outras 50% de 9 partes	1 parte
seg	7	2310	19	4	66%	-	210	mesmas de treinamento	2100
wdbc	2	569	30	5	66%	10 partes	50% de 9 partes	outras 50% de 9 partes	1 parte
wine	3	178	13	6	66%	10 partes	50% de 9 partes	outras 50% de 9 partes	1 parte
NIST	10	75089	132	32	66%	-	5000	10000	60089
LMD	10	1300	110	56	66%	3 partes	600	400	300

em [Ko *et al.*, 2008]. Os conjuntos iniciais de classificadores em [Ko *et al.*, 2008] foram criados por subespaços aleatórios (SA) [Ho, 1998], por *bagging* [Kuncheva *et al.*, 2002] e por *boosting* [Schapire *et al.*, 1998]. Um dos algoritmos fracos utilizados foi o k -NN. O presente trabalho utilizará apenas subespaços aleatórios e *bagging*, tendo como algoritmo base o 1 -NN (k -NN onde $k = 1$).

O algoritmo 1 -NN é baseado no paradigma de aprendizado por instância e, utilizando apenas uma instância mais próxima, pode ser considerado um classificador fraco. Um conjunto inicial constituído por classificadores fracos permite mostrar que a seleção dinâmica pode fornecer incremento na taxa de reconhecimento. Ou seja, a diferença na acurácia obtida pelos classificadores fracos individualmente, pelo voto majoritário simples de todos os classificadores do conjunto inicial e pelo método de seleção dinâmica de subconjunto de classificadores pode ser notada, comprovando a eficácia de escolher classificadores mais adequados para cada instância de teste.

A Tabela 4.1 resume as principais características de cada base de dados (a quantidade de classes, a quantidade de instâncias, a quantidade de atributos, a cardinalidade dos subespaços aleatórios, o percentual para o *bagging*, o particionamento para validação cruzada, a quantidade de instâncias selecionadas para treinamento, a quantidade de instâncias da base de dados de validação e a quantidade de instâncias da base de dados de teste). Cabe salientar que, no caso da validação cruzada de 10 partes, não foram colocadas na tabela as quantidades exatas de instâncias de cada parte para as bases de dados de treinamento, de validação e de teste. Ao invés de um número exato, foram especificadas quantas partes constituíram cada base. Por exemplo, na primeira linha da Tabela 4.1 tem-se a base de dados *letter*, com 20000 instâncias. Dividindo 20000 por 10 partes, têm-se 2000 instâncias em cada parte. Em uma primeira execução, as 2000 primeiras instâncias formaram a base de dados de teste, e 50% das 9 partes restantes (9000 instâncias) constituíram a base de dados de validação e as outras 9000 instâncias foram usadas para treinamento.

Todas as bases de dados foram normalizadas, exceto a *NIST SD19*, na qual as configurações utilizadas em [Ko *et al.*, 2008] foram preservadas. Inclusive, no caso da base de dados *NIST SD19* com os classificadores criados por subespaços aleatórios, todos os métodos analisados que incluíam distância Euclidiana na especificação dos vizinhos utilizaram a delimitação de vizinhos usada em alguns experimentos executados por [Ko, 2007], cujos detalhes apresentam-se na Seção 4.1.1.

O protocolo da maioria dos experimentos foi baseado na validação cruzada de 10 partes (*10-fold*), onde 50% das 9 partes usadas para treinamento foram separadas para a base de dados de validação do método. Dez classificadores compuseram o conjunto inicial. Esse protocolo foi usado em quase todos os experimentos, exceto na base de dados de segmentação de

imagens (base de dados *seg*) e nas duas bases de dados de grande escala (*NIST SD19* e *LMD*). Para a base de dados *seg*, foi utilizado o conjunto de treinamento (210 instâncias) e o conjunto de teste (2100 instâncias) sugeridos no repositório UCI. O conjunto de treinamento também foi utilizado para validação. Para a base de dados *NIST SD19* e *LMD*, utilizou-se o mesmo protocolo já usado por [Ko *et al.*, 2008] e [Costa *et al.*, 2012], respectivamente. Nos experimentos da base *NIST SD19*, foram utilizadas 3 bases de dados relacionados com o reconhecimento de números manuscritos: a base de dados de treinamento com 5000 instâncias (*hsf_{0_3}*), a base de dados de validação com 10000 instâncias (*hsf_{0_3}*) e a base de dados de teste com 60089 instâncias (*hsf_{7}*). No esquema de subespaços aleatórios, os 10 classificadores do conjunto inicial foram gerados com subconjuntos de características contendo somente 32 características de 132 (o mesmo protocolo de classificadores proposto em [Ko *et al.*, 2008]). No esquema de *bagging*, os 10 classificadores do conjunto inicial foram gerados com tamanho de *bag* de 66% do conjunto de treinamento.

Em [Costa *et al.*, 2012], a classificação de gêneros musicais *LMD* foi tratada numa abordagem que converte o sinal de áudio em espectrogramas e extrai características de sua representação visual. No presente trabalho, todas as características extraídas foram agrupadas em uma única base de dados. O conjunto total de características extraídas para cada música tinha 885 atributos. Do conjunto de características original, foram selecionadas somente 110 características pelo método CFS (*Correlation-based Feature Selection* - seleção de características baseadas em seleção) [Hall, 1999] [Hall *et al.*, 2009]. Foram utilizados os parâmetros de entrada padrões da ferramenta Weka [Hall *et al.*, 2009]. No esquema de subespaços aleatórios, os 10 classificadores no conjunto inicial foram gerados com subespaços de características contendo 56 características das 110, e no esquema de *bagging*, os 10 classificadores do conjunto foram gerados com tamanho de *bag* de 66% do conjunto de treinamento. Foi utilizado a validação cruzada de 3 partes como descrita nos experimentos de [Costa *et al.*, 2012]: 600 instâncias para treinamento, 400 para validação e 300 para teste.

4.1.1 Cálculo da Vizinhança para Experimentos com a Base *NIST SD19*

O uso de k vizinhos (pertencentes a uma base de dados de validação) de cada uma das instâncias da base de dados de teste é necessário para trabalhar com os métodos de seleção dinâmica de subconjunto de classificadores que utilizam oráculos mais próximos (KNORA¹) ou o k -NN. Em [Ko *et al.*, 2008] foi utilizada a distância Euclidiana para calcular os vizinhos, e o valor de k variou de 1 a 30 para os experimentos com o método k -NN e com o método KNORA.

No conjunto inicial de classificadores obtidos utilizando subespaços aleatórios, um experimento de [Ko, 2007] com a base de dados *NIST SD19* utilizou o conceito de delimitação de vizinhos. Assim, primeiramente, para cada instância de teste, calculou-se a distância Euclidiana em relação a todas as instâncias de teste. Depois, obteve-se o valor de *standard* (Equação 4.1) diminuindo-se a maior distância (*max_dist*) da menor distância (*min_dist*) encontrada para a instância em questão. O resultado foi dividido por 100 e, posteriormente, adicionado à menor distância (*min_dist*) encontrada. As instâncias de validação que possuíam distância menor que *standard* poderiam ser consideradas vizinhas e, conseqüentemente, selecionadas. As demais seriam descartadas, uma vez que não fazem parte da vizinhança de acordo com o critério estabelecido, ou seja, possuem uma distância maior que a mínima permitida.

¹Maiores informações sobre o método KNORA são apresentadas na Seção 2.3.1

$$standard = min_dist + \frac{max_dist - min_dist}{100}. \quad (4.1)$$

Dessa forma, ao executar os algoritmos k -NN e KNORA, algumas instâncias de teste não possuíam os k vizinhos requisitados. Por exemplo, suponha que uma determinada instância possui somente 5 vizinhos na base de dados de validação que têm distância menor que *standard*. No caso, se $k = 7$, seriam selecionados os 5 vizinhos e faltariam 2. Em [Ko, 2007] foi estipulado que seria pego o primeiro vizinho no lugar dos vizinhos omitidos. Isso implica em utilizar, nesse exemplo, 3 vezes o primeiro vizinho.

A Figura 4.1 mostra que essa metodologia proposta, nos algoritmos k -NN, KNORA-UNION e KNORA-ELIMINATE, traz resultados diferentes do que utilizar os k vizinhos mais próximos (sem levar em consideração o valor de *standard*). Na maioria dos casos, com a metodologia proposta atinge-se uma melhor taxa de reconhecimento, sendo que a maior taxa para o k -NN foi de 97,40% ($k = 3$), para o KNORA-UNION foi de 97,25% ($k = 1$), e para o KNORA-ELIMINATE foi de 97,52% ($k = 7$ e $k = 8$). Se fosse considerar todos os vizinhos (sem aplicar esse critério de delimitar a distância que pode ser considerada como vizinha), a maior taxa para o k -NN seria de 97,38% ($k = 1$), para o KNORA-UNION seria 97,25% ($k = 1$), e para o KNORA-ELIMINATE seria de 97,47% ($k = 4$).

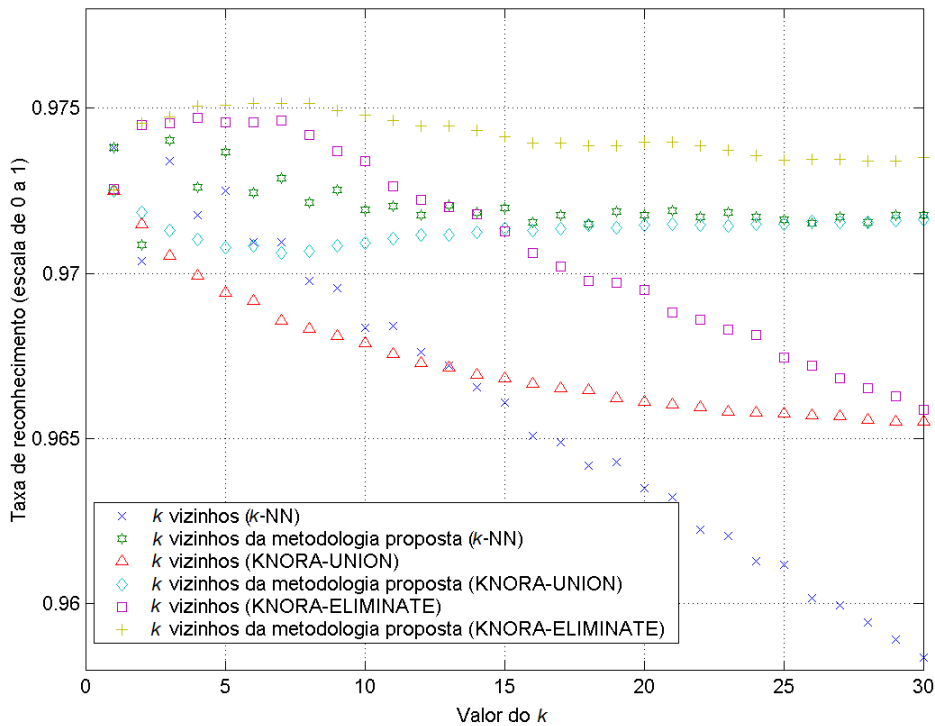


Figura 4.1: Resultados obtidos com os k vizinhos mais próximos e resultados obtidos com a metodologia apresentada de delimitação dos vizinhos na base de dados *NIST SD19*, para o método k -NN e para o método KNORA no conjunto inicial de classificadores criados por meio de subespaços aleatórios

Tabela 4.2: Melhores taxas de reconhecimento (TR) e valores de desvio padrão (σ) obtidos para comparação nos experimentos com conjunto inicial de classificadores criado com subespaços aleatórios

Base de dados	<i>I</i> -NN		MAJ		KE(<i>k</i>)		KU(<i>k</i>)		Oráculo (teste)	
	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ
letter	94,02	0,61	95,18	0,49	95,48 (1)	0,34	95,60 (3)	0,39	98,90	0,19
liver	57,13	8,32	65,82	3,29	65,49 (10)	4,80	68,12 (7)	3,54	97,69	2,65
pima	69,65	5,02	73,44	3,69	72,13 (2)	3,74	74,35 (6)	3,61	98,44	1,19
seg	91,90	-	88,33	-	88,33 (1-30)	-	88,33 (1-30)	-	99,52	-
wdbc	95,43	2,11	96,13	1,84	96,48 (1)	2,05	96,48 (1)	2,05	99,82	0,55
wine	93,30	4,84	94,41	4,54	97,75 (2,3)	2,91	96,67 (2,3)	4,68	100	0
NIST	93,34	-	96,28	-	97,52 (7,8)	-	97,25 (1)	-	99,95	-
LMD	66,45	2,73	69,22	1,39	68,89 (2)	1,95	70,00 (28,29)	1,00	90,22	1,50

Assim, nesse trabalho serão utilizados como padrão os k vizinhos estabelecidos por meio do valor de *standard* quando o método tratado necessitar de vizinhos (calculados com a distância Euclidiana) na base de dados *NIST SD19* para o conjunto inicial obtido com subespaços aleatórios.

4.2 Parâmetros de Comparação

A Tabela 4.2 apresenta as taxas de reconhecimento (TR) e os valores de desvio padrão (σ) obtidos utilizando *I*-NN baseado em todas as características da base de dados e utilizando a combinação pelo voto majoritário simples (MAJ) de todos os classificadores disponíveis no conjunto inicial de classificadores formado com subespaços aleatórios. Além disso, também mostra as melhores taxas de reconhecimento (e respectivos valores de desvio padrão) obtidas utilizando os métodos KNORA-ELIMINATE (KE) e KNORA-UNION (KU)² para a seleção dinâmica de subconjuntos de classificadores, bem como o valor do oráculo na base de dados de teste. O parâmetro k dos métodos foi avaliado de 1 a 30. A melhor taxa de reconhecimento obtida para cada base de dados está em negrito. Já a Tabela 4.3 apresenta os valores obtidos para o *I*-NN, para o MAJ, para o KE, para o KU e para o oráculo utilizando o conjunto inicial de classificadores formado com *bagging*.

Em todos os experimentos, exceto nos experimentos na base de dados *NIST SD19* da Tabela 4.2 (cujos valores foram obtidos em [Ko *et al.*, 2008]), utilizaram-se os parâmetros de entrada padrão do algoritmo de *I*-NN da ferramenta Weka, e os modelos foram treinados na base de dados de treinamento. Já o cálculo do voto majoritário, do valor do oráculo, e os algoritmos KNORA-ELIMINATE e KNORA-UNION foram implementados em Matlab, sendo que a parte de criação do conjunto inicial de classificadores foi realizada com o algoritmo de subespaços aleatórios e o algoritmo de *bagging* da ferramenta Weka. Na fase de criação utilizaram-se as bases de dados de treinamento, enquanto que as bases de dados de validação e de teste foram utilizadas na fase de teste.

²Maiores informações sobre os métodos KNORA-ELIMINATE e KNORA-UNION podem ser visualizadas na Seção 2.3.1

Tabela 4.3: Melhores taxas de reconhecimento (TR) e valores de desvio padrão (σ) obtidos para comparação nos experimentos com conjunto inicial de classificadores criado com *bagging*

Base de dados	I-NN		MAJ		KE(k)		KU(k)		Oráculo (teste)	
	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ
letter	94,02	0,61	93,79	0,51	94,34 (1)	0,46	94,17 (5)	0,42	98,24	0,30
liver	57,13	8,32	57,40	6,18	60,30 (26)	6,70	59,42 (3)	7,02	92,73	4,65
pima	69,65	5,02	70,95	4,81	72,26 (17)	3,42	71,99 (4)	5,23	92,32	2,16
seg	91,90	-	90,76	-	92,19 (1)	-	92,19 (1)	-	97,14	-
wdbc	95,43	2,11	96,13	2,00	96,48 (2-3)	2,04	96,13 (1,3)	1,62	98,59	1,39
wine	93,30	4,84	92,68	3,81	94,93 (6-8)	4,11	94,93 (6-8,23-30)	4,11	98,33	2,68
NIST	93,34	-	95,75	-	95,95 (11)	-	95,86 (23-30)	-	97,85	-
LMD	66,45	2,73	67,89	3,67	68,22 (6)	1,35	67,56 (5)	1,95	86,22	2,14

Na Tabela 4.2 a maioria dos melhores resultados foram obtidos pelo KNORA-UNION como podem ser visualizados nas bases de dados *letter*, *liver*, *pima*, *wdbc* e *LMD*. Nos experimentos da base de dados *letter* obteve-se 94,02% de reconhecimento com o algoritmo *I-NN*, 95,18% com o voto majoritário, 95,48% com o KNORA-ELIMINATE e 95,60% com o KNORA-UNION. Na base de dados *liver* obteve-se 57,13% de reconhecimento com o algoritmo *I-NN*, 65,82% com o voto majoritário, 65,49% com o KNORA-ELIMINATE e 68,12% com o KNORA-UNION. Na base de dados *pima* obteve-se 69,65% de reconhecimento com o algoritmo *I-NN*, 73,44% com o voto majoritário, 72,13% com o KNORA-ELIMINATE e 74,35% com o KNORA-UNION. Na base de dados *wdbc* obteve-se 95,43% de reconhecimento com o algoritmo *I-NN*, 96,13% com o voto majoritário, 96,48% com o KNORA-ELIMINATE e com o KNORA-UNION. Já na base *LMD* obteve-se 66,45% de reconhecimento com o algoritmo *I-NN*, 69,22% com o voto majoritário, 68,89% com o KNORA-ELIMINATE e 70% com o KNORA-UNION.

Nos experimentos da base de dados *NIST SD19* da Tabela 4.2, em [Ko *et al.*, 2008], os autores relataram que utilizando k -NN baseado nas 132 características, com $k = 1$, o desempenho na base de teste é de 93,34%. A combinação de todos os classificadores *I-NN* de 32 características disponível no conjunto inicial (100 elementos) pelo voto majoritário simples fornece 96,28% de precisão de classificação. Além disso, as melhores taxas de reconhecimento para a mesma base de dados foram relatadas como: 97,25% para o KNORA-UNION ($k = 1$) e 97,52% para o KNORA-ELIMINATE ($k = 7$ e $k = 8$). O oráculo para esse conjunto inicial de classificadores fracos é 99,95% de taxa de reconhecimento.

A base de dados *wine*, assim como a base de dados *NIST SD19*, também teve a maior taxa de reconhecimento da Tabela 4.2 sendo atingida pelo uso do KNORA-ELIMINATE (97,75%, para $k = 2$ e $k = 3$). As demais taxas de reconhecimento obtidas foram: 93,30% para o k -NN, 94,41% para o voto majoritário de todos os classificadores do conjunto inicial e 96,67% ($k = 2$ e $k = 3$) para o KNORA-UNION. Observa-se que o método KNORA-UNION também foi melhor que os demais métodos. O valor do oráculo para essa base de dados é de 100%.

A maior taxa de reconhecimento da Tabela 4.2 para a base de dados *seg* foi atingida pelo método *I-NN* (91,90%). Tanto o voto majoritário de todos os classificadores do conjunto inicial, como o KNORA-ELIMINATE, como o KNORA-UNION obtiveram 88,33% de reco-

nhecimento. A principal diferença entre essa base de dados e as demais é que o conjunto de dados de treinamento é o mesmo utilizado para validação. Isso pode ter influenciado nos resultados, uma vez que os classificadores foram testados em instâncias já vistas anteriormente na fase de treinamento.

Já na Tabela 4.3, as melhores taxas de reconhecimento foram obtidas utilizando o método KNORA-UNION, fato esse que ocorreu nas 8 bases de dados testadas. Na base de dados *wine* houve um empate na melhor taxa de reconhecimento do KNORA-ELIMINATE e do KNORA-UNION (94,93%).

Como pode ser observado, os dois métodos de SDSC (KNORA-ELIMINATE e KNORA-UNION) escolhidos para comparação utilizam vizinhança (que fornece informações locais sobre a acurácia) no momento da seleção do subconjunto. O valor do oráculo (limite superior de taxa de reconhecimento) não foi atingido em nenhuma das situações na Tabela 4.2 e na Tabela 4.3.

4.3 Testes Estatísticos

Os resultados dos métodos MAJ, KE e KU apresentados nas Tabelas 4.2 e 4.3 foram utilizados para comparação com resultados obtidos na adaptação de novas métricas de distância ao método KNORA (Seção 4.4.1), na seleção baseada em oráculos mais próximos usando informações adicionais de agrupamento (Seção 4.4.2) e usando informações de proporção de classes (Seção 4.4.3), bem como com resultados dos novos métodos propostos (Seção 4.5 e Seção 4.6): KE CU, KNN+KE CF, KU CU, KNN+KU CF, KNN(após KE) CU, KE+KNN CF, KNN(após KU) CU, KU+KNN CF, KNN+KE UF, KNN+KU UF, DESCOLA-ELIMINATE, DESCOLA-UNION, DESCOLA-UNION-U e DESLCA.

Os métodos foram testados em bases de dados de diferentes domínios. Foram obtidos valores para as taxas de reconhecimento, para cada método em cada base de dados. As taxas de reconhecimento são variáveis contínuas qualitativas. Assim, para avaliá-las, utilizou-se o teste não paramétrico de Friedman [Demsar, 2006] [Hollander & Wolfe, 1999]. Uma das características do teste de Friedman é que, além de possibilitar avaliar vários métodos em várias bases de dados independentes, os efeitos dos algoritmos somente são levados em consideração depois do ajuste de possíveis efeitos das bases de dados.

O teste de Friedman retorna um valor para uma variável denominada p , que declara um resultado com nível de significância de 95% se for menor que 0,05. Ele apenas indica se houve diferenças significativas nos resultados. Todavia, faz-se necessário saber entre quais métodos essas diferenças foram notadas. Para tanto, após o teste de Friedman, foi realizado um teste de múltiplas comparações onde a correção de Dunn-Sidak foi aplicada aos valores críticos. Posteriormente, foi utilizada a correção de Bonferroni-Dunn no teste de múltiplas comparações. Todos esses testes foram realizados com rotinas disponíveis no Matlab, a um nível de confiança de 95%.

A comparação entre as taxas de reconhecimento obtidas pelos métodos foi analisada conjuntamente em todas as bases de dados utilizando o teste de Friedman e os testes de múltiplas comparações. Porém, quando se deseja comparar pares de métodos em uma única base de dados, outras técnicas estatísticas devem ser utilizadas. Entre essas técnicas, tem-se o teste para diferenças entre proporções ([Dietterich *et al.*, 1995] [Dietterich, 1997]), o qual baseia-se nas diferenças entre a taxa de erro do algoritmo A (método A) e do algoritmo B (método B). Assume-se que a taxa de erro (proporção de exemplos que não foram reconhecidos) cor-

responde à probabilidade de erro do algoritmo em questão. Assim, uma distribuição normal padrão (aproximada) pode ser obtida por meio da Equação 4.2.

$$z = \frac{p_A - p_B}{\sqrt{\frac{2 \cdot p \cdot (1-p)}{n}}}, \quad (4.2)$$

onde p_A é a probabilidade de erro do algoritmo A , p_B é a probabilidade de erro do algoritmo B , p é a média das duas probabilidades de erro ($p = \frac{p_A + p_B}{2}$), e n é a quantidade de exemplos de teste.

A hipótese nula é rejeitada se $|z| > z_{0,975} = 1,96$ (para um teste de dois lados com a probabilidade de rejeitar incorretamente a hipótese nula de 0,05). O teste para diferenças entre proporções foi, então, implementado na linguagem de alto nível GNU Octave e utilizado para avaliar pares de algoritmos em uma das bases de dados (*NIST SD19*). A base de dados *NIST SD19* foi escolhida para uma avaliação mais detalhada pelo fato de ter sido utilizada, com as mesmas configurações, no trabalho de [Ko *et al.*, 2008].

4.4 Abordagens Preliminares

4.4.1 Adaptação de Novas Métricas de Distância ao Método KNORA

Os experimentos apresentados nessa seção avaliam a taxa de reconhecimento do método KNORA utilizando a distância de Canberra, a distância de Cosine e a distância de Pearson para encontrar os vizinhos mais próximos. As bases de dados e os conjuntos iniciais de classificadores utilizados são os mesmos descritos na Seção 4.1. O valor do parâmetro k dos métodos variou de 1 a 30. Foi realizada uma análise mais detalhada dos resultados do conjunto de 100 classificadores obtidos por subespaços aleatórios com 1-NN na base de dados *NIST SD19* (a mesma utilizada por [Ko *et al.*, 2008]).

Canberra

Os melhores resultados (juntamente com seus valores de k) obtidos utilizando a distância de Canberra no método KNORA para o conjunto inicial de classificadores formados por subespaços aleatórios são apresentados na Tabela 4.4. Já a Tabela 4.5 apresenta as melhores taxas de reconhecimento (e respectivos desvio padrão e valores de k) para os classificadores criados com *bagging*. Esses resultados foram comparados com o KNORA-ELIMINATE e o KNORA-UNION da Tabela 4.2 e da Tabela 4.3, respectivamente, os quais apresentam-se nas primeiras colunas da Tabela 4.4 e da Tabela 4.5.

O teste de Friedman [Demsar, 2006] [Hollander & Wolfe, 1999] aplicado nos resultados da Tabela 4.4 apresentou valor de p igual a 0,376, indicando que não houve diferenças significativas a um nível de confiança de 95%. Já na Tabela 4.5, o valor de p foi igual a 0,029, mostrando diferenças significativas a um nível de confiança de 95%. Porém, aplicando o teste de múltiplas comparações com a correção de Dunn-Sidak e a correção de Bonferroni-Dunn, não se pôde verificar sobre quais métodos essas diferenças ocorreram. Isso pode indicar que houve empate em alguns casos e que as diferenças foram pontuais, ou seja, em apenas algumas bases de dados (não na maioria). As diferenças também podem ter ocorrido entre os métodos KNORA-ELIMINATE e KNORA-UNION, e não entre o KE utilizando distância Euclidiana e

Tabela 4.4: Melhores taxas de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando a distância Euclidiana e a distância de Canberra no método KNORA aplicado no conjunto inicial de classificadores formados por subespaços aleatórios

Base de dados	Distância Euclidiana				Distância de Canberra			
	KE(k)		KU(k)		KE(k)		KU(k)	
	TR	σ	TR	σ	TR	σ	TR	σ
letter	95,48(1)	0,34	95,60(3)	0,39	95,47(1)	0,36	95,60(3)	0,41
liver	65,49(10)	4,80	68,12(7)	3,54	67,22(7)	4,37	67,53(6)	4,10
pima	72,13(2)	3,74	74,35(6)	3,61	71,87(2)	3,42	74,09(9)	3,89
seg	88,33(1-30)	-	88,33(1-30)	-	88,33(1-30)	-	88,33(1-30)	-
wdbc	96,48(1)	2,05	96,48(1)	2,05	96,48(1)	1,86	96,48(1)	1,86
wine	97,75(2,3)	2,91	96,67(2,3)	4,68	98,33(3)	2,68	97,22(2,3)	3,93
NIST	97,52(7,8)	-	97,25(1)	-	97,13(6-30)	-	96,99(8-10)	-
LMD	68,89(2)	1,95	70,00(28,29)	1,00	69,22(1)	3,34	70,00(26)	0,67

Tabela 4.5: Melhores taxas de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando a distância Euclidiana e a distância de Canberra no método KNORA aplicado no conjunto inicial de classificadores formados por *bagging*

Base de dados	Distância Euclidiana				Distância de Canberra			
	KE(k)		KU(k)		KE(k)		KU(k)	
	TR	σ	TR	σ	TR	σ	TR	σ
letter	94,34(1)	0,46	94,17(5)	0,42	94,20(1)	0,59	94,08(4)	0,46
liver	60,30(26)	6,70	59,42(3)	7,02	59,13(4)	9,34	57,99(28)	6,17
pima	72,26(17)	3,42	71,99(14)	5,23	72,65(1)	3,22	73,17(2)	4,14
seg	92,19(1)	-	92,19(1)	-	92,38(5)	-	91,62(2)	-
wdbc	96,48(2-3)	2,04	96,13(1,3)	1,62	96,31(1)	2,10	96,31(1)	2,10
wine	94,93(6-8)	4,11	94,93(6-8,23-30)	4,11	94,93(1-5)	4,11	94,93(1,3-6,24-30)	4,11
NIST	95,95(11)	-	95,86(23-30)	-	95,94(7,8)	-	95,83(7,8)	-
LMD	68,22(6)	1,35	67,56(5)	1,95	68,78(4)	1,35	67,89(5)	2,59

o KE utilizando distância de Canberra, ou entre o KU utilizando distância Euclidiana e o KU utilizando distância de Canberra.

Uma análise das diferenças de taxas de reconhecimento obtidas para cada execução do KNORA foi realizada na base de dados *NIST SD19* (com conjunto inicial de 100 classificadores, obtidos por subespaços aleatórios com k -NN). A Tabela 4.6 apresenta os valores obtidos com o KNORA-ELIMINATE e com o KNORA-UNION para k variando de 1 a 30, tanto para a distância Euclidiana como para a distância de Canberra. Utilizando a distância Euclidiana, os melhores resultados atingidos foram de 97,52% para o KNORA-ELIMINATE e de 97,25% para o KNORA-UNION, enquanto que utilizando Canberra, os melhores resultados atingidos foram de 97,13% para o KNORA-ELIMINATE e de 96,99% para o KNORA-UNION.

O teste de Friedman aplicado nos resultados da Tabela 4.6 teve valor de p igual a zero, mostrando que houve diferenças significativas a um nível de confiança de 95%. Testando múltiplas comparações com a correção de Dunn-Sidak e a correção de Bonferroni-Dunn, pôde-se notar diferenças entre o KE e o KU com a distância Euclidiana, entre o KE e o KU com a distância de Canberra, entre o KE com a distância Euclidiana e o KE com a distância de Canberra, e entre o KU com a distância Euclidiana e o KU com a distância de Canberra. Observando a Tabela 4.6, nota-se que a distância Euclidiana apresentou os melhores resultados.

Há, então, diferenças significativas entre as taxas de reconhecimento obtidas com o método KNORA utilizando um determinado valor de k , para a medida Euclidiana e para a medida de Canberra. Esse fato foi observado na base de dados *NIST SD19*. Porém, analisando do ponto de vista dos melhores resultados obtidos com as métricas para todas as bases de dados, não foi possível notar uma diferença expressiva por meio dos testes anteriores. Para analisar do

Tabela 4.6: Taxa de reconhecimento (em %) para cada k variando de 1 a 30 utilizando a distância Euclidiana e a distância de Canberra no KNORA para o conjunto inicial de classificadores criados por subespaços aleatórios da base de dados *NIST SD19*

k	Distância Euclidiana		Distância de Canberra	
	KNORA-ELIMINATE	KNORA-UNION	KNORA-ELIMINATE	KNORA-UNION
1	97,25	97,25	96,99	96,98
2	97,45	97,19	97,10	96,98
3	97,47	97,13	97,12	96,98
4	97,51	97,10	97,12	96,98
5	97,51	97,08	97,13	96,98
6	97,51	97,08	97,13	96,98
7	97,52	97,06	97,13	96,98
8	97,52	97,07	97,13	96,99
9	97,49	97,08	97,13	96,99
10	97,48	97,09	97,13	96,99
11	97,46	97,11	97,13	96,98
12	97,45	97,12	97,13	96,98
13	97,45	97,12	97,13	96,98
14	97,43	97,12	97,13	96,98
15	97,41	97,13	97,13	96,98
16	97,40	97,13	97,13	96,98
17	97,39	97,13	97,13	96,98
18	97,39	97,15	97,13	96,98
19	97,39	97,14	97,13	96,98
20	97,40	97,15	97,13	96,98
21	97,40	97,15	97,13	96,98
22	97,39	97,15	97,13	96,98
23	97,37	97,14	97,13	96,98
24	97,36	97,15	97,13	96,98
25	97,34	97,15	97,13	96,98
26	97,35	97,16	97,13	96,98
27	97,35	97,16	97,13	96,98
28	97,34	97,15	97,13	96,98
29	97,34	97,16	97,13	96,98
30	97,35	97,16	97,13	96,98

ponto de vista das melhores taxas de reconhecimento obtidas pelos métodos apenas na base de dados *NIST SD19*, pode-se realizar o teste para diferenças entre proporções.

A Tabela 4.7 apresenta os valores de z (Equação 4.2) obtidos no teste para diferenças entre proporções ([Dietterich *et al.*, 1995] [Dietterich, 1997]) da base de dados *NIST SD19* no conjunto inicial de classificadores formados por subespaços aleatórios, enquanto a Tabela 4.8 apresenta os valores de z obtidos com os métodos aplicados no conjunto inicial de classificadores formados por *bagging*.

Os valores de módulo de z das Tabelas 4.7 e 4.8 que são maiores do que 1,96 indicam diferenças a um nível de confiança de 95%. Assim, na Tabela 4.7 houve diferenças entre os métodos:

- a) KE com distância Euclidiana e KU com distância Euclidiana;
- b) KE com distância Euclidiana e KE com distância de Canberra;
- c) KE com distância Euclidiana e KU com distância de Canberra;
- d) KU com distância Euclidiana e KU com distância de Canberra.

Assim, observa-se que tanto o KNORA-ELIMINATE quanto o KNORA-UNION apresentaram diferenças quando utilizaram distância Euclidiana ou distância de Canberra na base de dados *NIST SD19*, com o conjunto inicial de classificadores criados utilizando subespaços aleatórios.

Já nos experimentos com o conjunto inicial criado utilizando *bagging* na base de dados *NIST SD19* (Tabela 4.8) não foram notadas diferenças a esse nível de confiança.

Tabela 4.7: Valores de z obtidos no teste para diferenças entre proporções para KE e KU utilizando a distância Euclidiana e para KE e KU utilizando a distância de Canberra, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados *NIST SD19*

Método	KE com distância Euclidiana	KU com distância Euclidiana	KE com distância de Canberra
KU com distância Euclidiana	2,93	-	-1,26
KE com distância de Canberra	4,19	1,26	-
KU com distância de Canberra	5,62	2,69	1,44

Tabela 4.8: Valores de z obtidos no teste para diferenças entre proporções para KE e KU utilizando a distância Euclidiana e para KE e KU utilizando a distância de Canberra, aplicado no conjunto inicial de classificadores formados por *bagging* da base de dados *NIST SD19*

Método	KE com distância Euclidiana	KU com distância Euclidiana	KE com distância de Canberra
KU com distância Euclidiana	0,79	-	0,70
KE com distância de Canberra	0,09	-0,70	-
KU com distância de Canberra	1,05	0,26	0,96

Nas análises estatísticas, alguns pontos podem ser salientados:

- O teste de Friedman (aplicado em todas as bases de dados) revelou diferenças significativas nas execuções dos conjuntos iniciais obtidos com *bagging*. Com subespaços aleatórios, isso não ocorreu.
- O teste de Friedman (aplicado na base de dados *NIST SD19* para o conjunto de classificadores criado com subespaços aleatórios) indicou diferenças, para cada valor de k , nos resultados das execuções.
- Testes de múltiplas comparações (aplicado na base de dados *NIST SD19* para o conjunto de classificadores criado com subespaços aleatórios) apontaram diferenças entre o KE com a distância Euclidiana e o KE com a distância de Canberra. O mesmo ocorreu para o KU. A distância Euclidiana apresentou os melhores resultados.
- O teste para diferenças entre proporções (aplicado na base de dados *NIST SD19* para o conjunto de classificadores criado com subespaços aleatórios) também indicou distinção entre o KE com a distância Euclidiana e o KE com a distância de Canberra, e entre o KU com a distância Euclidiana e o KU com a distância de Canberra. Com *bagging*, isso não ocorreu a esse nível de confiança.

Desses resultados, pôde-se concluir que, em uma análise global, existiram diferenças entre as melhores taxas de reconhecimento do KNORA-ELIMINATE e do KNORA-UNION, quando a equação de Canberra foi utilizada para calcular a distância. Porém, essas diferenças não ocorreram em todas as bases de dados da mesma maneira, com todos os conjuntos iniciais de classificadores executados. Ou seja, analisando individualmente cada base de dados, em algumas essas diferenças são significativas, enquanto que em outras não. Assim, a melhor alternativa seria a análise individual em cada base de dados.

Cosine

Os melhores resultados (juntamente com seus valores de desvio padrão e de k) para o conjunto inicial de classificadores formados por subespaços aleatórios são apresentados na Tabela 4.9. Já a Tabela 4.10 apresenta as melhores taxas de reconhecimento para os classificadores criados com *bagging*. Foram comparados esses resultados com o KNORA-ELIMINATE e o KNORA-UNION da Tabela 4.2 e da Tabela 4.3, respectivamente.

Tabela 4.9: Melhores taxas de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando a distância Euclidiana e a distância de Cosine no método KNORA aplicado no conjunto inicial de classificadores formados por subespaços aleatórios

Base de dados	Distância Euclidiana				Distância de Cosine			
	KE(k)		KU(k)		KE(k)		KU(k)	
	TR	σ	TR	σ	TR	σ	TR	σ
letter	95,48(1)	0,34	95,60(3)	0,39	95,50(2)	0,31	95,59(2)	0,37
liver	65,49(10)	4,80	68,12(7)	3,54	64,06(1)	8,44	67,82(4)	5,53
pima	72,13(2)	3,74	74,35(6)	3,61	72,27(2)	3,52	74,22(15)	3,03
seg	88,33(1-30)	-	88,33(1-30)	-	88,33(1-30)	-	88,33(1-30)	-
wdbc	96,48(1)	2,05	96,48(1)	2,05	96,13(1)	2,33	96,48(25)	2,50
wine	97,75(2,3)	2,91	96,67(2,3)	4,68	98,33(2)	2,68	97,75(1)	2,91
NIST	97,52(7,8)	-	97,25(1)	-	97,49(9,10,12-15)	-	97,25(1)	-
LMD	68,89(2)	1,95	70,00(28,29)	1,00	68,67(1)	2,33	70,00(17)	0,67

Tabela 4.10: Melhores taxas de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando a distância Euclidiana e a distância de Cosine no método KNORA aplicado no conjunto inicial de classificadores formados por *bagging*

Base de dados	Distância Euclidiana				Distância de Cosine			
	KE(k)		KU(k)		KE(k)		KU(k)	
	TR	σ	TR	σ	TR	σ	TR	σ
letter	94,34(1)	0,46	94,17(5)	0,42	94,32(1)	0,50	94,18(5)	0,47
liver	60,30(26)	6,70	59,42(3)	7,02	59,12(30)	6,81	58,84(3)	6,23
pima	72,26(17)	3,42	71,99(14)	5,23	72,79(18)	3,33	71,86(3)	5,40
seg	92,19(1)	-	92,19(1)	-	92,29(1)	-	92,29(1)	-
wdbc	96,48(2-3)	2,04	96,13(1,3)	1,62	96,31(4)	2,55	96,13(4)	2,16
wine	94,93(6-8)	4,11	94,93(6-8,23-30)	4,11	94,38(2)	4,54	94,93(17)	4,11
NIST	95,95(11)	-	95,86(23-30)	-	95,94(4)	-	95,84(3)	-
LMD	68,22(6)	1,35	67,56(5)	1,95	68,56(1)	1,71	67,33(16)	2,89

O teste de Friedman [Demsar, 2006] [Hollander & Wolfe, 1999] aplicado nos resultados da Tabela 4.9 apresentou valor de p igual a 0,377, indicando que não houve diferenças significativas a um nível de confiança de 95%. Já na Tabela 4.10, o valor de p foi igual a 0,033, mostrando diferenças significativas a um nível de confiança de 95%. Mas, o teste de múltiplas comparações com a correção de Dunn-Sidak e a correção de Bonferroni-Dunn, não demonstrou sobre quais métodos essas diferenças ocorreram.

Uma análise das diferenças de taxas de reconhecimento obtidas para cada execução do KNORA foi realizada na base de dados *NIST SD19* (com conjunto inicial de 100 classificadores, obtidos por subespaços aleatórios com k -NN). A Tabela 4.11 apresenta os valores obtidos com o KNORA-ELIMINATE e com o KNORA-UNION para k variando de 1 a 30, tanto para a distância Euclidiana como para a distância de Cosine. Utilizando a distância Euclidiana, os melhores resultados atingidos foram de 97,52% para o KNORA-ELIMINATE e de 97,25% para o KNORA-UNION, enquanto que utilizando Cosine, os melhores resultados atingidos foram de 97,49% para o KNORA-ELIMINATE e de 97,25% para o KNORA-UNION.

Tabela 4.11: Taxa de reconhecimento (em %) para cada k variando de 1 a 30 utilizando a distância Euclidiana e a distância de Cosine no KNORA para o conjunto inicial de classificadores criados por subespaços aleatórios da base de dados *NIST SD19*

k	Distância Euclidiana		Distância de Cosine	
	KNORA-ELIMINATE	KNORA-UNION	KNORA-ELIMINATE	KNORA-UNION
1	97,25	97,25	97,25	97,25
2	97,45	97,19	97,40	97,21
3	97,47	97,13	97,44	97,20
4	97,51	97,10	97,47	97,19
5	97,51	97,08	97,47	97,20
6	97,51	97,08	97,48	97,20
7	97,52	97,06	97,48	97,20
8	97,52	97,07	97,48	97,21
9	97,49	97,08	97,49	97,22
10	97,48	97,09	97,49	97,21
11	97,46	97,11	97,48	97,21
12	97,45	97,12	97,49	97,21
13	97,45	97,12	97,49	97,21
14	97,43	97,12	97,49	97,21
15	97,41	97,13	97,49	97,21
16	97,40	97,13	97,48	97,21
17	97,39	97,13	97,48	97,22
18	97,39	97,15	97,48	97,22
19	97,39	97,14	97,47	97,22
20	97,40	97,15	97,47	97,22
21	97,40	97,15	97,47	97,22
22	97,39	97,15	97,47	97,22
23	97,37	97,14	97,47	97,22
24	97,36	97,15	97,47	97,23
25	97,34	97,15	97,47	97,23
26	97,35	97,16	97,47	97,23
27	97,35	97,16	97,47	97,24
28	97,34	97,15	97,48	97,24
29	97,34	97,16	97,48	97,24
30	97,35	97,16	97,48	97,24

O teste de Friedman utilizado nos valores apresentados na Tabela 4.11 teve valor de p igual a zero, mostrando que houve diferenças significativas a um nível de confiança de 95%. Testando múltiplas comparações com a correção de Dunn-Sidak e a correção de Bonferroni-Dunn, pôde-se notar diferenças entre o KE e o KU com a distância Euclidiana e entre o KE e o KU com a distância de Cosine, sendo que o KE saiu-se melhor. Também observou-se que há diferenças expressivas entre o KU com a distância Euclidiana e o KU com a distância de Cosine (sendo que Cosine foi melhor) e entre KU com a distância de Cosine e KE com a distância Euclidiana (sendo que KE com a distância Euclidiana foi melhor). Não houve diferenças importantes entre o KE com a distância Euclidiana e o KE com a distância de Cosine. KE com a distância Euclidiana foi o método que apresentou os melhores resultados da Tabela 4.11.

Há, então, diferenças significativas entre as taxas de reconhecimento obtidas com o método KNORA-UNION utilizando um determinado valor de k , para a medida Euclidiana e para a medida de Cosine. Esse fato foi observado na base de dados *NIST SD19*. Porém, analisando do ponto de vista dos melhores resultados obtidos com as métricas em todas as bases de dados, nos conjuntos iniciais criados com subespaços aleatórios, não foi possível notar uma diferença expressiva.

Já a Tabela 4.12 apresenta os valores de z (Equação 4.2) obtidos no teste para diferenças entre proporções ([Dietterich *et al.*, 1995] [Dietterich, 1997]) da base de dados *NIST SD19* no conjunto inicial de classificadores formados por subespaços aleatórios, enquanto a Ta-

bela 4.13 apresenta os valores de z obtidos com os métodos aplicados no conjunto inicial de classificadores formados por *bagging*.

Os valores de módulo de z das Tabelas 4.12 e 4.13 que são maiores do que 1,96 indicam diferenças a um nível de confiança de 95%. Assim, na Tabela 4.12 houve diferenças entre os métodos:

- a) KE com distância Euclidiana e KU com distância Euclidiana;
- b) KE com distância Euclidiana e KU com distância de Cosine;
- c) KU com distância Euclidiana e KE com distância de Cosine;
- d) KE com distância de Cosine e KU com distância de Cosine.

Já nos experimentos com o conjunto inicial criado utilizando *bagging* (Tabela 4.13) não foram notadas diferenças a esse nível de confiança.

Tabela 4.12: Valores de z obtidos no teste para diferenças entre proporções para KE e KU utilizando a distância Euclidiana e para KE e KU utilizando a distância de Cosine, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados *NIST SD19*

Método	KE com distância Euclidiana	KU com distância Euclidiana	KE com distância de Cosine
KU com distância Euclidiana	2,93	-	2,60
KE com distância de Cosine	0,33	-2,60	-
KU com distância de Cosine	2,93	0	2,60

Tabela 4.13: Valores de z obtidos no teste para diferenças entre proporções para KE e KU utilizando a distância Euclidiana e para KE e KU utilizando a distância de Cosine, aplicado no conjunto inicial de classificadores formados por *bagging* da base de dados *NIST SD19*

Método	KE com distância Euclidiana	KU com distância Euclidiana	KE com distância de Cosine
KU com distância Euclidiana	0,79	-	0,70
KE com distância de Cosine	0,09	-0,70	-
KU com distância de Cosine	0,96	0,17	0,87

Nas análises estatísticas, alguns pontos podem ser salientados:

- O teste de Friedman (aplicado em todas as bases de dados) mostrou diferenças significativas nos resultados dos conjuntos iniciais obtidos com *bagging*. O mesmo não foi notado com subespaços aleatórios.
- O teste de Friedman (aplicado na base de dados *NIST SD19* para o conjunto de classificadores criado com subespaços aleatórios) indicou diferenças, para cada valor de k , nos resultados das execuções.
- Testes de múltiplas comparações (aplicado na base de dados *NIST SD19* para o conjunto de classificadores criado com subespaços aleatórios, para cada vaor de k) apontaram diferenças entre o KU com a distância Euclidiana e o KU com a distância de Cosine (Cosine obteve resultado melhor), mas não entre KE com distância Euclidiana e KE com distância de Cosine.

- O teste para diferenças entre proporções (aplicado na base de dados *NIST SD19* para o conjunto de classificadores criado com subespaços aleatórios) indicou que os resultados obtidos pelos métodos foram distintos, mas não apontou diferenças entre KE com distância Euclidiana e KE com distância de Cosine.
- O teste para diferenças entre proporções (aplicado na base de dados *NIST SD19* para o conjunto de classificadores criado com *bagging*) não indicou diferenças com nível de significância de 95%.

Com a distância de Cosine, então, somente no teste de múltiplas comparações, após aplicar Friedman nas melhores taxas de reconhecimento da base de dados *NIST SD19* (subespaços aleatórios, para cada valor de k) é que foram apontadas diferenças no KNORA-UNION. No KNORA-ELIMINATE, os testes não mostraram diferenças entre os resultados com a distância Euclidiana e com a distância de Cosine.

Pearson

Os melhores resultados (juntamente com seus valores de desvio padrão e de k) são apresentados na Tabela 4.14 e na Tabela 4.15 para os conjuntos de classificadores obtidos utilizando subespaços aleatórios e *bagging*, respectivamente. Foram comparados esses resultados com o KNORA-ELIMINATE e o KNORA-UNION da Tabela 4.2 e da Tabela 4.3, respectivamente.

Tabela 4.14: Melhores taxas de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando a distância Euclidiana e a distância de Pearson no método KNORA aplicado no conjunto inicial de classificadores formados por subespaços aleatórios

Base de dados	Distância Euclidiana				Distância de Pearson			
	KE(k)		KU(k)		KE(k)		KU(k)	
	TR	σ	TR	σ	TR	σ	TR	σ
letter	95,48(1)	0,34	95,60(3)	0,39	95,43(1)	0,27	95,57(3)	0,46
liver	65,49(10)	4,80	68,12(7)	3,54	65,28(1)	10,90	68,13(7)	4,92
pima	72,13(2)	3,74	74,35(6)	3,61	72,01(2)	4,38	74,22(7)	3,99
seg	88,33(1-30)	-	88,33(1-30)	-	88,33(1)	-	88,33(1)	-
wdbc	96,48(1)	2,05	96,48(1)	2,05	96,13(1)	1,84	96,30(2)	1,96
wine	97,75(2,3)	2,91	96,67(2,3)	4,68	97,75(1)	2,91	97,75(1)	2,91
NIST	97,52(7,8)	-	97,25(1)	-	97,44(4,5,7)	-	97,25(1)	-
LMD	68,89(2)	1,95	70,00(28,29)	1,00	68,33(1)	3,18	70,00(16)	1,20

Tabela 4.15: Melhores taxas de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando a distância Euclidiana e a distância de Pearson no método KNORA aplicado no conjunto inicial de classificadores formados por *bagging*

Base de dados	Distância Euclidiana				Distância de Pearson			
	KE(k)		KU(k)		KE(k)		KU(k)	
	TR	σ	TR	σ	TR	σ	TR	σ
letter	94,34(1)	0,46	94,17(5)	0,42	94,33(1)	0,51	94,18(3)	0,50
liver	60,30(26)	6,70	59,42(3)	7,02	59,71(3)	9,17	59,42(3)	5,91
pima	72,26(17)	3,42	71,99(14)	5,23	72,65(24)	4,76	71,73(16)	4,50
seg	92,19(1)	-	92,19(1)	-	92,19(1)	-	92,19(1)	-
wdbc	96,48(2-3)	2,04	96,13(1,3)	1,62	96,13(10)	2,00	95,96(3)	2,20
wine	94,93(6-8)	4,11	94,93(6-8,23-30)	4,11	94,93(9)	4,11	94,93(9)	4,11
NIST	95,95(11)	-	95,86(23-30)	-	95,93(4)	-	95,85(7)	-
LMD	68,22(6)	1,35	67,56(5)	1,95	69,22(3)	1,02	67,67(3)	2,65

O teste de Friedman [Demsar, 2006] [Hollander & Wolfe, 1999] aplicado nos resultados da Tabela 4.14 apresentou valor de p igual a 0,134, indicando que não houve diferenças significativas a um nível de confiança de 95%. Já na Tabela 4.15, o valor de p foi de 0,003, mostrando diferenças significativas a um nível de confiança de 95%. Aplicando-se o teste de múltiplas comparações com a correção de Dunn-Sidak e a correção de Bonferroni-Dunn nos resultados da Tabela 4.15, pôde-se verificar que essas diferenças ocorreram entre o KE e o KU da distância Euclidiana, e entre o KE da distância Euclidiana e o KU da distância de Pearson. Nesses casos, os testes mostraram que o KE com a distância Euclidiana foi melhor. Não foram apontadas diferenças entre o KE da distância Euclidiana e o KE da distância de Pearson, e nem entre o KU da distância Euclidiana e o KE da distância de Pearson. Ou seja, não foram indicadas diferenças nos melhores resultados atingidos somente pela mudança da métrica de distância.

Uma análise das diferenças de taxas de reconhecimento obtidas para cada execução do KNORA foi realizada na base de dados *NIST SD19* (com conjunto inicial de 100 classificadores, obtidos por subespaços aleatórios com k -NN). A Tabela 4.16 apresenta os valores obtidos com o KNORA-ELIMINATE e com o KNORA-UNION para k variando de 1 a 30, tanto para a distância Euclidiana como para a distância de Pearson. Utilizando a distância Euclidiana, os melhores resultados atingidos foram de 97,52% para o KNORA-ELIMINATE e de 97,25% para o KNORA-UNION, enquanto que utilizando Pearson, os melhores resultados atingidos foram de 97,44% para o KNORA-ELIMINATE e de 97,25% para o KNORA-UNION.

Tabela 4.16: Taxa de reconhecimento (em %) para cada k variando de 1 a 30 utilizando a distância Euclidiana e a distância de Pearson no KNORA para o conjunto inicial de classificadores criados por subespaços aleatórios da base de dados *NIST SD19*

k	Distância Euclidiana		Distância de Pearson	
	KNORA-ELIMINATE	KNORA-UNION	KNORA-ELIMINATE	KNORA-UNION
1	97,25	97,25	97,25	97,25
2	97,45	97,19	97,38	97,20
3	97,47	97,13	97,41	97,21
4	97,51	97,10	97,44	97,19
5	97,51	97,08	97,44	97,19
6	97,51	97,08	97,43	97,20
7	97,52	97,06	97,44	97,19
8	97,52	97,07	97,43	97,20
9	97,49	97,08	97,43	97,20
10	97,48	97,09	97,42	97,21
11	97,46	97,11	97,42	97,21
12	97,45	97,12	97,42	97,21
13	97,45	97,12	97,42	97,21
14	97,43	97,12	97,43	97,22
15	97,41	97,13	97,42	97,22
16	97,40	97,13	97,42	97,22
17	97,39	97,13	97,42	97,22
18	97,39	97,15	97,42	97,22
19	97,39	97,14	97,42	97,22
20	97,40	97,15	97,42	97,22
21	97,40	97,15	97,42	97,22
22	97,39	97,15	97,42	97,23
23	97,37	97,14	97,42	97,23
24	97,36	97,15	97,42	97,23
25	97,34	97,15	97,42	97,23
26	97,35	97,16	97,42	97,23
27	97,35	97,16	97,42	97,23
28	97,34	97,15	97,42	97,23
29	97,34	97,16	97,42	97,23
30	97,35	97,16	97,42	97,23

Na Tabela 4.16, o teste de Friedman apresentou $p = 0$, mostrando que houve diferenças significativas a um nível de confiança de 95%. O KNORA-ELIMINATE apresentou os melhores resultados. Testando múltiplas comparações com a correção de Dunn-Sidak e a correção de Bonferroni-Dunn, foram apontadas diferenças entre todos os métodos, exceto entre o KE com a distância Euclidiana e as estratégias do KNORA com a distância de Pearson.

Sintetizando, há diferenças significativas entre as taxas de reconhecimento obtidas com o método KNORA-UNION utilizando um determinado valor de k , para a medida Euclidiana e para a medida de Pearson. Esse fato foi observado na base de dados *NIST SD19*.

Já a Tabela 4.17 apresenta os valores de z (Equação 4.2) obtidos no teste para diferenças entre proporções ([Dietterich *et al.*, 1995] [Dietterich, 1997]) da base de dados *NIST SD19* no conjunto inicial de classificadores formados por subespaços aleatórios, enquanto a Tabela 4.18 apresenta os valores de z obtidos com os métodos aplicados no conjunto inicial de classificadores formados por *bagging*.

Os valores de módulo de z das Tabelas 4.17 e 4.18 que são maiores do que 1,96 indicam diferenças a um nível de confiança de 95%. Como pode-se observar, não houve diferença entre os resultados de KU com distância Euclidiana e de KU e com distância de Pearson na Tabela 4.17. Os outros métodos apresentaram resultados diferentes entre si para essa base de dados.

Já nos experimentos com o conjunto inicial criado utilizando *bagging* (Tabela 4.18) não foram notadas diferenças a esse nível de confiança.

Tabela 4.17: Valores de z obtidos no teste para diferenças entre proporções para KE e KU utilizando a distância Euclidiana e para KE e KU utilizando a distância de Pearson, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados *NIST SD19*

Método	KE com distância Euclidiana	KU com distância Euclidiana	KE com distância de Pearson
KU com distância Euclidiana	2,93	-	2,05
KE com distância de Pearson	0,88	-2,05	-
KU com distância de Pearson	2,93	0	2,05

Tabela 4.18: Valores de z obtidos no teste para diferenças entre proporções para KE e KU utilizando a distância Euclidiana e para KE e KU utilizando a distância de Pearson, aplicado no conjunto inicial de classificadores formados por *bagging* da base de dados *NIST SD19*

Método	KE com distância Euclidiana	KU com distância Euclidiana	KE com distância de Pearson
KU com distância Euclidiana	0,79	-	0,61
KE com distância de Pearson	0,18	-0,61	-
KU com distância de Pearson	0,87	0,09	0,70

Nas análises estatísticas, alguns pontos podem ser salientados:

- O teste de Friedman (aplicado em todas as bases de dados) mostrou diferenças significativas nos resultados dos conjuntos iniciais obtidos com *bagging*. O mesmo não foi notado com subespaços aleatórios.
- O teste de Friedman (aplicado na base de dados *NIST SD19* para o conjunto de classificadores criado com subespaços aleatórios) indicou diferenças, para cada valor de k , nos resultados das execuções.

- Testes de múltiplas comparações (aplicado na base de dados *NIST SD19* para o conjunto de classificadores criado com subespaços aleatórios) apenas não apontaram diferenças entre o KE com a distância Euclidiana e as estratégias do KNORA com a distância de Pearson..
- O teste para diferenças entre proporções (aplicado na base de dados *NIST SD19* para o conjunto de classificadores criado com subespaços aleatórios) indicou que os resultados obtidos pelos métodos entre si foram distintos, exceto no KU com distância Euclidiana comparado com o KU com distância de Pearson no conjunto inicial criado com subespaços aleatórios. Nesse caso, a taxa de reconhecimento foi igual.
- O teste para diferenças entre proporções (aplicado na base de dados *NIST SD19* para o conjunto de classificadores criado com *bagging*) não indicou diferenças com nível de significância de 95%.

Diferenças significativas entre as melhores taxas de reconhecimento do KE com distância Euclidiana e do KE com distância de Pearson não foram apontadas em nenhum dos testes realizados. Já entre o KU com distância Euclidiana e o KU com distância de Pearson, houve diferença apenas para o teste que considerou as taxas de reconhecimento obtidas para cada valor de k na base de dados *NIST SD19*, para subespaços aleatórios.

As novas métricas de cálculo de vizinhança permitiram a seleção de diferentes classificadores, uma vez que os resultados modificaram-se em relação aos obtidos pelo KNORA-ELIMINATE e pelo KNORA-UNION na Seção 4.2. Porém, as métricas avaliadas não permitiram incremento significativo na melhor taxa de reconhecimento nem do KNORA-ELIMINATE e nem do KNORA-UNION.

Nas 3 novas métricas foram apontadas diferenças a um nível de significância de 95%, com o teste de Friedman em todas as bases de dados nos conjuntos iniciais criados com *bagging*. Porém, o teste de múltiplas comparações não indicou diferenças entre o KE com distância Euclidiana e o KE com a nova métrica, e nem entre o KU com a distância Euclidiana e o KU com a nova métrica.

O teste de Friedman não apontou diferenças nos conjuntos iniciais criados com subespaços aleatórios (analisando todas as bases de dados) com o uso das 3 novas métricas. Porém, analisando individualmente uma das bases de dados (e seu conjunto de classificadores criados com subespaços aleatórios), para cada valor de k , sempre houve diferenças nos resultados do KU. Já quanto ao KE, apenas Canberra demonstrou diferenças entre seus resultados e o do KE original (com distância Euclidiana).

Já no teste para diferenças entre proporções, aplicado nessa mesma base de dados, com classificadores criados com subespaços aleatórios, apenas na distância de Canberra é que foram percebidas diferenças com os melhores resultados do KE e do KU originais. Com *bagging*, nos 3 casos não foram notadas diferenças.

Também objetivando verificar se as taxas de reconhecimento se alterarão, a próxima seção tratará dos experimentos de outra abordagem preliminar, que utiliza informações de agrupamento juntamente com o KNORA.

4.4.2 Método de Seleção baseada em Oráculos Mais Próximos com Informações Adicionais de Agrupamento

O método de seleção baseada em k oráculos mais próximos com informações adicionais de agrupamento, denominado KLUNORA, utiliza informações sobre o reconhecimento local dos classificadores, do ponto de vista do k -NN, como em [Ko *et al.*, 2008]. Já do algoritmo K -médias são obtidos dados sobre a frequência das classes.

O agrupamento é um método de aprendizado de máquina não-supervisionado. Assim, primeiramente, a classe real da base de dados de validação de cada um dos domínios aplicados foi removida, permanecendo apenas os atributos previsores em cada registro. Após isso, a base foi submetida ao processo de agrupamento.

Para o processo de agrupamento foi utilizado o método K -médias, implementado na ferramenta Weka [Hall *et al.*, 2009]. Como o uso do K -médias tinha a finalidade de obter dados sobre as classes, o parâmetro de entrada correspondente ao número de agrupamentos que deveriam ser criados foi igual à quantidade de classes distintas em cada base de dados. Ao final do processo, foram gerados c centroides, pressupondo que cada centroide corresponderia a uma classe dominante.

Nota-se que o processo de criação dos agrupamentos é realizado em um momento anterior à classificação dinâmica, resultando em valores de centroides armazenados para uso posterior. Obtidos os agrupamentos na base de dados de validação, bem como a frequência relativa de cada classe nos agrupamentos, o próximo passo foi a aplicação dessas informações no KLUNORA.

O algoritmo KLUNORA teve duas variações: o KLUNORA-ELIMINATE e o KLUNORA-UNION. No KLUNORA-ELIMINATE, após o cálculo dos k vizinhos da instância de teste, são escolhidos os classificadores que reconhecem toda essa vizinhança de validação (ou o maior número de vizinhos dentro desse limite). Já no KLUNORA-UNION, após o cálculo dos k vizinhos da instância de teste, são selecionados os classificadores que reconhecem pelo menos uma instância dessa vizinhança de validação. Além disso, cada classificador pode votar tantas vezes quanto for a quantidade de vizinhos que reconhece. Depois de obtido o subconjunto de classificadores, tanto no KLUNORA-ELIMINATE quanto no KLUNORA-UNION, a instância de teste é vinculada ao agrupamento mais próximo, utilizando-se a distância Euclidiana. Sabendo-se à qual agrupamento a instância de teste pertence, cada voto recebido dos classificadores selecionados no conjunto é multiplicado pela frequência relativa da classe no agrupamento da instância de teste. Posteriormente, esses votos ponderados são somados, e a classe escolhida para cada instância de teste é aquela que possui o maior valor do somatório.

As Tabelas 4.19 e 4.20 apresentam as melhores médias³ de taxa de reconhecimento (e respectivos valores de desvio padrão e do parâmetro k) obtidas utilizando o voto majoritário entre todos os classificadores do conjunto inicial, o método KNORA-ELIMINATE (KE), o método KNORA-UNION (KU), o método KLUNORA-ELIMINATE e o método KLUNORA-UNION no conjunto de classificadores gerados com subespaços aleatórios e no conjunto de classificadores gerados com *bagging*, respectivamente. O valor do parâmetro k dos métodos variou de 1 a 30. A melhor média de taxa de reconhecimento obtida para cada base de dados nas tabelas apresenta-se em negrito.

³Quando foi trabalhado com validação cruzada em uma determinada base de dados, é apresentada a melhor média de taxa de reconhecimento obtida nas execuções e o desvio padrão. Caso contrário, apenas é apresentada a melhor taxa de reconhecimento obtida.

Tabela 4.19: Melhores médias da taxa de reconhecimento (em %) e respectivos valores de desvio padrão obtidos utilizando o algoritmo KLUNORA no conjunto inicial de classificadores gerados com subespaços aleatórios e I -NN

Base de dados	MAJ		KE(k)		KU(k)		KLUNORA-ELIMINATE(k)		KLUNORA-UNION(k)	
	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ
letter	95,18	0,49	95,48	0,34	95,60	0,39	94,67	0,25	94,35	0,42
			(1)		(3)		(3)		(2)	
liver	65,82	3,29	65,49	4,80	68,12	3,54	68,42	6,38	67,55	7,15
			(10)		(7)		(1)		(1)	
pima	73,44	3,69	72,13	3,74	74,35	3,61	72,78	5,65	73,83	2,97
			(2)		(6)		(9)		(2)	
seg	88,33	-	88,33	-	88,33	-	87,05	-	87,05	-
			(1-30)		(1-30)		(1)		(7)	
wdbc	96,13	1,84	96,48	2,05	96,48	2,05	94,02	2,55	92,43	4,04
			(1)		(1)		(8)		(1)	
wine	94,41	4,54	97,75	2,91	96,67	4,68	97,19	4,74	96,05	5,37
			(2,3)		(2,3)		(10)		(1)	
NIST	96,28	-	97,52	-	97,25	-	92,03	-	88,16	-
			(7,8)		(1)		(29,30)		(1)	
LMD	69,22	1,39	68,89	1,95	70,00	1,00	67,89	1,02	68,33	2,00
			(2)		(28,29)		(4)		(3)	

Na Tabela 4.19 observa-se que a única base de dados onde o KLUNORA obteve a melhor média de taxa de reconhecimento foi a *liver*, onde o KLUNORA-ELIMINATE com $k = 1$ obteve 68,42% de precisão. Nas demais bases de dados o KLUNORA obteve índices menores de acertos do que o KNORA. Essa diferença aparentemente não é muita, exceto na base de dados *NIST SD19*. Algumas das principais diferenças da base de dados *NIST SD19* em relação às demais diz respeito ao fato de que o conjunto inicial é constituído por 100 classificadores, enquanto que nas demais tem apenas 10 classificadores, e ao fato de que a distância Euclidiana é calculada utilizando uma delimitação especial dos vizinhos (Seção 4.1.1). Esses fatores podem ter influenciado na taxa de reconhecimento do KLUNORA.

Já na Tabela 4.20, o KLUNORA-ELIMINATE apresentou o melhor resultado em 3 das 8 bases de dados testadas e empatou com o KNORA-ELIMINATE em uma base de dados. Nas bases de dados *liver*, *pima* e *wine*, tanto o KLUNORA-ELIMINATE foi melhor que o KNORA-ELIMINATE quanto o KLUNORA-UNION foi melhor que o KNORA-UNION. Ou seja, nessas 3 bases de dados, as informações de classes nos agrupamentos influenciaram positivamente. Na base de dados onde houve o empate (*LMD*) a taxa de reconhecimento foi de 68,22% ($k = 6$), que é o mesmo valor obtido para o KNORA-ELIMINATE. No entanto, o KLUNORA-ELIMINATE obteve um menor desvio padrão.

A significância dos resultados foi avaliada por meio do teste não paramétrico de Friedman [Demsar, 2006] [Hollander & Wolfe, 1999]. Os valores das melhores médias das taxas de reconhecimento das 8 bases de dados obtidos pelos métodos MAJ, KE, KU, KLUNORA-ELIMINATE e KLUNORA-UNION foram analisados. O valor obtido no teste de Friedman para uma variável denominada p foi 0,024 nos experimentos de subespaços aleatórios e de 0,054 nos experimentos de *bagging*. Isso confirmou um resultado expressivo com nível de confiança de 95% nos resultados da Tabela 4.19. Quanto à Tabela 4.20, não se confirmou resultado significativo a esse nível de confiança, ou seja, não se pôde tirar conclusões sobre quais métodos se sobressaem em todas as bases de dados.

Após o teste de Friedman, um teste de múltiplas comparações com a correção de Dunn-Sidak aplicada aos valores críticos foi realizado. Esse teste demonstrou os pares de algoritmos que eram significativamente diferentes. Assim, no caso da Tabela 4.19, houve diferença

significativa entre os valores das médias das taxas de reconhecimento do método KNORA-UNION e KLUNORA-UNION, ou seja, nesse caso o KLUNORA obteve resultados significativamente piores que o KNORA. Usando a correção de Bonferroni-Dunn, as conclusões foram as mesmas.

Se houvessem sido feitos experimentos apenas com as bases de dados onde o KLUNORA-UNION foi melhor ou houve empate (*liver*, *pima*, *wine* e *LMD*) na Tabela 4.20 o valor de p seria de aproximadamente 0,01, confirmando um resultado significativamente melhor para KLUNORA-UNION em relação à MAJ pelo teste de múltiplas escolhas com a correção de Dunn-Sidak. Disso nota-se a importância do teste em bases de dados diversificadas.

Com 95% de confiança, o teste de Friedman confirmou que o KLUNORA-UNION foi significativamente pior que o KNORA-UNION nas bases de dados com conjuntos de classificadores criados com subespaços aleatórios e não houve diferenças nas taxas de reconhecimento nas bases de dados com conjuntos de classificadores criados com *bagging*.

Tabela 4.20: Melhores médias de taxa de reconhecimento (em %) e respectivos valores de desvio padrão obtidos utilizando o algoritmo KLUNORA no conjunto inicial de classificadores gerados com *bagging* e *1-NN*

Base de dados	MAJ		KE(k)		KU(k)		KLUNORA-ELIMINATE(k)		KLUNORA-UNION(k)	
	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ
letter	93,79	0,51	94,34 (1)	0,46	94,17 (5)	0,42	93,41 (2)	0,42	93,06 (3)	0,50
liver	57,40	6,18	60,30 (26)	6,70	59,42 (3)	7,02	61,14 (1)	4,94	60,87 (2)	6,50
pima	70,95	4,81	72,26 (17)	3,42	71,99 (4)	5,23	72,78 (16)	3,47	72,65 (25)	5,05
seg	90,76	-	92,19 (1)	-	92,19 (1)	-	92,14 (4)	-	91,71 (1)	-
wdbc	96,13	2,00	96,48 (2-3)	2,04	96,13 (1,3)	1,62	95,95 (9)	2,89	95,25 (1)	2,49
wine	92,68	3,81	94,93 (6-8)	4,11	94,93 (6-8,23-30)	4,11	96,63 (3)	3,91	96,08 (2)	4,59
NIST	95,75	-	95,95 (11)	-	95,86 (23-30)	-	95,23 (30)	-	94,89 (29)	-
LMD	67,89	3,67	68,22 (6)	1,35	67,56 (5)	1,95	68,22 (6)	0,69	68,00 (3)	2,03

Observou-se que em algumas das bases de dados (Tabela 4.19 e Tabela 4.20) houve um incremento na taxa de reconhecimento. Então, em alguns casos as informações de classe fornecidas pelo agrupamento foram úteis para o reconhecimento de mais instâncias de teste nessas bases de dados. Uma análise mais detalhada dos agrupamentos obtidos foi realizada no conjunto de 100 classificadores obtidos por subespaços aleatórios com *1-NN* na base de dados *NIST SD19*, que foi a mesma utilizada por [Ko *et al.*, 2008]. O objetivo da análise, primeiramente, foi verificar a predominância das classes em cada agrupamento.

A distribuição de frequência absoluta das classes reais na base de dados de validação foi calculada para cada agrupamento do método *K*-médias. Supunha-se que em cada agrupamento haveria uma classe dominante, entre as 10 possíveis (10 classes) da base de dados original. A Tabela 4.21 mostra exatamente isso na base *NIST SD19*. As linhas indicam a classe real das instâncias na base de dados de validação, e as colunas mostram os agrupamentos obtidos, numerados de 0 a 9. Por exemplo, no agrupamento 0 obtido pelo algoritmo *K*-médias, a classe real predominante é a 2, no agrupamento 1 é a classe 6 e assim por diante. Nota-se que nos agrupamentos 7 e 8, apesar de haver uma classe dominante, a diferença não é tão grande em relação a uma segunda classe dominante. O agrupamento 8, inclusive, teve apenas 7 instân-

cias na classe predominante. Provavelmente esses agrupamentos, especialmente o de número 8, constituem-se de instâncias de mais difícil categorização. Os agrupamentos 0 e 6 também não tiveram tanta diferença entre a classe predominante e as demais classes, como acontece nos agrupamentos 1, 2, 3, 4, 5 e 9.

Tabela 4.21: Frequência absoluta de instâncias de validação nos agrupamentos do método K -médias, categorizados por classes, para a base de dados *NIST SD19*

Classe real	Agrupamento ao qual pertence									
	0	1	2	3	4	5	6	7	8	9
0	3	30	0	7	922	8	1	0	0	29
1	24	25	4	40	0	29	6	55	3	814
2	460	43	2	0	0	471	7	4	2	11
3	371	2	0	15	4	535	21	34	4	14
4	0	16	861	47	0	4	13	21	4	34
5	151	16	2	742	0	74	1	1	7	6
6	1	937	5	24	1	2	0	0	5	25
7	3	0	0	8	0	2	624	342	3	18
8	5	29	17	403	244	7	112	63	2	118
9	3	0	56	20	6	1	532	377	0	5

Já a Tabela 4.22 mostra as taxas de reconhecimento obtidas para o KNORA-ELIMINATE, para o KNORA-UNION, para KLUNORA-ELIMINATE e para o KLUNORA-UNION na base de dados *NIST SD19* (com o conjunto de classificadores criados por subespaços aleatórios), para cada valor de k (1 a 30). As melhores taxas de reconhecimento para o KLUNORA apresentam-se em negrito: 92,03% ($k = 29$ e $k = 30$) para o KLUNORA-ELIMINATE e 88,16% ($k = 1$) para o KLUNORA-UNION. Esses valores foram inferiores aos obtidos pelo método KNORA-ELIMINATE e pelo método KNORA-UNION, visualizados na segunda e na terceira colunas da Tabela 4.22. O melhor desempenho para o KLUNORA-ELIMINATE foi obtido quando $k = 29$ e $k = 30$. Talvez, se fosse aumentado o valor do k , a taxa de reconhecimento poderia ser melhor.

Na Tabela 4.22, aplicando o teste de Friedman a um nível de confiança de 95% para os resultados em cada k em relação aos métodos, o valor de p obtido foi zero e o teste de múltiplas comparações com a correção de Dunn-Sidak mostrou que o KE foi melhor que o KU, que o KLUNORA-ELIMINATE e que KLUNORA-UNION; que o KU foi melhor que o KLUNORA-ELIMINATE e que o KLUNORA-UNION; e que o KLUNORA-ELIMINATE foi melhor que o KLUNORA-UNION. O KLUNORA foi pior que o KNORA. As mesmas conclusões foram obtidas com a correção de Bonferroni-Dunn.

Já a Tabela 4.23 apresenta os valores de z (Equação 4.2) obtidos no teste para diferenças entre proporções ([Dietterich *et al.*, 1995] [Dietterich, 1997]) da base de dados *NIST SD19* no conjunto inicial de classificadores formados por subespaços aleatórios, enquanto a Tabela 4.24 apresenta os valores de z obtidos com os métodos aplicados no conjunto inicial de classificadores formados por *bagging*.

Os valores de módulo de z das Tabelas 4.23 e 4.24 que são maiores do que 1,96 indicam diferenças a um nível de confiança de 95%. Assim, na Tabela 4.23 houve diferenças entre todos os métodos.

Já nos experimentos com o conjunto inicial criado utilizando *bagging* (Tabela 4.24) as diferenças foram entre KLUNORA-ELIMINATE e os três métodos utilizados para comparação (MAJ, KE e KU). O mesmo ocorreu com o KLUNORA-UNION. Também notaram-se diferenças entre os resultados do KLUNORA-ELIMINATE e do KLUNORA-UNION.

Tabela 4.22: Taxa de reconhecimento (em %) para k variando de 1 a 30 utilizando os métodos KNORA-ELIMINATE, KNORA-UNION, KLUNORA-ELIMINATE e KLUNORA-UNION no conjunto inicial de classificadores criados com subespaços aleatórios da base de dados *NIST SD19*

Valor de k	KNORA-ELIMINATE	KNORA-UNION	KLUNORA-ELIMINATE	KLUNORA-UNION
1	97,25	97,25	88,16	88,16
2	97,45	97,19	89,13	87,91
3	97,47	97,13	89,75	87,80
4	97,51	97,10	90,24	87,73
5	97,51	97,08	90,57	87,70
6	97,51	97,08	90,85	87,68
7	97,52	97,06	91,04	87,66
8	97,52	97,07	91,22	87,67
9	97,49	97,08	91,34	87,67
10	97,48	97,09	91,48	87,70
11	97,46	97,11	91,55	87,71
12	97,45	97,12	91,63	87,70
13	97,45	97,12	91,67	87,69
14	97,43	97,12	91,73	87,72
15	97,41	97,13	91,77	87,73
16	97,40	97,13	91,76	87,74
17	97,39	97,13	91,82	87,74
18	97,39	97,15	91,82	87,76
19	97,39	97,14	91,86	87,77
20	97,40	97,15	91,88	87,78
21	97,40	97,15	91,91	87,79
22	97,39	97,15	91,93	87,80
23	97,37	97,14	91,95	87,82
24	97,36	97,15	91,96	87,82
25	97,34	97,15	91,97	87,82
26	97,35	97,16	91,99	87,84
27	97,35	97,16	92,00	87,85
28	97,34	97,15	92,00	87,86
29	97,34	97,16	92,03	87,87
30	97,35	97,16	92,03	87,88

Para fins comparativos, também foi feito um experimento na base de dados *NIST SD19* onde a classificação era baseada apenas nos agrupamentos formados pelo algoritmo K -médias (sem o uso do conjunto de classificadores). Após a localização de qual agrupamento cada instância de teste pertence, foi atribuído à instância de teste a classe predominante (na base de validação) no agrupamento selecionado. A taxa de reconhecimento foi de 54,18%. Ou seja, o uso de subconjunto de classificadores com K -médias ainda foi melhor do que utilizar apenas a classe predominante no agrupamento ao qual cada uma das instâncias de teste pertence.

A seleção baseada em k oráculos mais próximos com informações de frequência relativa de classes em agrupamento, da maneira como foi abordada, não se apresentou promissora para a combinação de resultados no processo de classificação de sistemas baseados em múltiplos classificadores com SDSC. De todos os testes realizados, quando houve diferenças, o KLUNORA apresentou os piores resultados. Além disso, nos conjuntos iniciais criados por

Tabela 4.23: Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA e KLUNORA, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados *NIST SD19*

Método	MAJ	KE	KU	KLUNORA-ELIMINATE
KE	-12,40	-	-2,93	-42,76
KU	-9,50	2,93	-	-40,17
KLUNORA-ELIMINATE	31,40	42,76	40,17	-
KLUNORA-UNION	52,55	62,93	60,59	22,46

Tabela 4.24: Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA e KLUNORA, aplicado no conjunto inicial de classificadores formados por *bagging* da base de dados *NIST SD19*

Método	MAJ	KE	KU	KLUNORA-ELIMINATE
KE	-1,74	-	-0,79	-6,08
KU	-0,95	0,79	-	-5,29
KLUNORA-ELIMINATE	4,34	6,08	5,29	-
KLUNORA-UNION	7,06	8,79	8,01	2,72

bagging, analisando todas as bases de dados por meio do teste de Friedman, não foram notadas diferenças nas médias de taxa de reconhecimento.

A próxima seção tratará dos resultados de outra adaptação para o KNORA. Porém, agora serão utilizadas informações sobre proporção de classes reconhecidas.

4.4.3 Uso de Informações sobre Proporção de Classes Reconhecidas

Os experimentos nessa seção tratam do uso das probabilidades dos classificadores de acertarem cada uma das classes no método de k oráculos mais próximos. Primeiramente, na base de dados de validação, cada classificador obtém a porcentagem de acertos por classe. Posteriormente, no momento em que o classificador é escolhido para dar um voto para a classe que prediz, o voto tem o peso dessa porcentagem de acertos. Isso foi testado em adaptações do KNORA-ELIMINATE e do KNORA-UNION, denominadas KNORA-ELIMINATE-P e KNORA-UNION-P, respectivamente.

As melhores médias de taxa de reconhecimento obtidas (e respectivos valores de desvio padrão e de k), em cada base de dados, para os métodos MAJ (voto majoritário entre todos os classificadores do conjunto inicial), KNORA-ELIMINATE (KE), KNORA-UNION (KU), KNORA-ELIMINATE-P (KE-P) e KNORA-UNION-P (KU-P) são apresentadas na Tabela 4.25 (conjunto de classificadores criados utilizando subespaços aleatórios e 1 -NN) e na Tabela 4.26 (conjunto de classificadores criados utilizando *bagging* e 1 -NN). O valor do parâmetro k dos métodos variou de 1 a 30. Os melhores resultados entre os 5 métodos das tabelas apresentam-se em negrito.

O teste não paramétrico de Friedman [Demsar, 2006] [Hollander & Wolfe, 1999] foi utilizado para avaliar a significância dos resultados. Primeiramente, apenas os valores das médias das taxas de reconhecimento obtidas pelos métodos MAJ, KE, KU, KNORA-ELIMINATE-P (KE-P) e KNORA-UNION-P (KU-P) foram analisados nas 8 bases de dados. O valor obtido para p foi 0,0999 nos experimentos de subespaços aleatórios e de 0,00048 nos experimentos de *bagging*. Então, quanto à Tabela 4.25, não se confirmou resultado significativo a esse nível de confiança, ou seja, não se pôde tirar conclusões sobre quais métodos se sobressaem em todas as bases de dados. No entanto, pôde-se confirmar um resultado significativo com nível de confiança de 95% nos resultados da Tabela 4.26.

Um teste de múltiplas comparações utilizando a correção de Dunn-Sidak foi aplicada aos valores críticos após o teste de Friedman. Assim, foram determinados quais pares de algoritmos eram significativamente diferentes. No caso da Tabela 4.26, houve diferença significativa entre os valores das médias das taxas de reconhecimento dos métodos MAJ e KNORA-ELIMINATE, e dos métodos MAJ e KNORA-ELIMINATE-P. Porém, esses resultados foram melhores que MAJ devido à metodologia de seleção do KNORA-ELIMINATE, e não devido ao uso de proporção de classes por classificador. Se a proporção de classes por classificador

Tabela 4.25: Melhores médias de taxa de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando probabilidades de classe por classificador como peso no método de oráculos mais próximos no conjunto inicial de classificadores formados por subespaços aleatórios

Base de dados	MAJ		KE(k)		KU(k)		KE-P(k)		KU-P(k)	
	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ
letter	95,18	0,49	95,48	0,34	95,60	0,39	95,54	0,33	95,63	0,29
			(1)		(3)		(1)		(2)	
liver	65,82	3,29	65,49	4,80	68,12	3,54	68,43	6,89	67,55	7,36
			(10)		(7)		(1)		(1)	
pima	73,44	3,69	72,13	3,74	74,35	3,61	72,00	3,72	73,83	3,13
			(2)		(6)		(2)		(25)	
seg	88,33	-	88,33	-	88,33	-	88,33	-	88,33	-
			(1-30)		(1-30)		(1)		(1)	
wdbc	96,13	1,84	96,48	2,05	96,48	2,05	96,48	2,05	96,48	2,05
			(1)		(1)		(1)		(1)	
wine	94,41	4,54	97,75	2,91	96,67	4,68	97,19	3,96	96,63	3,91
			(2,3)		(2,3)		(2)		(1)	
NIST	96,28	-	97,52	-	97,25	-	97,48	-	97,17	-
			(7,8)		(1)		(8)		(1)	
LMD	69,22	1,39	68,89	1,95	70,00	1,00	68,89	1,71	70,11	1,58
			(2)		(28,29)		(2)		(15)	

tivesse alterado o resultado, seria apontada também a diferença entre o KNORA-ELIMINATE e o KNORA-ELIMINATE-P. Usando a correção de Bonferroni-Dunn, as conclusões foram as mesmas.

Tabela 4.26: Melhores médias de taxa de reconhecimento (em %) e respectivos valores de desvio padrão (σ) e de k utilizando probabilidades de classe por classificador como peso no método de oráculos mais próximos no conjunto inicial de classificadores formados por *bagging*

Base de dados	MAJ		KE(k)		KU(k)		KE-P(k)		KU-P(k)	
	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ
letter	93,79	0,51	94,34	0,46	94,17	0,42	94,45	0,44	94,21	0,44
			(1)		(5)		(1)		(3)	
liver	57,40	6,18	60,30	6,70	59,42	7,02	61,44	5,58	60,86	6,55
			(26)		(3)		(1)		(3)	
pima	70,95	4,81	72,26	3,42	71,99	5,23	72,26	3,42	71,61	4,55
			(17)		(4)		(17)		(2)	
seg	90,76	-	92,19	-	92,19	-	92,05	-	92,05	-
			(1)		(1)		(1)		(1)	
wdbc	96,13	2,00	96,48	2,04	96,13	1,62	96,48	2,04	96,31	1,94
			(2-3)		(1,3)		(2)		(13)	
wine	92,68	3,81	94,93	4,11	94,93	4,11	94,93	4,11	94,38	4,54
			(6-8)		(6-8,23-30)		(6)		(1)	
NIST	95,75	-	95,95	-	95,86	-	95,84	-	95,78	-
			(11)		(23-30)		(12)		(6)	
LMD	67,89	3,67	68,22	1,35	67,56	1,95	69,11	1,07	68,33	3,18
			(6)		(5)		(6)		(11)	

Já a Tabela 4.27 apresenta os valores de z (Equação 4.2) obtidos no teste para diferenças entre proporções ([Dietterich *et al.*, 1995] [Dietterich, 1997]) da base de dados *NIST SD19* no conjunto inicial de classificadores formados por subespaços aleatórios, enquanto a Tabela 4.28 apresenta os valores de z obtidos com os métodos aplicados no conjunto inicial de classificadores formados por *bagging*.

Os valores de módulo de z das Tabelas 4.27 e 4.28 que são maiores do que 1,96 indicam diferenças a um nível de confiança de 95%. Assim, na Tabela 4.27 não houve diferenças apenas entre os métodos KE e KE-P, e entre os métodos KU e KU-P.

Já nos experimentos com o conjunto inicial criado utilizando *bagging* (Tabela 4.28) não foram percebidas diferenças. Não se encontraram diferenças a esse nível de confiança entre KE e KE-P, e nem entre KU e KU-P. Ou seja, informações de proporções não se demonstraram interessantes para o KE e o KU tanto nos experimentos com conjunto inicial criado por subespaços aleatórios quanto nos experimentos com o *bagging*.

Tabela 4.27: Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA e KNORA-P, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados *NIST SD19*

Método	MAJ	KE	KU	KE-P
KE	-12,40	-	-2,93	-0,44
KU	-9,50	2,93	-	2,49
KE-P	-11,96	0,44	-2,49	-
KU-P	-8,67	3,77	0,84	3,33

Tabela 4.28: Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA e KNORA-P, aplicado no conjunto inicial de classificadores formados por *bagging* da base de dados *NIST SD19*

Método	MAJ	KE	KU	KE-P
KE	-1,74	-	-0,79	-0,96
KU	-0,95	0,79	-	-0,17
KE-P	-0,78	0,96	0,17	-
KU-P	-0,26	1,48	0,69	0,52

O uso de probabilidade de reconhecimento de classes por classificador como peso nos votos dos subconjuntos escolhidos por k oráculos mais próximos (método KNORA) não apresentou resultados promissores na tentativa de atingir o oráculo do conjunto inicial de classificadores.

Então, a próxima seção apresentará os resultados obtidos com a primeira nova abordagem proposta nesse trabalho, que trata da fusão do KNORA com o k -NN embutido no método.

4.5 Fusão do Método KNORA com o Método k -NN

Os experimentos apresentados nessa seção avaliam diferentes estratégias que combinam os resultados do método KNORA com os resultados do k -NN embutido no método. As bases de dados e os conjuntos iniciais de classificadores utilizados são os mesmos descritos na Seção 4.1.

No KNORA, o k -NN era utilizado apenas para definir os vizinhos de uma instância de teste durante a criação do conjunto de classificadores, sendo que os classificadores selecionados eram aqueles que acertavam os vizinhos de acordo com as especificações dos esquemas do KNORA. Aqui, além do k -NN definir os vizinhos, também pode fornecer votos (a classe que possuem) no processo de classificação. Isso é utilizado como informação adicional, de acordo com as 5 estratégias detalhadas na Seção 3.2.

As melhores médias de taxa de reconhecimento e respectivos valores de desvio padrão obtidos nos conjuntos de classificadores criados com subespaços aleatórios são apresentados nas Tabelas 4.29 e 4.30, enquanto que os obtidos nos conjuntos de classificadores criados com *bagging* são apresentados nas Tabelas 4.31 e 4.32. O valor do parâmetro k variou de 1 a 30,

e o parâmetro Y (%) para os esquemas condicionais foram avaliados de 10% a 100%, com incremento de 10%. O melhor resultado para cada base de dados está em negrito.

As estratégias KNORA CU, KNN+KNORA CF, KNN CU, KNORA+KNN CF e KNN+KNORA UF, apresentadas anteriormente na Seção 3.2, foram aplicadas com o KNORA-UNION ou com o KNORA-ELIMINATE. Portanto, as abreviações utilizadas nessa seção indicam ou o uso do KNORA-UNION, citado como KNORA(UNION) ou KU, ou o uso do KNORA-ELIMINATE, citado como KNORA(ELIMINATE) ou KE.

Comparando as Tabelas 4.29 e 4.30 com a Tabela 4.2 (MAJ, KE e KU) da página 53, observa-se que pelo menos uma das estratégias propostas (KNORA CU, KNN+KNORA CF, KNN CU, KNORA+KNN CF ou KNN+KNORA UF) apresentou melhor taxa de reconhecimento que o método 1 -NN, que o voto majoritário simples de todos os classificadores do conjunto inicial, que o KNORA-ELIMINATE original e que o KNORA-UNION original, embora o valor do oráculo não tenha sido atingido. O mesmo aconteceu comparando as Tabelas 4.31 e 4.32 com a Tabela 4.3 (página 54).

Foi utilizado o teste de Friedman a um nível de confiança de 95% para comparar os métodos MAJ, KE, KU e os métodos de fusão do KNORA com o k -NN. Para o conjunto inicial criado com subespaços aleatórios, o valor de p foi zero, indicando que houve diferenças significativas. Cada célula marcada com um X na Tabela 4.33 (página 78) indica onde foram encontradas diferenças aplicando o teste de múltiplas comparações utilizando correção de Dunn-Sidak e de Bonferroni-Dunn.

Desses resultados (Tabela 4.33), os métodos que superaram tanto o MAJ, quanto o KNORA-ELIMINATE, quanto o KNORA-UNION nos conjuntos iniciais de classificadores criados com subespaços aleatórios foram o KU CU e o KNN+KU CF.

No conjunto inicial criado com *bagging*, o valor de p também foi de zero, indicando que houve diferenças significativas. A Tabela 4.34 (página 78) mostra entre quais métodos essas diferenças foram notadas utilizando Bonferroni-Dunn. Já para a correção de Dunn-Sidak, as diferenças apresentadas foram as mesmas de Bonferroni-Dunn, excetuando-se a diferença entre os métodos KNN+KE CF e KU+KNN CF. Os métodos KE CU, KNN+KE CF, KU CU e KNN+KU CF superaram no conjunto inicial de classificadores criados com *bagging* os 3 métodos (MAJ, KE e KU) com os quais foram comparados todos os resultados desse capítulo.

Os dois métodos que apresentaram incrementos significativos em relação ao MAJ, ao KE, e ao KU na taxa de reconhecimento tanto para conjuntos iniciais criados por subespaços aleatórios quanto para conjuntos iniciais criados por *bagging* foram o KU CU e o KNN+KU CF.

No método KU CU, primeiramente é executado o k -NN. Se menos que Y % de todos os vizinhos da instância de teste pertencem a mesma classe predominante, é executado o KNORA-UNION. Então, somente a saída do KNORA-UNION é utilizada. Caso contrário, utiliza-se apenas o k -NN.

Já o método KNN+KU CF diferencia-se do método KU CU pelo fato de utilizar a combinação das saídas do k -NN com as saídas do KNORA-UNION no lugar do uso apenas do KNORA-UNION. Assim, primeiramente executa-se o k -NN. Se menos que Y % de todos os vizinhos da instância de teste pertencem à mesma classe predominante, combinam-se as saídas do k -NN com as saídas do KNORA-UNION. Caso contrário, utiliza-se somente a saída do algoritmo k -NN.

Tabela 4.29: Melhores médias de taxa de reconhecimento (TR) e valores de desvio padrão correspondentes (σ) obtidos pelos esquemas condicionais de fusão do KNORA com o k -NN nos conjuntos iniciais de classificadores criados utilizando subespaços aleatórios

Base de dados	Execução de KNORA após execução do k -NN								Execução de k -NN após execução do KNORA								
	KE CU (k ;%)		KNN+KE CF(k ;%)		KU CU (k ;%)		KNN+KU CF(k ;%)		KNN(após KE) CU(k ;%)		KE+KNN CF(k ;%)		KNN(após KU) CU(k ;%)		KU+KNN CF(k ;%)		
	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ	
letter	96,22 (2;60-100)	0,27	96,38 (2;60-100)	0,31	96,29 (2;60-100)	0,25	(2;60-100)	96,34 (2;60-100)	0,25	96,36 (2;60-100)	0,32	96,17 (2;80-100)	0,34	96,23 (3;60)	0,24	95,98 (2;70)	0,28
liver	68,09 (20;70)	5,98	66,97 (2;60-100)	9,31	69,54 (12;70)	3,38	69,26 (12;70)	2,97	67,29 (1;70)	8,52	67,85 (2;70-100)	7,38	68,42 (13;60)	4,28	68,69 (2;70-100)	4,87	
pima	74,87 (5;70-80)	4,56	74,21 (16;60-100)	4,49	75,66 (6;70-80)	3,84	75,65 (21;80)	3,89	72,91 (3;80)	4,08	73,17 (3;80)	4,69	75,52 (27;70)	4,12	74,61 (21;60-100)	4,01	
seg	92,48 (1;10-100)	-	92,48 (1;10-100)	-	92,48 (1;10-100)	-	92,48 (1;10-100)	-	92,86 (1;80)	-	92,14 (90-100)	-	92,86 (1;80)	-	89,67 (1;60-100)	-	
wdbc	96,84 (2;60-100)	2,45	96,84 (2;60-100)	2,45	96,66 (2;60-100)	2,27	96,66 (2;60-100)	2,27	96,84 (1;70)	1,81	97,01 (2;80-100)	2,50	96,84 (1;70)	1,81	96,65 (1;60)	1,77	
wine	98,30 (2;60-100)	3,79	98,30 (2;60-100)	3,79	98,33 (11-14;60)	3,75	98,33 (8-11;70)	3,75	98,30 (4;80-100)	3,79	98,30 (7;60-100)	3,79	98,30 (7;60-100)	3,79	97,75 (21-25;70)	3,88	
NIST	97,66 (4;80-100)	-	97,73 (4;80)	-	97,54 (2;70-100)	-	97,54 (2;70-100)	-	97,58 (7;50)	-	97,70 (7;70-100)	-	97,48 (1;60)	-	97,30 (1;60-100)	-	
LMD	71,00 (2;60-100)	0,33	71,56 (3;70-100)	1,84	71,89 (13;60)	3,01	72,33 (13;60)	3,33	70,11 (5;70)	1,84	70,44 (2;80-100)	1,35	72,22 (6;70)	1,02	70,78 (19;50)	1,02	

Tabela 4.30: Melhores médias de taxa de reconhecimento (TR) e valores de desvio padrão correspondentes (σ) obtidos pelos esquemas incondicionais de fusão do KNORA com o k -NN nos conjuntos iniciais de classificadores criados utilizando subespaços aleatórios

Base de dados	Incondicional			
	KNN+KE UF(k)		KNN+KU UF(k)	
	TR	σ	TR	σ
letter	96,34(3)	0,27	96,03(2)	0,31
liver	68,43(2)	6,88	68,98(2)	5,08
pima	74,21(10)	4,76	74,61(21)	4,01
seg	92,14(7)	-	89,67(1)	-
wdbc	97,01(2)	2,50	96,48(6)	2,21
wine	98,30(2)	3,79	97,78(2,3)	3,88
NIST	97,74(6)	-	97,30(1)	-
LMD	71,44(3)	0,64	70,67(19)	1,00

Tabela 4.31: Melhores médias de taxa de reconhecimento (TR) e valores de desvio padrão correspondentes (σ) obtidos pelos esquemas condicionais de fusão do KNORA com o k -NN nos conjuntos iniciais de classificadores criados utilizando *bagging*

Base de dados	Execução de KNORA após execução do k -NN								Execução de k -NN após execução do KNORA							
	KE CU (k ;%)		KNN+KE CF(k ;%)		KU CU (k ;%)		KNN+KU CF(k ;%)		KNN(após KE) CU(k ;%)		KE+KNN CF(k ;%)		KNN(após KU) CU(k ;%)		KU+KNN CF(k ;%)	
	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ
letter	95,25 (2;60-100)	0,44	95,37 (2;60-100)	0,49	95,30 (2;60-100)	0,43	95,36 (2;60-100)	0,48	95,25 (1;80)	0,49	94,96 (3;80-100)	0,49	94,76 (3;70)	0,51	94,57 (2;70-100)	0,62
liver	67,27 (14;60)	9,89	66,38 (29;10-50)	8,32	66,38 (29;10-50)	8,32	66,38 (29;10-50)	8,32	62,33 (15;80-100)	8,72	62,33 (15;80-100)	8,72	65,81 (29;100)	10,09	60,02 (3;60-100)	6,44
pima	75,13 (11;70)	3,03	74,47 (6;70-100)	4,32	74,86 (11;70)	3,52	74,99 (30;60)	5,04	73,44 (8;90-100)	3,86	73,31 (7;90-100)	4,43	75,00 (18;90)	3,20	72,65 (5;60)	4,82
seg	92,48 (1;10-100)	-	92,48 (1;10-100)	-	92,48 (1;10-100)	-	92,48 (1;10-100)	-	93,14 (2;100)	-	92,76 (7;80-100)	-	92,48 (1;90-100)	-	92,24 (1;50-100)	-
wdbc	96,83 (2;60-100)	2,00	97,01 (3;70-100)	2,36	97,01 (7;60-70)	1,87	97,18 (13;70)	1,71	96,83 (7;60)	1,63	96,83 (7;60)	1,63	96,83 (11;70)	1,63	96,31 (1;60)	1,31
wine	97,78 (14;50)	3,88	97,78 (14;50)	3,88	97,78 (14;50)	3,88	97,78 (14;50)	3,88	96,63 (11; 80-100)	3,91	96,63 (11; 80-100)	3,91	97,22 (14; 90-100)	3,93	94,93 (23-30; 10-100)	4,11
NIST	97,53 (2;60-100)	-	97,55 (2;60-100)	-	97,53 (2;60-100)	-	97,54 (2;60-100)	-	97,11 (1;100)	-	96,91 (9-10;100)	-	96,98 (5;100)	-	96,16 (28-30; 90-100)	-
LMD	70,56 (6;70)	2,12	71,78 (5;80)	2,36	70,44 (6;70)	2,87	70,89 (6;70)	2,83	70,44 (8;80)	1,58	70,56 (6;90,100)	0,77	70,78 (6;80)	0,51	68,56 (6;60-100)	2,01

Tabela 4.32: Melhores médias de taxa de reconhecimento (TR) e valores de desvio padrão correspondentes (σ) obtidos pelos esquemas incondicionais de fusão do KNORA com o k -NN nos conjuntos iniciais de classificadores criados utilizando *bagging*

Base de dados	Incondicional			
	KNN+KE UF(k)		KNN+KU UF(k)	
	TR	σ	TR	σ
letter	95,16(3)	0,51	94,84(1)	0,59
liver	65,49(23)	6,95	60,85(2)	9,84
pima	74,34(6)	4,56	72,52(5)	4,60
seg	92,24(1)	-	92,24(1)	-
wdbc	96,83(3-4)	2,17	96,48(2)	2,04
wine	97,19(13,16-19)	3,96	95,49(1)	4,40
NIST	97,50(11)	-	96,54(1)	-
LMD	71,44(5)	1,92	68,89(2)	1,90

Tabela 4.33: Métodos que mostraram diferenças utilizando correção de Dunn-Sidak e de Bonferroni-Dunn no teste de múltiplas comparações com os métodos MAJ, KNORA e de fusão do k -NN com o KNORA, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios

Método	MAJ	KE	KU
KE CU	X		
KNN+KE CF	X		
KU CU	X	X	X
KNN+KU CF	X	X	X
KNN(após KE) CU			
KE+KNN CF			
KNN(após KU) CU	X		
KU+KNN CF			
KNN+KE UF	X	X	
KNN+KU UF			

Tabela 4.34: Métodos que mostraram diferenças utilizando correção de Bonferroni-Dunn no teste de múltiplas comparações com os métodos MAJ, KNORA e de fusão do k -NN com o KNORA, aplicado no conjunto inicial de classificadores formados por *bagging*

Método	MAJ	KE	KU	KU+KNN CF
KE CU	X	X	X	
KNN+KE CF	X	X	X	X
KU CU	X	X	X	
KNN+KU CF	X	X	X	X
KNN(após KE) CU	X			
KE+KNN CF				
KNN(após KU) CU	X		X	
KU+KNN CF				
KNN+KE UF	X			
KNN+KU UF				

Os métodos KE+KNN CF, KU+KNN CF e KNN+KU UF não foram indicados em nenhum dos testes com múltiplas comparações como melhores. Os demais, utilizando ou a correção de Dunn-Sidak ou a correção de Bonferroni-Dunn, mostraram-se significativamente melhores pelo menos em um caso.

Uma análise mais detalhada dos resultados do conjunto de 100 classificadores obtidos por subespaços aleatórios com k -NN na base de dados *NIST SD19* (a mesma utilizada por [Ko *et al.*, 2008]) foi realizada. A Tabela 4.35 e a Tabela 4.36 apresentam as melhores taxas de reconhecimento para as 5 estratégias de fusão propostas. Conforme já citado, o valor do parâmetro k variou de 1 a 30, e o parâmetro Y para os esquemas condicionais foram avaliados de 10% a 100%, com incremento de 10%.

Os resultados de cada estratégia para a base de dados *NIST SD19* são apresentados de acordo com a seguinte sequência:

- KNORA CU: A Figura 4.2 e a Figura 4.3 mostram os resultados quando usando esse esquema com KNORA-ELIMINATE e KNORA-UNION, respectivamente. A melhor taxa de reconhecimento foi 97,66% (com $Y = 80%$ ou mais) quando utilizando o KNORA-ELIMINATE, e 97,54% (com $Y = 70%$ ou mais) quando utilizando KNORA-UNION. Como pode-se observar, ambos os casos têm melhores resultados que o KNORA sozinho (ELIMINATE 97,52% e UNION 97,25%).
- KNN+KNORA CF: Nesse esquema também observaram-se melhores resultados que o uso do KNORA-ELIMINATE e que o uso do KNORA-UNION sozinhos. Nas Figuras 4.4

Tabela 4.35: Melhores taxas de reconhecimento em % com seus valores de (k) correspondentes para os esquemas condicionais da fusão do KNORA com o k -NN, aplicados no conjunto inicial de classificadores criados com subespaços aleatórios na base de dados *NIST SD19*

Esquema de Fusão	Y%									
	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
KE CU	97,40 (3)	97,40 (3)	97,40 (3)	97,42 (3)	97,40 (3)	97,64 (2)	97,64 (3)	97,66 (4)	97,66 (4)	97,66 (4)
KU CU	97,40 (3)	97,40 (3)	97,40 (3)	97,42 (3)	97,42 (3)	97,48 (4)	97,54 (2)	97,54 (2)	97,54 (2)	97,54 (2)
KNN+KE CF	97,40 (3)	97,40 (3)	97,40 (3)	97,42 (3)	97,42 (3)	97,65 (2)	97,70 (3)	97,73 (4)	97,73 (6,7)	97,73 (6,7)
KNN+KU CF	97,40 (3)	97,40 (3)	97,40 (3)	97,42 (3)	97,42 (3)	97,48 (4)	97,54 (2)	97,54 (2)	97,54 (2)	97,54 (2)
KNN (após KE) CU	97,52 (7,8)	97,52 (7,8)	97,53 (8)	97,53 (8)	97,58 (7)	97,55 (3,5)	97,46 (3)	97,43 (1)	97,40 (1,3)	97,38 (3)
KNN (após KU) CU	97,25 (1)	97,25 (1)	97,27 (1)	97,34 (1)	97,39 (1)	97,48 (1)	97,43 (1)	97,43 (3)	97,41 (3)	97,40 (3)
KE+KNN CF	97,52 (7,8)	97,52 (7,8)	97,53 (8)	97,56 (4)	97,62 (6)	97,69 (5)	97,70 (7)	97,70 (7)	97,70 (7)	97,70 (7)
KU+KNN CF	97,25 (1)	97,25 (1)	97,26 (1)	97,28 (1)	97,29 (1)	97,30 (1)	97,30 (1)	97,30 (1)	97,30 (1)	97,30 (1)

Tabela 4.36: Melhores taxas de reconhecimento em % com seus valores de (k) correspondentes para os esquemas incondicionais da fusão do KNORA com o k -NN, aplicados no conjunto inicial de classificadores criados com subespaços aleatórios na base de dados *NIST SD19*

Esquema	Taxa de Reconhecimento
KNN+KE UF	97,74%(6)
KNN+KU UF	97,30%(1)

e 4.5, pode-se ver as taxas de reconhecimento para esse esquema utilizando KNORA-ELIMINATE e KNORA-UNION, respectivamente. As melhores taxas de reconhecimento foram 97,73% (com $Y = 80%$ ou mais) quando usando KNORA-ELIMINATE, e 97,54% (com $Y = 70%$ ou mais) quando usando KNORA-UNION.

- KNN CU: As Figuras 4.6 e 4.7 apresentam os resultados executando k -NN (depois de KNORA-ELIMINATE) CU e k -NN (depois de KNORA-UNION) CU, respectivamente. A melhor taxa de reconhecimento foi 97,58% (com $Y = 50%$) quando usando k -NN (depois de KNORA-ELIMINATE) CU, que foi melhor que KNORA-ELIMINATE sozinho (97,52%). A melhor taxa de reconhecimento foi 97,48% (com $Y = 60%$) quando usando k -NN (depois de KNORA-UNION) CU, que foi melhor que KNORA-UNION sozinho (97,25%).
- KNORA+KNN CF: A Figura 4.8 apresenta os resultados do KE+KNN CF (fusão condicional das saídas do KNORA-ELIMINATE com as saídas do k -NN). Já a Figura 4.9 apresenta os resultados do KU+KNN CF (fusão condicional das saídas do KNORA-UNION com as saídas do k -NN). As melhores taxas de reconhecimento foram 97,70% (com $Y = 70%$ ou mais) quando utilizando KNORA-ELIMINATE, e 97,30% (com $Y = 60%$ ou mais) quando utilizando KNORA-UNION.
- KNN+KNORA UF: Na Figura 4.10 são apresentados as taxas de reconhecimento obtidas para o KNN+KNORA(UNION) UF e o KNN+KNORA(ELIMINATE) UF, juntamente com as obtidas pelo KNORA-UNION e pelo KNORA-ELIMINATE. A melhor taxa de reconhecimento observada para o KNN+KNORA(ELIMINATE) UF, que é a fusão in-

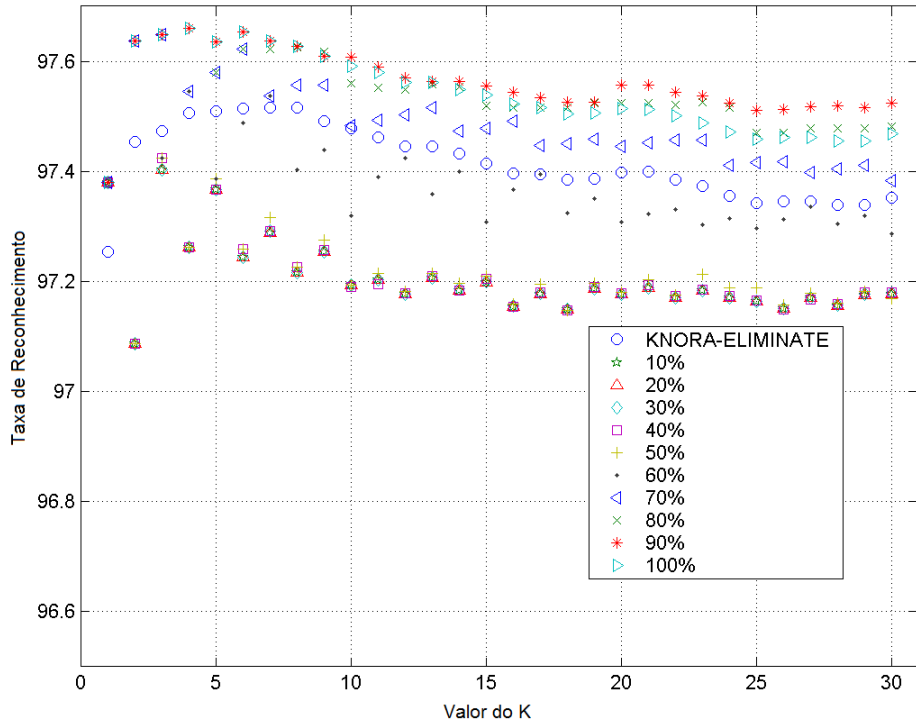


Figura 4.2: Taxas de reconhecimento na estratégia KNORA(ELIMINATE) CU

condicional das saídas do k -NN com as do KNORA-ELIMINATE, foi 97,74% ($k = 6$). Já para o KNN+KNORA (UNION) UF, que é a fusão incondicional das saídas do k -NN com as do KNORA-UNION, a melhor taxa foi de foi 97,30% ($k = 1$).

O melhor resultado de todos os experimentos no conjunto inicial de classificadores criados por subespaços aleatórios na base de dados *NIST SD19* foi 97,74% ($k = 6$), atingido pelo KNN+KNORA (ELIMINATE) UF. Obtiveram-se resultados melhores que 93,34% (k -NN baseado em 132 características com $k = 1$), do que 96,28% (combinação de todos os classificadores k -NN baseados em 32 características, disponíveis no conjunto inicial), do que 97,25% (KNORA-UNION com $k = 1$) e do que 97,52% (KNORA-ELIMINATE com $k = 7$ e $k = 8$).

Na maioria das figuras citadas (Figura 4.2, Figura 4.3, Figura 4.4, Figura 4.5, Figura 4.6, Figura 4.7, Figura 4.8, Figura 4.9 e Figura 4.10), pode-se observar que o melhor resultado aparece quando o k tem o valor em torno de 7 ou 8. Uma explicação para isso pode estar no fato de que os experimentos na base de dados *NIST SD19* utilizaram vizinhos delimitados (Seção 4.1.1, na página 51).

O desempenho do oráculo (99,95%) na base de dados *NIST SD19* (conjunto inicial de classificadores criados com subespaços aleatórios) não foi atingido. Entretanto, mostrou-se que pode haver melhorias nos resultados do KNORA considerando informações adicionais do k -NN usado para selecionar os subconjuntos.

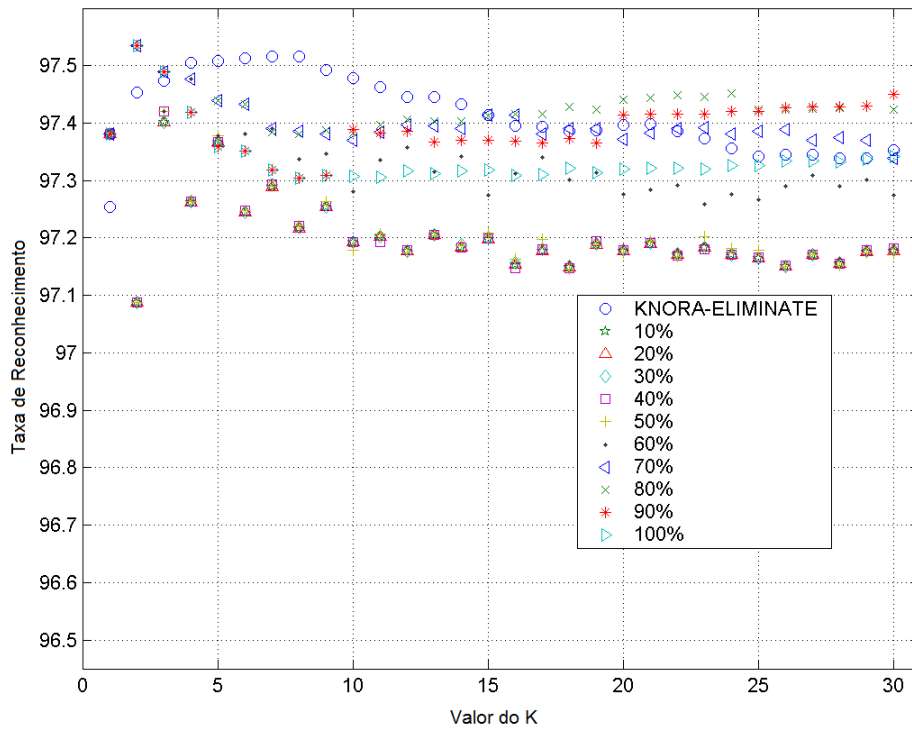


Figura 4.3: Taxas de reconhecimento na estratégia KNORA(UNION) CU

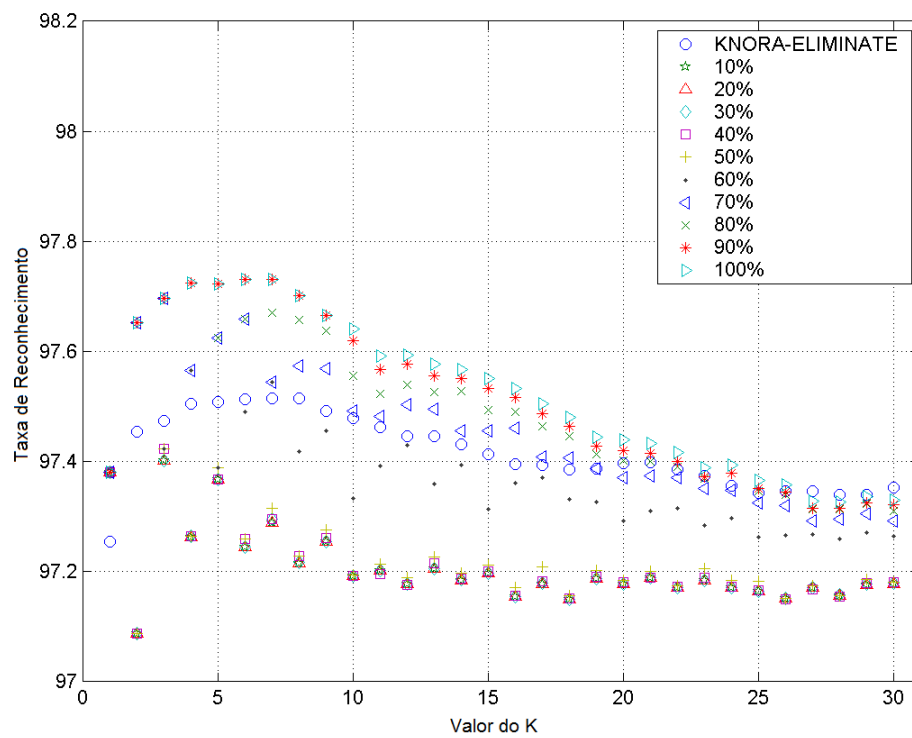


Figura 4.4: Taxas de reconhecimento na estratégia KNN+KNORA(ELIMINATE) CF

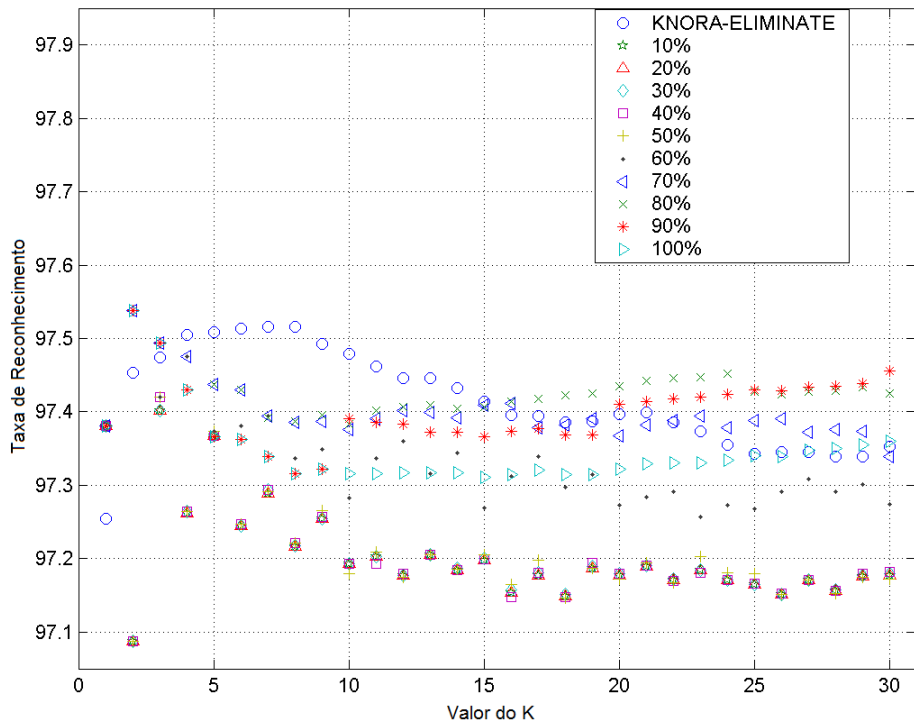


Figura 4.5: Taxas de reconhecimento na estratégia KNN+KNORA(UNION) CF

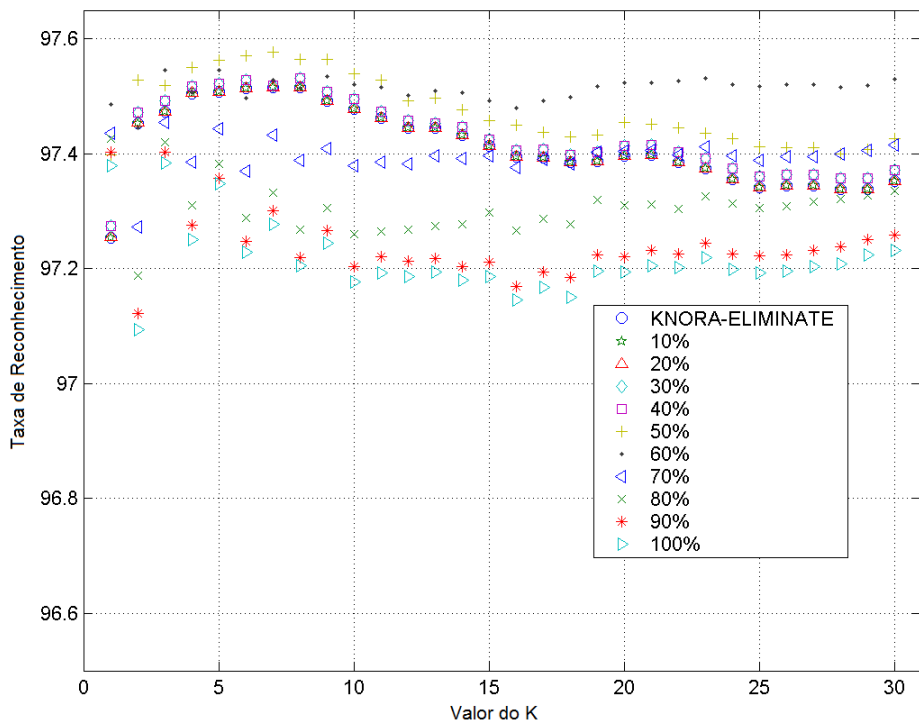


Figura 4.6: Taxas de reconhecimento no KNN (após KNORA-ELIMINATE) CU

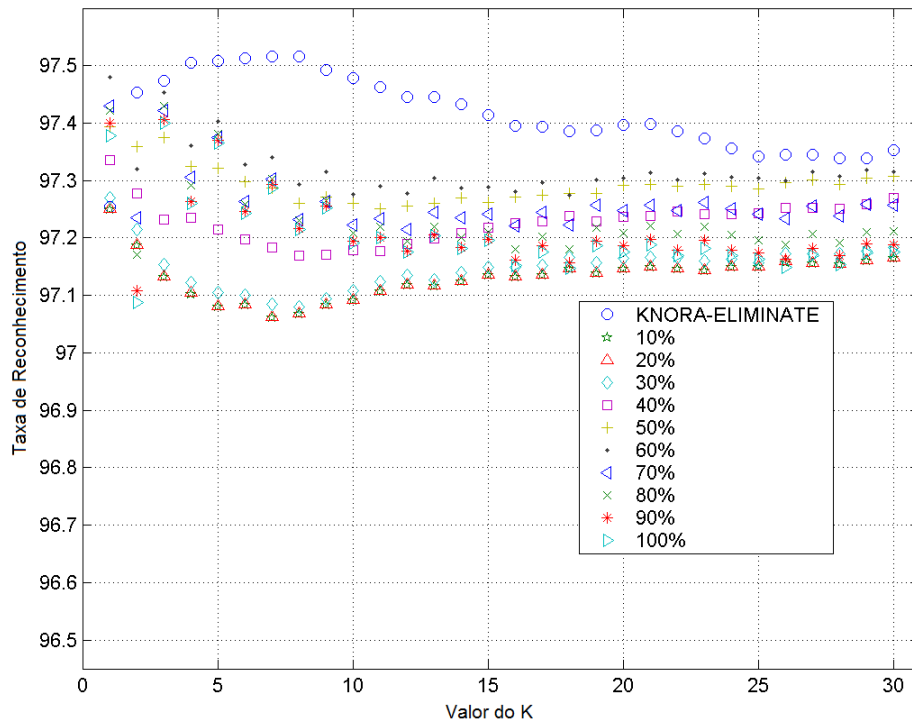


Figura 4.7: Taxas de reconhecimento no KNN (após KNORA-UNION) CU

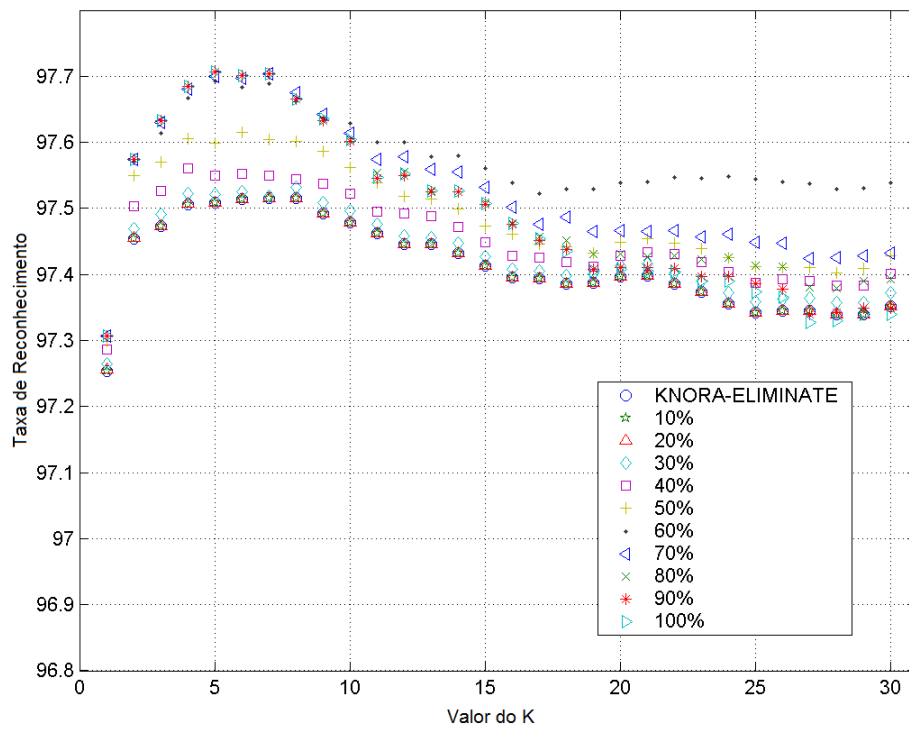


Figura 4.8: Taxa de reconhecimento utilizando a estratégia KE+KNN CF

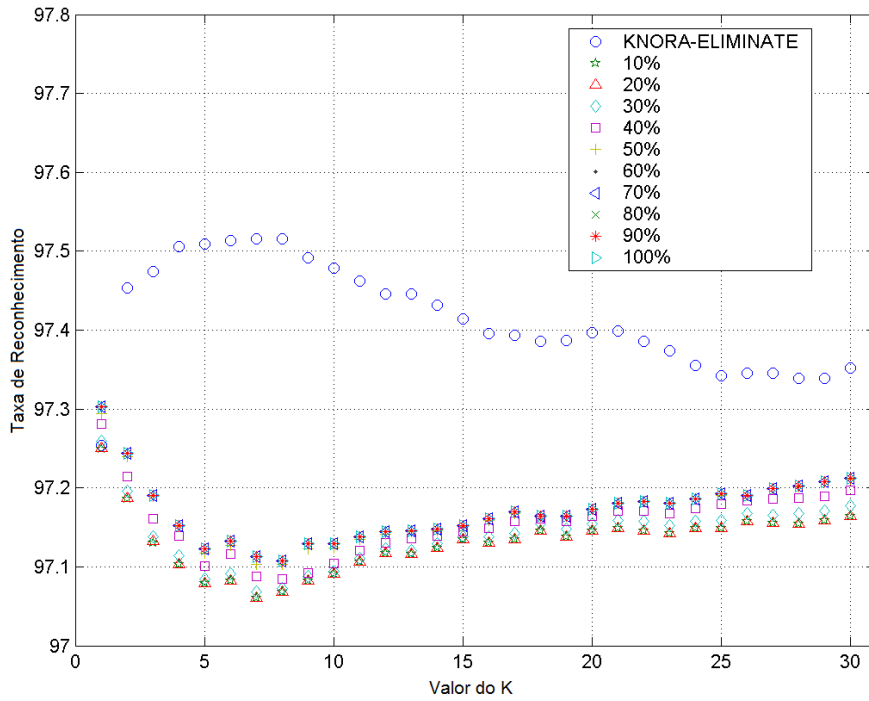


Figura 4.9: Taxa de reconhecimento utilizando a estratégia KU+KNN CF

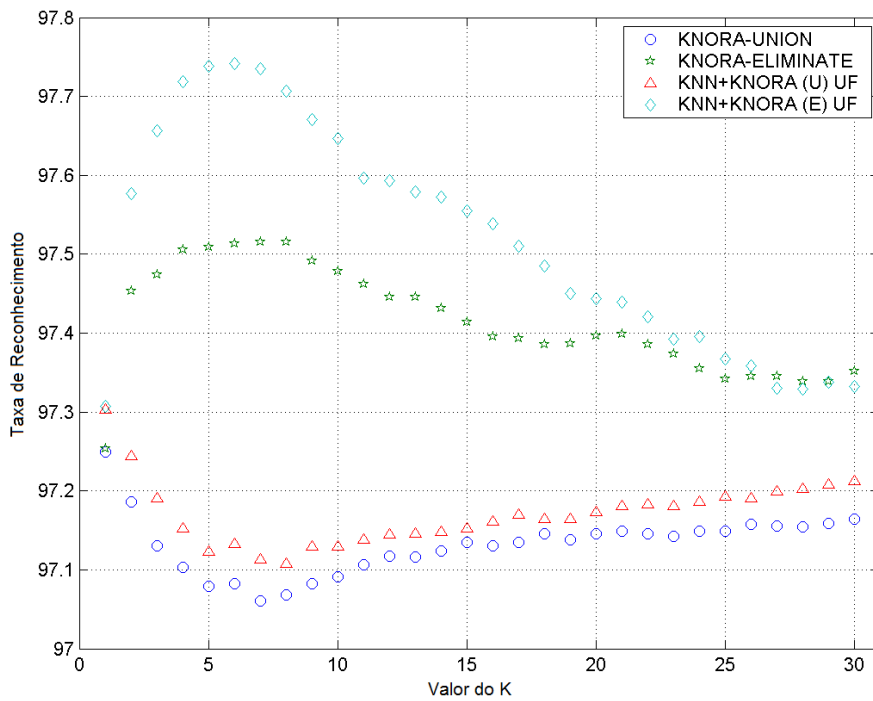


Figura 4.10: Taxas de reconhecimento para KNORA-UNION, KNORA-ELIMINATE, KNN+KNORA(UNION) UF e KNN+KNORA(ELIMINATE) UF

A Tabela 4.37 apresenta os valores de z (Equação 4.2) obtidos no teste para diferenças entre proporções ([Dietterich *et al.*, 1995] [Dietterich, 1997]) da base de dados *NIST SD19* no conjunto inicial de classificadores formados por subespaços aleatórios, enquanto a Tabela 4.39 apresenta os valores de z obtidos com os métodos aplicados no conjunto inicial de classificadores formados por *bagging*.

Os valores de módulo de z das Tabelas 4.37 e 4.39 que são maiores do que 1,96 indicam diferenças a um nível de confiança de 95%. Assim, na Tabela 4.38, as células marcadas com um X representam onde houve diferenças nos valores das Tabela 4.37 a esse nível de confiança. O mesmo ocorre com a Tabela 4.40, que mostra os métodos que apresentaram diferenças de acordo com os valores da Tabela 4.39.

Tanto na Tabela 4.38 quanto na Tabela 4.40, o MAJ apresentou diferenças com os 10 esquemas de fusão do KNORA e do seu k -NN embutido (5 estratégias, com 2 variações cada: ELIMINATE e UNION). Na Tabela 4.38 também podem ser notadas diferenças entre o MAJ e o KE, e entre o MAJ e o KU (fato que não ocorreu nos experimentos dos classificadores obtidos por *bagging*).

Na Tabela 4.38, o KE apresentou diferenças com os seguintes esquemas propostos: KNN+KE CF, KE+KNN CF, KU+KNN CF, KNN+KE UF e KNN+KU UF. Já o KU apresentou diferenças com os 10 métodos de fusão propostos, exceto KU+KNN CF e KNN+KU UF. Na Tabela 4.40 houve diferenças entre o KE e todos os esquemas de fusão do KNORA com o k -NN (exceto no KU+KNN CF), e entre o KU e os 10 esquemas de fusão.

Na Tabela 4.37 (ou Tabela 4.38) e na Tabela 4.39 (ou Tabela 4.40) também houve diferenças entre os esquemas propostos, como entre o KE CU e o KNN(após KU) CU, o KE CU e o KU+KNN CF, etc.

Como pode-se observar nos resultados, a informação adicional fornecida pelo k -NN construído no método KNORA, originalmente utilizado para definir os vizinhos da instância de teste, permitiu um incremento na taxa de reconhecimento da maioria das estratégias avaliadas. Avaliando as 8 bases de dados, os métodos KU CU e o KNN+KU CF superaram nas duas metodologias de criação de conjunto (subespaços aleatórios e *bagging*) os métodos utilizados para comparação (MAJ, KNORA-ELIMINATE e KNORA-UNION). Já os métodos KE+KNN CF e KNN+KU UF não apresentaram diferenças no teste de múltiplas comparações depois do teste de Friedman.

Para o conjunto de classificadores criados com subespaços aleatórios na base de dados *NIST SD19*, os melhores resultados foram atingidos quando a fusão incondicional era utilizada. Nesse caso, quando foi aplicado o teste para diferenças entre proporções, houve diferenças entre o KE e 5 dos 10 esquemas de fusão propostos; e entre o KU e 8 dos 10 esquemas de fusão. Quanto ao MAJ, as diferenças foram com os 10 esquemas propostos.

No conjunto de classificadores criados com *bagging* na base de dados *NIST SD19*, tanto MAJ quanto KU apresentaram diferenças com os 10 esquemas de fusão. Já KE apresentou diferenças com 9 dos 10 esquemas.

Conclui-se, então, que informações adicionais da vizinhança podem ter um papel importante no processo de classificação de um método de seleção dinâmica de subconjunto de classificadores. A próxima seção apresenta os resultados obtidos com outro novo método, o qual utiliza informações de acurácia geral e de acurácia de classe obtidas na vizinhança.

Tabela 4.37: Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA e de fusão do k -NN com o KNORA, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados *NIST SD19*

Método	MAJ	KE	KU	KE CU	KNN+KE CF	KU CU	KNN+KU CF	KNN(após KE) CU	KE+KNN CF	KNN(após KU) CU	KU+KNN CF	KNN+KE UF
KE	-12,40	-	-2,93	1,58	2,39	0,22	0,22	0,67	2,04	-0,44	-2,40	2,51
KU	-9,50	2,93	-	4,51	5,32	3,16	3,16	3,60	4,97	2,49	0,53	5,43
KE CU	-13,95	-1,58	-4,51	-	0,81	-1,36	-1,36	-0,91	0,46	-2,03	-3,98	0,93
KNN+KE CF	-14,75	-2,39	-5,32	-0,81	-	-2,17	-2,17	-1,72	-0,35	-2,83	-4,79	0,12
KU CU	-12,62	-0,22	-3,16	1,36	2,17	-	0,00	0,45	1,82	-0,67	-2,62	2,28
KNN+KU CF	-12,62	-0,22	-3,16	1,36	2,17	0,00	-	0,45	1,82	-0,67	-2,62	2,28
KNN(após KE) CU	-13,06	-0,67	-3,60	0,91	1,72	-0,45	-0,45	-	1,37	-1,12	-3,07	1,83
KE+KNN CF	-14,41	-2,04	-4,97	-0,46	0,35	-1,82	-1,82	-1,37	-	-2,49	-4,44	0,46
KNN(após KU) CU	-11,96	0,44	-2,49	2,03	2,83	0,67	0,67	1,12	2,49	-	-1,96	2,95
KU+KNN CF	-10,03	2,40	-0,53	3,98	4,79	2,62	2,62	3,07	4,44	1,96	-	4,90
KNN+KE UF	-14,86	-2,51	-5,43	-0,93	-0,12	-2,28	-2,28	-1,83	-0,46	-2,95	-4,90	-
KNN+KU UF	-10,03	2,40	-0,53	3,98	4,79	2,62	2,62	3,07	4,44	1,96	0,00	4,90

Tabela 4.38: Distinções no teste para diferenças entre proporções com os métodos MAJ, KNORA e de fusão do k -NN com o KNORA, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados *NIST SD19*

Método	MAJ	KE	KU	KE CU	KNN+KE CF	KU CU	KNN+KU CF	KNN(após KE) CU	KE+KNN CF	KNN(após KU) CU	KU+KNN CF	KNN+KE UF
KE	X		X		X				X		X	X
KU	X	X		X	X	X	X	X	X	X		X
KE CU	X		X							X	X	
KNN+KE CF	X	X	X			X	X			X	X	
KU CU	X		X		X						X	X
KNN+KU CF	X		X		X						X	X
KNN(após KE) CU	X		X								X	
KE+KNN CF	X	X	X							X	X	
KNN(após KU) CU	X		X	X	X				X			X
KU+KNN CF	X	X	X	X	X	X	X	X	X			X
KNN+KE UF	X	X	X			X	X			X	X	-
KNN+KU UF	X	X		X	X	X	X	X	X			X

Tabela 4.39: Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA e de fusão do k -NN com o KNORA, aplicado no conjunto inicial de classificadores formados por *bagging* da base de dados *NIST SD19*

Método	MAJ	KE	KU	KE CU	KNN+KE CF	KU CU	KNN+KU CF	KNN(após KE) CU	KE+KNN CF	KNN(após KU) CU	KU+KNN CF	KNN+KE UF
KE	-1,74	-	-0,79	15,42	15,64	15,42	15,53	10,99	8,97	9,67	1,87	15,10
KU	-0,95	0,79	-	16,19	16,41	16,19	16,30	11,77	9,75	10,45	2,66	15,87
KE CU	-17,12	-15,42	-16,19	-	0,22	0	0,11	-4,51	-6,54	-5,83	-13,59	-0,33
KNN+KE CF	-17,34	-15,64	-16,41	-0,22	-	-0,22	-0,11	-4,73	-6,76	-6,06	-13,80	-0,56
KU CU	-17,12	-15,42	-16,19	0	0,22	-	0,11	-4,51	-6,54	-5,83	-13,59	-0,33
KNN+KU CF	-17,23	-15,53	-16,30	-0,11	0,11	-0,11	-	-4,62	-6,65	-5,95	-13,69	-0,45
KNN(após KE) CU	-12,71	-10,99	-11,77	4,51	4,73	4,51	4,62	-	-2,04	-1,33	-9,13	4,17
KE+KNN CF	-10,69	-8,97	-9,75	6,54	6,76	6,54	6,65	2,04	-	0,71	-7,11	6,20
KNN(após KU) CU	-11,39	-9,67	-10,45	5,83	6,06	5,83	5,95	1,33	-0,71	-	-7,81	5,50
KU+KNN CF	-3,61	-1,87	-2,66	13,59	13,80	13,59	13,69	9,13	7,11	7,81	-	13,26
KNN+KE UF	-16,80	-15,10	-15,87	0,33	0,56	0,33	0,45	-4,17	-6,20	-5,50	-13,26	-
KNN+KU UF	-7,11	-5,38	-6,16	10,12	10,34	10,12	10,23	5,63	3,60	4,31	-3,51	9,79

Tabela 4.40: Distinções no teste para diferenças entre proporções com os métodos MAJ, KNORA e de fusão do k -NN com o KNORA, aplicado no conjunto inicial de classificadores formados por *bagging* da base de dados *NIST SD19*

Método	MAJ	KE	KU	KE CU	KNN+KE CF	KU CU	KNN+KU CF	KNN(após KE) CU	KE+KNN CF	KNN(após KU) CU	KU+KNN CF	KNN+KE UF
KE				X	X	X	X	X	X	X		X
KU				X	X	X	X	X	X	X	X	X
KE CU	X	X	X					X	X	X	X	
KNN+KE CF	X	X	X					X	X	X	X	
KU CU	X	X	X					X	X	X	X	
KNN+KU CF	X	X	X					X	X	X	X	
KNN(após KE) CU	X	X	X	X	X	X	X		X		X	X
KE+KNN CF	X	X	X	X	X	X	X	X			X	X
KNN(após KU) CU	X	X	X	X	X	X	X				X	X
KU+KNN CF	X	X	X	X	X	X	X	X	X	X		X
KNN+KE UF	X	X	X					X	X	X	X	
KNN+KU UF	X	X	X	X	X	X	X	X	X	X	X	X

4.6 Método de Seleção baseada em Acurácia Local Geral e de Classe

O método proposto de seleção dinâmica de subconjunto de classificadores baseado na acurácia local geral e na acurácia local de classe (DESCOLA) escolhe os subconjuntos de classificadores dinamicamente em duas fases. Na primeira fase (OLA) são preselecionados os classificadores que acertam os k vizinhos mais próximos. Na segunda fase (LCA), esses classificadores são agrupados em subconjuntos com ambiguidade zero e, para cada subconjunto, é calculada a média de acurácia local de classe. O subconjunto com maior média de acurácia local de classe é, então, escolhido para classificar a instância.

A Tabela 4.41 resume os melhores resultados obtidos considerando os conjuntos iniciais de classificadores gerados utilizando a estratégia de subespaços aleatórios. Essa tabela apresenta as médias⁴ de taxa de reconhecimento juntamente com os valores de desvio padrão obtidos na combinação de todos os classificadores do conjunto inicial usando voto majoritário (MAJ), no desempenho do oráculo de cada conjunto inicial, no método KNORA-ELIMINATE (KE) com seus valores de k , no método KNORA-UNION (KU) com seus valores de k , no método DESCOLA-ELIMINATE (DE), no método DESCOLA-UNION (DU) e no método DESCOLA-UNION-U (DU-U) com seus valores de k e v correspondentes. A variável k está relacionada aos vizinhos do KNORA original e à fase “OLA” no método DESCOLA. A variável v representa o número de vizinhos na fase “LCA” do método DESCOLA proposto. Além disso, a última coluna da Tabela 4.41 (DESLCA) representa o método com a aplicação direta da fase “LCA” no conjunto inicial de classificadores fracos, objetivando selecionar o subconjunto sem a fase de “OLA”. Na Tabela 4.42 têm-se os resultados considerando o conjunto inicial de classificadores gerados pela estratégia de *bagging*.

Todos os experimentos foram executados considerando valores de k e de v no intervalo de 1 a 30. As melhores médias de taxa de reconhecimento obtidas estão em negrito. Para cada uma das bases de dados, as melhores médias de taxa de reconhecimento foram fornecidas pelo DESCOLA ou pelo DESLCA. Quando utilizaram-se conjuntos iniciais de classificadores baseados no método de criação de subespaços aleatórios, a maioria dos melhores resultados foram atingidos pelo DE (Tabela 4.41), enquanto que para os conjuntos iniciais gerados utilizando *bagging*, a maior parte dos melhores resultados foram obtidos pelo DU-U (Tabela 4.42).

Na Tabela 4.41, o DE obteve a melhor taxa de reconhecimento em 4 das 8 bases de dados, e empatou em 2 bases de dados ou com o DU, ou com o DU-U ou com o DESLCA. O DU empatou com o melhor resultado em 1 das bases de dados. O DU-U obteve o melhor resultado em 1 base de dados, e empatou com a maior taxa de reconhecimento em 2 bases. Finalmente, o DESLCA atingiu a melhor taxa de reconhecimento em 1 base de dados e empatou com a maior taxa em outra.

Já na Tabela 4.42, o DE atingiu o melhor resultado em 2 das bases de dados, o DU em 1 base (e empatou com a maior taxa de reconhecimento em 1 também), o DU-U em 3 bases (e empatou com a maior taxa de reconhecimento em 1 base de dados), e o DESLCA obteve o melhor desempenho em 1 base. Analisando os valores obtidos na Tabela 4.41 e na Tabela 4.42, observa-se que os 4 esquemas propostos (DE, DU, DU-U e DESLCA) apresentaram taxa de reconhecimento superior à obtida pelo MAJ.

⁴Quando foi trabalhado com validação cruzada em uma determinada base de dados, é apresentada a melhor média de taxa de reconhecimento obtida nas execuções e o desvio padrão. Caso contrário, apenas é apresentada a melhor taxa de reconhecimento obtida.

Tabela 4.41: Melhor média de taxa de reconhecimento (TR) e valores de desvio padrão correspondentes (σ) utilizando subespaços aleatórios para a criação do conjunto inicial de classificadores

Base de dados	MAJ		Oráculo (teste)		KE(k)		KU(k)		DE ($k;v$)		DU ($k;v$)		DU-U ($k;v$)		DESLCA (v)	
	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ
letter	95,18	0,49	98,90	0,19	95,48 (1)	0,34	95,60 (3)	0,39	96,16 (2;2)	0,66	95,98 (1;10-30)	0,73	95,99 (1;4)	0,38	95,98 (1)	0,37
liver	65,82	3,29	97,69	2,65	65,49 (10)	4,80	68,12 (7)	3,54	68,15 (15;1)	5,37	67,85 (2;1)	10,87	67,85 (2;1)	10,87	68,69 (2)	7,70
pima	73,44	3,69	98,44	1,19	72,13 (2)	3,74	74,35 (6)	3,61	73,70 (4;1)	4,31	74,73 (4;2)	5,78	74,86 (24;15)	3,53	73,82 (29)	3,60
seg	88,33	-	99,52	-	88,33 (1-30)	-	88,33 (1-30)	-	92,62 (1;1-30)	-	92,62 (1;1-30)	-	92,62 (1;1-30)	-	92,62 (1)	-
wdbc	96,13	1,84	99,82	0,55	96,48 (1)	2,05	96,48 (1)	2,05	97,01 (2;2)	2,49	96,13 (1;1-30)	2,59	96,13 (1;1-30)	2,59	96,13 (1)	2,59
wine	94,41	4,54	100	0	97,75 (2,3)	2,91	96,67 (2,3)	4,68	98,30 (2;2)	3,79	97,75 (2;1)	3,92	97,75 (2;1)	3,92	97,19 (2)	3,96
NIST	96,28	-	99,95	-	97,52 (7,8)	-	97,25 (1)	-	97,67 (4;6)	-	97,46 (10;3-30)	-	97,40 (1;3-30)	-	97,46 (9,10)	-
LMD	69,22	1,39	90,22	1,50	68,89 (2)	1,95	70,00 (28,29)	1,00	70,00 (2;2)	1,86	69,78 (3;4,5)	1,90	70,00 (2;18)	1,86	69,67 (2)	1,33
													(3;18,19,29,30)			

Tabela 4.42: Melhor média de taxa de reconhecimento (TR) e desvios de padrão correspondentes (σ) utilizando *bagging* para a criação do conjunto inicial de classificadores

Base de dados	MAJ		Oráculo (teste)		KE(k)		KU(k)		DE ($k;\nu$)		DU ($k;\nu$)		DU-U ($k;\nu$)		DESLCA (ν)	
	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ	TR	σ
letter	93,79	0,51	98,24	0,30	94,34 (1)	0,46	94,17 (5)	0,42	95,02 (1;1)	0,44	95,12 (1;4)	0,38	95,13 (1;4)	0,40	95,11 (1)	0,37
liver	57,40	6,18	92,73	4,65	60,30 (26)	6,70	59,42 (3)	7,02	64,11 (30;1)	9,81	65,79 (30;5-7)	8,28	65,53 (24;24)	10,51	65,50 (30)	8,22
pima	70,95	4,81	92,32	2,16	72,26 (17)	3,42	71,99 (4)	5,23	73,83 (18;7)	3,35	74,60 (20;3)	3,65	74,87 (18;15)	1,93	74,47 (20)	3,63
seg	90,76	-	97,14	-	92,19 (1)	-	92,19 (1)	-	92,62 (1;5)	-	92,52 (2;1)	-	92,52 (2;1)	-	92,43 (1)	-
wdbc	96,13	2,00	98,59	1,39	96,48 (2-3)	2,04	96,13 (1,3)	1,62	97,18 (2;2)	1,90	96,31 (20;1)	1,29	96,31 (6;4)	2,26	96,13 (2)	2,01
wine	92,68	3,81	98,33	2,68	94,93 (6-8)	4,11	94,93 (6-8,23-30)	4,11	97,22 (15;4-5)	2,93	97,22 (15;2-30)	2,93	97,78 (6;7;9) (8;9-12) (11;7-12) (15;8-14)	3,88	97,22 (15)	2,93
NIST	95,75	-	97,85	-	95,95 (11)	-	95,86 (23-30)	-	96,88 (3-4;1)	-	96,90 (4;5-30)	-	96,88 (6;5)	-	96,93 (4)	-
LMD	67,89	3,67	86,22	2,14	68,22 (6)	1,35	67,56 (5)	1,95	70,78 (9;1)	3,37	71,33 (9;6-7)	2,60	71,33 (10;5)	2,08	71,11 (9)	2,80

O teste não paramétrico de Friedman [Demsar, 2006] [Hollander & Wolfe, 1999] foi utilizado para avaliar a significância dos resultados. Os valores das médias das taxas de reconhecimento obtidas pelos métodos MAJ, KE, KU, DE, DU, DU-U e DESLCA foram analisados nas 8 bases de dados. O valor obtido para p foi 0,0013 nos experimentos de subespaços aleatórios e aproximadamente zero nos experimentos de *bagging*. Isso confirmou um resultado significativo com intervalo de confiança de 95%.

Após o teste de Friedman, foi realizado um teste de múltiplas comparações onde a correção de Dunn-Sidak foi aplicada aos valores críticos. Nesse teste foram determinados quais pares de algoritmos eram significativamente diferentes. Assim, no caso da Tabela 4.41, houve diferença significativa entre os valores das médias das taxas de reconhecimento dos métodos MAJ e DE, como pode ser visualizado na Tabela 4.43. Já quanto à Tabela 4.42, notou-se que há diferenças significativas nos resultados obtidos entre os algoritmos MAJ e DE, entre os algoritmos MAJ e DU, entre os algoritmos MAJ e DU-U, entre os algoritmos MAJ e DESLCA, entre os algoritmos KU e DU e entre os algoritmos KU e DU-U (Tabela 4.44). Usando a correção de Bonferroni-Dunn, as conclusões foram as mesmas.

Tabela 4.43: Métodos que mostraram diferenças utilizando correção de Dunn-Sidak no teste de múltiplas comparações com os métodos MAJ, KNORA, DESCOLA e DESLCA, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios

Método	MAJ	KE	KU
DE	X		
DU			
DU-U			
DESLCA			

Tabela 4.44: Métodos que mostraram diferenças utilizando correção de Dunn-Sidak e de Bonferroni-Dunn no teste de múltiplas comparações com os métodos MAJ, KNORA, DESCOLA e DESLCA, aplicado no conjunto inicial de classificadores formados por *bagging*

Método	MAJ	KE	KU
DE	X		
DU	X		X
DU-U	X		X
DESLCA	X		

A Tabela 4.45 apresenta os valores de z (Equação 4.2) obtidos no teste para diferenças entre proporções ([Dietterich *et al.*, 1995] [Dietterich, 1997]) da base de dados *NIST SD19* no conjunto inicial de classificadores formados por subespaços aleatórios, enquanto a Tabela 4.47 apresenta os valores de z obtidos com os métodos aplicados no conjunto inicial de classificadores formados por *bagging*.

Os valores de módulo de z das Tabelas 4.45 e 4.47 que são maiores do que 1,96 indicam diferenças a um nível de confiança de 95%. Assim, as células marcadas com um X na Tabela 4.46 indicam entre quais métodos houve diferenças na Tabela 4.45, enquanto que as células marcadas com um X na Tabela 4.48 representam onde foram as diferenças da Tabela 4.47.

No caso da Tabela 4.47 (conjunto inicial criado com *bagging*), não houve diferenças nos métodos propostos entre si, apenas em relação ao MAJ, ao KE e ao KU. Os 4 esquemas apresentaram resultados distintos de MAJ, KE e KU. Na Tabela 4.45 isso ocorreu apenas em relação ao MAJ. Em relação ao KU, as diferenças foram entre 3 dos 4 esquemas tratados.

Tabela 4.45: Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA, DESCOLA e DESLCA, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados *NIST SD19*

Método	MAJ	KE	KU	DE	DU	DU-U
KE	-12,40	-	-2,93	1,70	-0,66	-1,32
KU	-9,50	2,93	-	4,63	2,27	1,61
DE	-14,07	-1,70	-4,63	-	-2,36	-3,02
DU	-11,75	0,66	-2,27	2,36	-	-0,66
DU-U	-11,10	1,32	-1,61	3,02	0,66	-
DESLCA	-11,75	0,66	-2,27	2,36	0,00	-0,66

Tabela 4.46: Distinções no teste para diferenças entre proporções com os métodos MAJ, KNORA, DESCOLA e DESLCA, aplicado no conjunto inicial de classificadores formados por subespaços aleatórios da base de dados *NIST SD19*

Método	MAJ	KE	KU	DE	DU	DU-U
KE	X		X			
KU	X	X		X	X	
DE	X		X	X	X	X
DU	X		X	X		
DU-U	X			X		
DESLCA	X		X	X		

Tabela 4.47: Valores de z obtidos no teste para diferenças entre proporções com os métodos MAJ, KNORA, DESCOLA e DESLCA, aplicado no conjunto inicial de classificadores formados por *bagging* da base de dados *NIST SD19*

Método	MAJ	KE	KU	DE	DU	DU-U
KE	-1,74	-	-0,79	8,67	8,87	8,67
KU	-0,95	0,79	-	9,45	9,65	9,45
DE	-10,40	-8,67	-9,45	-	0,20	0
DU	-10,59	-8,87	-9,65	-0,20	-	-0,20
DU-U	-10,40	-8,67	-9,45	0	0,20	-
DESLCA	-10,89	-9,17	-9,95	-0,50	-0,30	-0,50

Tabela 4.48: Distinções no teste para diferenças entre proporções com os métodos MAJ, KNORA, DESCOLA e DESLCA, aplicado no conjunto inicial de classificadores formados por *bagging* da base de dados *NIST SD19*

Método	MAJ	KE	KU	DE	DU	DU-U
KE				X	X	X
KU				X	X	X
DE	X	X	X			
DU	X	X	X			
DU-U	X	X	X			
DESLCA	X	X	X			

De acordo com os experimentos dessa seção, a melhor taxa de reconhecimento, para cada uma das bases, foi obtida por alguma das variações do DESCOLA, nomeadas DESCOLA-ELIMINATE, DESCOLA-UNION e DESCOLA-UNION-U, ou pelo DESLCA. Além disso, o DESCOLA também ultrapassou (ou permaneceu igual) os resultados obtidos pelo voto majoritário dos classificadores do conjunto inicial, com a vantagem de poder ter menos classificadores utilizados no processo de classificação.

Quanto aos resultados do DESLCA, pode-se afirmar que foram promissores, mas é importante notar que nesse método, todos os classificadores do conjunto inicial foram selecionados para todas as instâncias que foram classificadas. O desempenho do oráculo não foi atingido, mas o método DESCOLA e o método DESLCA provaram ser uma boa alternativa para a seleção dinâmica de subconjunto de classificadores.

4.7 Conclusão

O presente capítulo tratou dos resultados experimentais de 3 adaptações no método KNORA e de 3 novas abordagens em 8 bases de dados distintas, sendo que 2 eram de grande dimensão. Foram utilizados conjuntos de classificadores criados por 2 técnicas diferentes: subespaços aleatórios e *bagging*. Os métodos propostos foram comparados com a taxa de reconhecimento obtida pelo voto majoritário de todos os classificadores do conjunto inicial, pelo método KNORA-ELIMINATE e pelo método KNORA-UNION, sendo esses dois últimos propostos por [Ko *et al.*, 2008].

Os resultados obtidos nas 8 bases de dados foram analisados por meio dos testes de Friedman e de múltiplas comparações. Uma das bases de dados de grande dimensão (com as mesmas configurações de [Ko *et al.*, 2008]) foi selecionada para uma análise mais específica, onde o teste para diferenças entre proporções foi realizado.

Primeiramente foram usadas diferentes medidas de distância para encontrar os vizinhos no método KNORA (que utiliza originalmente a distância Euclidiana), como Canberra, Cosine e Pearson. As novas métricas de cálculo de vizinhança permitiram a seleção de diferentes classificadores. Quanto às taxas de reconhecimento, o teste de Friedman apontou diferenças com as 3 novas distâncias nos conjuntos formados por *bagging*, mas não foi possível listar entre quais métodos isso aconteceu.

No teste para diferenças entre proporções para a medida de Canberra, observou-se que o KNORA teve mudanças no desempenho (a distância Euclidiana foi melhor) no conjunto inicial formado por subespaços aleatórios da base de dados selecionada. Nas outras 2 medidas, não se pôde notar diferença na melhor taxa de reconhecimento do KNORA para essa base. As métricas avaliadas não permitiram incremento na melhor taxa de reconhecimento nem do KNORA-ELIMINATE e nem do KNORA-UNION.

Depois, foram feitos testes com a segunda abordagem preliminar, que tratava do uso de informações adicionais de agrupamento no método KNORA. Os classificadores selecionados tiveram seu voto ponderado pela frequência relativa da sua classe no agrupamento ao qual pertencia a instância a ser classificada. Nos conjuntos de classificadores criados com subespaços aleatórios, o KNORA-UNION foi melhor que sua variação com agrupamento e não houve diferenças significativas nas melhores taxas de reconhecimento das bases de dados com conjuntos de classificadores criados com *bagging*. Nos testes para diferenças entre proporções, também notou-se que o KNORA original era melhor. Da maneira como foi abordada, não pareceu ser

uma alternativa interessante para auxílio no processo de classificação de sistemas baseados em múltiplos classificadores com seleção dinâmica de subconjunto.

Uma terceira abordagem preliminar usou a proporção de classes reconhecidas por classificador como peso nos votos do KNORA. Os resultados também não indicaram diferenças entre o KNORA original e o KNORA com o uso desses pesos.

A seguir, foram realizados experimentos com as três novas abordagens. Na primeira, juntamente com o método KNORA, foram utilizadas as classes obtidas pelo k -NN embutido no método, de acordo com alguns critérios. Dos 10 esquemas de fusão apresentados (5 estratégias, com 2 variações cada: ELIMINATE e UNION), 7 foram indicados, por meio de testes de múltiplas comparações, com diferenças positivas em relação a pelo menos um dos métodos: uso do voto majoritário de todos os classificadores do conjunto inicial, KNORA-ELIMINATE ou KNORA-UNION. Desses, 2 esquemas foram apontados tanto nos experimentos dos conjuntos iniciais de classificadores criados por subespaços aleatórios como dos conjuntos iniciais de classificadores criados por *bagging* como melhores que os 3 métodos citados para comparação (MAJ, KNORA-ELIMINATE e KNORA-UNION). Esses dois esquemas analisavam os votos do k -NN com condições para utilizar ou não os votos do KNORA-UNION (individualmente, ou fundidos com os votos do k -NN).

No teste para diferenças entre proporções na base de dados previamente selecionada, houve diferenças entre o KNORA-ELIMINATE e 5 dos 10 esquemas de fusão propostos; e entre o KNORA-UNION e 8 dos 10 esquemas de fusão. Quanto ao uso do voto majoritário de todos os classificadores do conjunto inicial, as diferenças foram com os 10 esquemas propostos. No conjunto de classificadores criados com *bagging*, tanto MAJ quanto KNORA-UNION apresentaram diferenças com os 10 esquemas de fusão; enquanto o KNORA-ELIMINATE apresentou diferenças com 9 dos 10 esquemas.

A informação adicional fornecida pelo k -NN construído no método KNORA, originalmente utilizado para definir os vizinhos da instância de teste, permitiu um incremento na taxa de reconhecimento da maioria das estratégias avaliadas. Isso significa que informação adicional da vizinhança tem um peso importante no processo de classificação do método KNORA.

Depois, trabalhou-se com o método DESCOLA, onde a seleção dinâmica dos classificadores foi realizada em duas fases. Na primeira fase (fase “OLA”), foram preselecionados os classificadores que reconhecem os vizinhos (da base de dados de validação) de uma instância de teste. Na segunda fase (fase “LCA”), esses classificadores foram agrupados de acordo com o valor de ambiguidade. Então, a média do LCA (*Local Class Accuracy* - acurácia de classe local dos classificadores) foi calculada em cada grupo. O subconjunto com maior média de LCA foi escolhido para fornecer a classe final à instância de teste. Juntamente com o método DESCOLA, também foram realizados testes com o método DESLCA, que era constituído apenas pela fase “LCA”.

De acordo com os experimentos, para cada base de dados, a melhor taxa de reconhecimento de uma das variações do DESCOLA, nomeadas DESCOLA-ELIMINATE, DESCOLA-UNION e DESCOLA-UNION-U, foram melhores que as taxas de reconhecimento do KNORA original na maioria dos casos. Além disso, o DESCOLA (e o DESLCA) também ultrapassou (ou permaneceu igual) os resultados obtidos pelo voto majoritário dos classificadores do conjunto inicial, com a vantagem de poder ter menos classificadores utilizados no processo de classificação.

Quando utilizaram-se conjuntos iniciais de classificadores baseados no método de criação de subespaços aleatórios, a maioria dos melhores resultados foram atingidos pelo

DESCOLA-ELIMINATE, enquanto que para os conjuntos iniciais gerados utilizando *bagging*, a maior parte dos melhores resultados foram obtidos pelo DESCOLA-UNION-U.

Quanto aos resultados do DESLCA, pode-se afirmar que foram promissores, mas é interessante notar que nesse método, todos os classificadores do conjunto inicial foram utilizados para calcular o valor de LCA. Ou seja, precisou-se consultar todos classificadores disponíveis, para todas as instâncias de teste. Quando há muitos classificadores no conjunto inicial, e a primeira fase do método DESCOLA consegue preselecionar poucos classificadores, o DESCOLA pode ter um processamento mais rápido que o DESLCA.

O desempenho do oráculo não foi atingido mas, de todos os experimentos realizados, o método DESCOLA (e DESLCA) e o método que utilizava o k -NN embutido do KNORA na avaliação provaram ser uma boa alternativa para a seleção dinâmica de subconjuntos de classificadores.

O próximo capítulo apresenta a conclusão do trabalho, salientando os pontos mais importantes abordados, vantagens e desvantagens, bem como direções futuras.

Capítulo 5

Conclusão

A seleção dinâmica de subconjuntos de classificadores visa escolher os classificadores que parecem mais adequados a cada instância de teste. O limite superior de reconhecimento que um método de SDSC pode obter é denominado oráculo. Todos os métodos de SDSC tentam aproximar-se desse valor. Com esse objetivo, o presente trabalho estudou o uso de informações da vizinhança das instâncias de teste para auxiliar na escolha dos classificadores. Essas informações englobaram principalmente o reconhecimento dos vizinhos pelos classificadores, ou seja, a acurácia local.

Primeiramente, foi realizado um estudo sobre o estado da arte de SDSC, onde foram apresentados conceitos relacionados a sistemas baseados em múltiplos classificadores e analisados alguns trabalhos recentes. Um dos trabalhos que recebeu uma atenção especial foi o KNORA, devido à sua metodologia de uso de informações da vizinhança. O método KNORA analisa a capacidade que os classificadores têm de reconhecer as instâncias de uma base de validação, instâncias essas consideradas parecidas com a instância a ser classificada. Os valores de distância Euclidiana calculados entre os atributos da instância de teste e os atributos de cada uma das instâncias de validação são que determinam quais são as instâncias “vizinhas”. Um dos esquemas do KNORA, denominado KNORA-ELIMINATE, seleciona apenas os classificadores que reconhecem o maior número de vizinhos entre os disponíveis. Já um outro esquema, denominado KNORA-UNION, seleciona todos os classificadores que possuem ao menos um reconhecimento na vizinhança. Nesse caso, o voto do classificador selecionado será ponderado pela quantidade de vizinhos que o mesmo reconhece.

Depois, foram propostas e implementadas 3 adaptações no método KNORA para avaliar o impacto que essas modificações causariam. Todos os experimentos foram realizados em 8 base de dados, sendo que 2 eram de grande dimensão. Foram utilizados conjuntos de classificadores criados por 2 técnicas distintas: subespaços aleatórios e *bagging*. O *k*-NN foi utilizado como algoritmo fraco em todos os conjuntos iniciais de classificadores.

Assim, como primeira adaptação, foram usadas diferentes medidas de distância para encontrar os vizinhos, como Canberra, Cosine e Pearson. Em um teste no conjunto inicial formado por subespaços aleatórios em uma determinada base de dados, observou-se que o KNORA teve mudanças no desempenho com a medida de Canberra (a distância Euclidiana, que é do KNORA original, foi melhor). Nas outras 2 medidas, não se pôde notar diferença na melhor taxa de reconhecimento do KNORA para essa base. Em relação ao KNORA original, não houve grandes diferenças nas taxas de reconhecimento, porém elas se modificaram, indicando que alguns classificadores diferentes foram selecionados para compor os subconjuntos. As métricas

avaliadas não permitiram incremento significativo na melhor taxa de reconhecimento nem do KNORA-ELIMINATE e nem do KNORA-UNION.

Na sequência, propôs-se o uso da frequência relativa das classes no agrupamento (ao qual à instância de teste pertence) como peso nos votos dos classificadores selecionados pelo KNORA. Não houve melhorias significativas nos resultados. Da maneira como foi abordada, não parece ser uma alternativa interessante para auxílio no processo de classificação de sistemas baseados em múltiplos classificadores com seleção dinâmica de subconjunto. Já a terceira adaptação proposta usou a proporção de classes por classificador como pesos nos votos. Os resultados também não foram promissores.

A seguir, três novas abordagens foram tratadas. Na primeira nova abordagem, a informação adicional fornecida pelo k -NN construído no método KNORA, originalmente utilizado para definir os vizinhos da instância de teste, permitiu um incremento na taxa de reconhecimento da maioria das estratégias avaliadas. Cinco estratégias diferentes foram propostas, combinando ou não os votos do k -NN com o KNORA, de acordo com algumas condições. A primeira estratégia utilizava apenas os votos do KNORA se não fosse atingida uma determinada proporção de votos para uma mesma classe no k -NN. A segunda estratégia utilizava a combinação dos votos do KNORA com o k -NN se não fosse atingida essa proporção de votos para uma mesma classe no k -NN. A terceira estratégia utilizava apenas os votos do k -NN se não fosse atingida uma determinada proporção de votos para a mesma classe no KNORA enquanto que a quarta estratégia unia os votos do KNORA com o k -NN se não fosse atingida essa condição. Já a última estratégia unia os votos do KNORA com os votos do k -NN incondicionalmente.

Dos 10 esquemas de fusão do método k -NN com o método KNORA apresentados (5 estratégias, com 2 variações cada: ELIMINATE e UNION), 7 foram melhores do que pelo menos um dos métodos: uso do voto majoritário de todos os classificadores do conjunto inicial, KNORA-ELIMINATE ou KNORA-UNION. Inclusive, dois desses esquemas superaram os 3 métodos citados. Esses dois esquemas analisavam os votos do k -NN com condições para utilizar ou não os votos do KNORA-UNION (individualmente, ou unidos com os votos do k -NN).

A principal vantagem de utilizar uma estratégia de fusão, além da possibilidade de incrementos na taxa de reconhecimento, é que foi realizada apenas uma pequena mudança na metodologia no KNORA, aproveitando o procedimento de cálculo da vizinhança já utilizado. Assim, as classes reais dos vizinhos puderam auxiliar na classificação. Como principal desvantagem, tem-se o fato de haver mais um parâmetro de entrada para ser configurado nas estratégias condicionais.

Já a segunda nova abordagem (método DESCOLA) tratou da seleção dinâmica dos classificadores que compõem o subconjunto sendo realizada de forma complementar em duas fases. Na primeira fase são preselecionados os classificadores de acordo com seus valores de acurácia local geral (OLA). Na segunda fase esses classificadores são agrupados utilizando o valor de ambiguidade, para que então seja calculada a média de acurácia de classe local (LCA) de cada grupo. O subconjunto com maior média de LCA é escolhido para fornecer a classe final à instância de teste. Uma variação do método DESCOLA também foi proposta, onde somente a segunda fase foi aplicada. Essa variação nomeou-se DESLCA (terceira nova abordagem).

O DESCOLA (e o DESLCA) ultrapassou (ou permaneceu igual) as taxas de reconhecimento obtidas pelo voto majoritário dos classificadores do conjunto inicial, com a vantagem de poder ter menos classificadores utilizados, no caso do DESCOLA, no processo de classificação. Analisando as bases de dados em conjunto, também obteve melhor desempenho de reconhecimento que o KNORA original. Testes estatísticos comprovaram a significância des-

ses incrementos. Os resultados do DESLCA foram promissores, porém, todos os classificadores do conjunto inicial foram utilizados para calcular o valor de LCA.

A principal vantagem do método DESCOLA é que a acurácia de classes também pôde ser considerada no processo, permitindo a seleção de um subconjunto de classificadores mais adaptados para cada uma das instâncias desconhecidas. O fato de se ter uma primeira fase para preselecionar os classificadores permitiu que o processo não precisasse ser repetido com todos os classificadores. Já o DESLCA teve isso como desvantagem, pois todos os classificadores foram utilizados. O DESLCA é computacionalmente mais caro que o voto majoritário de todos os classificadores do conjunto inicial porque, além de utilizar os classificadores, também faz as análises do LCA. Porém, a sua taxa de reconhecimento é normalmente melhor.

O desempenho do oráculo não foi atingido mas, de todos os experimentos realizados, o método DESCOLA e o método que utilizava o k -NN embutido do KNORA na avaliação provaram ser uma boa alternativa para a seleção dinâmica de subconjuntos de classificadores. Assim, os resultados reforçaram o fato de que informações da vizinhança (especialmente a acurácia local) podem ser de grande utilidade em métodos de seleção dinâmica de subconjunto de classificadores.

Como trabalho futuro, planeja-se avaliar o método *boosting* como uma alternativa para criar o conjunto inicial de classificadores, e também o uso de outros classificadores ao invés do k -NN. Além disso, também almeja-se testar o uso das duas novas abordagens propostas (fusão do k -NN como o KNORA e método DESCOLA) em conjunto.

Referências Bibliográficas

- [ELE, ND] (ND). E. E. project: ROARS, Phoneme database from ELENA project. URL <<http://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/>>.
- [Asuncion & Newman, 2007] Asuncion, A.; Newman, D. J. (2007). UCI machine learning repository. URL <<http://www.ics.uci.edu/~mlearn/MLRepository.html>>.
- [Benediktsson *et al.*, 1997] Benediktsson, J.; Sveinsson, J.; Ersoy, O.; Swain, P. (1997). Parallel consensual neural networks. *IEEE Transaction on Neural Networks*, v.8, n.1, p.54–64.
- [Bittencourt, 2005] Bittencourt, V. G. (2005). Aplicação de técnicas de aprendizado de máquina no reconhecimento de classes estruturais de proteínas. Dissertação (Mestrado), Programa de Pós-Graduação em Engenharia Elétrica, Centro de Tecnologia, Universidade Federal do Rio Grande do Norte. 101p.
- [Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine Learning*, v.24, n.2, p.123–140.
- [Breiman *et al.*, 1984] Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.
- [Canuto, 2001] Canuto, A. M. P. (2001). *Combining Neural Networks and Fuzzy Logic for Applications in Character Recognition*. Phd thesis, Department of Electronics, University of Kent, Canterbury, UK. 196p.
- [Cho, 1999] Cho, S. (1999). Pattern recognition with neural networks combined by genetic algorithm. *Fuzzy Sets and Systems*, v.103, n.2, p.339–347.
- [Costa *et al.*, 2012] Costa, Y. M. G.; Oliveira, L. S.; Koerich, A. L.; Gouyon, F. (2012). Comparing textural features for music genre classification. *WCCI 2012 IEEE World Congress on Computational Intelligence*, p. 1867–1872, Brisbane, Australia.
- [Cover & Hart, 1967] Cover, T. M.; Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, v.13, n.1, p.21–27.
- [Davis *et al.*, 2004] Davis, J.; Costa, V. S.; Ong, I. M.; Dutra, I. (2004). Using Bayesian classifiers to combine rules. *3rd Workshop on Multi-Relational Data Mining*, Seattle, USA.
- [Demsar, 2006] Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, v.7, p.1–30.

- [Didaci & Giacinto, 2004] Didaci, A.; Giacinto, G. (2004). Dynamic classifier selection by adaptive k-nearest-neighbourhood rule. *5th Int. Workshop on Multiple Classifier Systems (MCS 2004)*, p. 174 – 183. Springer-Verlag.
- [Didaci *et al.*, 2005] Didaci, L.; Giacinto, G.; Roli, F.; Marcialis, G. L. (2005). A study on the performances of dynamic classifier selection based on local accuracy estimation. *Pattern Recognition*, v.38, n.11, p.2188–2191.
- [Dietterich, 1997] Dietterich, T. G. (1997). Approximate statistical tests for comparing supervised classification learning algorithms.
- [Dietterich, 2000] Dietterich, T. G. (2000). Ensemble methods in machine learning. *Multiple Classifier Systems*, v. 1857, p. 1–15. Springer-Verlag.
- [Dietterich *et al.*, 1995] Dietterich, T. G.; Hild, H.; Bakiri, G. (1995). A comparison of ID3 and backpropagation for English text-to-speech mapping. *Machine Learning*, v.18, p.51–80.
- [Efron & Tibshirani, 1993] Efron, B.; Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Monographs on Statistics and Applied Probability; 57. Chapman and Hall, New York, NY.
- [Fauvel *et al.*, 2006] Fauvel, M.; Chanussot, J.; Benediktsson, J. A. (2006). Decision fusion for the classification of urban remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, v.44, n.10-1, p.2828–2838.
- [Frank & Asuncion, 2010] Frank, A.; Asuncion, A. (2010). UCI machine learning repository. URL <<http://archive.ics.uci.edu/ml>>.
- [Gader *et al.*, 1996] Gader, P. D.; Mohamed, M. A.; Keller, J. M. (1996). Fusion of handwritten word classifiers. *Pattern Recognition Letters*, v.17, n.6, p.577–584.
- [Giacinto & Roli, 1997a] Giacinto, G.; Roli, F. (1997a). Adaptive selection of image classifiers. *Image Analysis and Processing, 9th International Conference, ICIAP '97*, v. 1310 de *Lecture Notes in Computer Science*, p. 38–45, Florence, Italy. Springer.
- [Giacinto & Roli, 1997b] Giacinto, G.; Roli, F. (1997b). Ensembles of neural networks for soft classification of remote sensing images. *Proceedings of the European Symposium on Intelligent Techniques*, p. 166–170, Bari, Italy.
- [Giacinto & Roli, 1999] Giacinto, G.; Roli, F. (1999). Methods for dynamic classifier selection. *10th International Conference on Image Analysis and Processing (ICIAP 1999)*, p. 659–664, Venice, Italy. IEEE Computer Society.
- [Giacinto & Roli, 2001] Giacinto, G.; Roli, F. (2001). Dynamic classifier selection based on multiple classifier behaviour. *Pattern Recognition*, v.34, n.9, p.1879–1881.
- [Hall *et al.*, 2009] Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explorations*, v.11, n.1, p.10–18.
- [Hall, 1999] Hall, M. A. (1999). *Correlation-based Feature Subset Selection for Machine Learning*. Tese (Doutorado), Department of Computer Science, University of Waikato, Hamilton, New Zealand.

- [Hansen & Salamon, 1990] Hansen, L.; Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.12, p.993–1001.
- [Hashem, 1997] Hashem, S. (1997). Optimal linear combinations of neural networks. *Neural Networks*, v.10, n.4, p.599–614.
- [Ho, 1998] Ho, T. K. (1998). The random space method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.20, n.8, p.832–844.
- [Hollander & Wolfe, 1999] Hollander, M.; Wolfe, D. A. (1999). *Nonparametric Statistical Methods*. Wiley-Interscience, 2 edição.
- [Ivakhnenko, 1970] Ivakhnenko, A. G. (1970). Heuristic self-organization in problems of engineering cybernetics. *Automatica*, v.6, n.2, p.207–219.
- [Kittler & Alkoot, 2003] Kittler, J.; Alkoot, F. (2003). Sum versus vote fusion in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.25, p.110–115.
- [Kittler *et al.*, 1998] Kittler, J.; Hatef, M.; Duin, R. P. W.; Matas, J. (1998). On combining classifiers. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, v.20, n.3, p.226–239.
- [Ko, 2007] Ko, A. H. R. (2007). *Static and Dynamic Selection of Ensemble of Classifiers*. Phd thesis, École de Technologie Supérieure - Université du Québec, MONTREAL, Canada. 246p.
- [Ko *et al.*, 2008] Ko, A. H. R.; Sabourin, R.; Britto Jr, A. S. (2008). From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, v.41, n.5, p.1735–1748.
- [Kuncheva, 2002a] Kuncheva, L. (2002a). Switching between selection and fusion in combining classifiers: an experiment. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, v.32, n.2, p.146–156.
- [Kuncheva, 2002b] Kuncheva, L. I. (2002b). A theoretical study on six classifier fusion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.24, n.2, p.281–286.
- [Kuncheva, 2004a] Kuncheva, L. I. (2004a). *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience.
- [Kuncheva, 2004b] Kuncheva, L. I. (2004b). Ludmila Kuncheva Collection. URL <<http://www.bangor.ac.uk/~mas00a/activities/patrecl.html>>.
- [Kuncheva & Rodrigues, 2007] Kuncheva, L. I.; Rodrigues, J. J. (2007). Classifier ensembles with a random linear oracle. *IEEE Transactions on Knowledge and Data Engineering*, p. 1–11.
- [Kuncheva *et al.*, 2002] Kuncheva, L. I.; Skurichina, M.; Duin, R. P. W. (2002). An experimental study on diversity for bagging and boosting with linear classifiers. *Information Fusion*, v.3, n.4, p.245–258.

- [Kuncheva & Whitaker, 2003] Kuncheva, L. I.; Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, v.51, n.2, p.181–207.
- [Lipnickas & Korbicz, 2004] Lipnickas, A.; Korbicz, J. (2004). Adaptive selection of neural networks for a committee decision,. *International Scientific Journal of Computing*, v.3, n.2, p.23–30.
- [Lu, 1996] Lu, Y. (1996). Knowledge integration in a multiple classifier system. *Applied Intelligence*, v.6, n.2, p.75–86.
- [MacQueen, 1967] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. Cam, L. M. L.; Neyman, J., editores, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, v. 1, p. 281–297. University of California Press.
- [Maia, 2008] Maia, F. R. Z. (2008). Introdução aos sistemas multi-classificadores (SMC). Trabalho de Conclusão de Curso, Departamento de Informática, Universidade Estadual de Ponta Grossa.
- [Mitchell, 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York.
- [Oliveira, 2008] Oliveira, D. F. (2008). Dilema da diversidade-accurácia: Um estudo empírico no contexto de multiclassificadores. Dissertação (Mestrado), Universidade Federal do Rio Grande do Norte. Programa de Pós-Graduação em Sistemas e Computação.
- [Paradedda, 2007] Paradedda, R. B. (2007). Utilizando pesos estáticos e dinâmicos em sistemas multi-classificadores com diferentes níveis de diversidade. Exame de qualificação de dissertação (mestrado), Universidade Federal do Rio Grande do Norte.
- [Ranawana & Palade, 2006] Ranawana, R.; Palade, V. (2006). Multi-classifier systems: Review and a roadmap for developers. *International Journal of Hybrid Intelligent Systems*, v.3, n.1, p.35–61.
- [Santana *et al.*, 2006] Santana, A.; Soares, R. G. F.; Canuto, A. M. P.; Souto, M. C. P. (2006). A dynamic classifier selection method to build ensembles using accuracy and diversity. *Proceedings of the Ninth Brazilian Symposium on Neural Networks (SBRN'06)*, p. 36–41.
- [Santos, 2008] Santos, E. M. D. (2008). *Static and dynamic overproduction and selection of classifier ensembles with genetic algorithms*. Phd thesis, École de Technologie Supérieure. Université du Québec, Montreal.
- [Santos *et al.*, 2008] Santos, E. M. D.; Sabourin, R.; Maupin, P. (2008). A dynamic overproduce-and-choose strategy for the selection of classifier ensembles. *Pattern Recognition*, v.41, n.10, p.2993–3009.
- [Schapire *et al.*, 1998] Schapire, R.; Freund, Y.; Bartlett, P.; Lee, W. (1998). Boosting the margin: a new explanation for the effectiveness of voting methods. *Ann. Stat.*, v.26, n.5, p.1651–1686.

- [Seewald, 2002] Seewald, A. K. (2002). How to make stacking better and faster while also taking care of an unknown weakness. *Proceedings of the Nineteenth International Conference on Machine Learning, ICML '02*, p. 554–561, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Seewald, 2003] Seewald, A. K. (2003). Towards a theoretical framework for ensemble classification. *Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI '03*, p. 1443–1444, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Silla Jr. *et al.*, 2008] Silla Jr., C. N.; Koerich, A. L.; Kaestner, C. A. A. (2008). The latin music database. *ISMIR 2008, 9th International Conference on Music Information Retrieval*, p. 451–456, Drexel University, Philadelphia, PA, USA.
- [Skurichina & Duin, 2001] Skurichina, M.; Duin, R. P. W. (2001). Bagging and the random subspace method for redundant feature spaces. Kittler, J.; Roli, F., editores, *Multiple Classifier Systems*, v. 2096 de *Lecture Notes in Computer Science*, p. 1–10. Springer.
- [Stefano *et al.*, 2002] Stefano, C. D.; Cioppa, A. D.; Marcelli, A. (2002). An adaptive weighted majority vote rule for combining multiple classifiers. *ICPR (2)*, p. 192–195.
- [Ting & Witten, 1999] Ting, K. M.; Witten, I. H. (1999). Issues in Stacked Generalization. *Journal of Artificial Intelligence Research*, v.10, p.271–289.
- [Tremblay *et al.*, 2004] Tremblay, G.; Sabourin, R.; Maupin, P. (2004). Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm. *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04)*, v. 1 de *ICPR '04*, p. 208, Washington, DC, USA. IEEE Computer Society.
- [Vriesmann *et al.*, 2010] Vriesmann, L. M.; Britto Jr., A. S.; Oliveira, L. S.; Sabourin, R.; Ko, A. (2010). Using additional neighborhood information in a dynamic ensemble selection method: improving the KNORA approach. *17th International Conference on Systems, Signals and Image Processing (IWSSIP 2010)*, p. 420–423, Rio de Janeiro, Brazil.
- [Vriesmann *et al.*, 2012] Vriesmann, L. M.; Britto Jr., A. S.; Oliveira, L. S.; Sabourin, R.; Ko, A. (2012). Improving a dynamic ensemble selection method based on oracle information. *International Journal of Innovative Computing and Applications (IJICA)*. Special Issue on: Intelligent Image and Signal Processing. In Press.
- [Witten & Frank, 2005] Witten, I. H.; Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edição.
- [Woloszynski & Kurzynski, 2010] Woloszynski, T.; Kurzynski, M. (2010). A measure of competence based on randomized reference classifier for dynamic ensemble selection. *Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR '10*, p. 4194–4197, Washington, DC, USA. IEEE Computer Society.
- [Woloszynski & Kurzynski, 2011] Woloszynski, T.; Kurzynski, M. (2011). A probabilistic model of classifier competence for dynamic ensemble selection. *Pattern Recognition*, v.44, n.10-11, p.2656–2668.

- [Wolpert, 1992] Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, v.5, p.241–259.
- [Woods *et al.*, 1997] Woods, K.; Kegelmeyer, Jr., W. P.; Bowyer, K. (1997). Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.19, n.4, p.405–410.
- [Xiang *et al.*, 2005] Xiang, C.; Ding, S.; Lee, T. H. (2005). Geometrical interpretation and architecture selection of MLP. *IEEE Transactions on Neural Networks*, v.16, n.1, p.84–96.
- [Xiao & He, 2008] Xiao, J.; He, C. (2008). Adaptive selection of classifier ensemble based on GMDH. *Proceedings of the 2008 International Seminar on Future Information Technology and Management Engineering*, FITME '08, p. 61–64, Washington, DC, USA. IEEE Computer Society.
- [Xiao & He, 2009] Xiao, J.; He, C. (2009). Dynamic classifier ensemble selection based on GMDH. Yu, L.; Lai, K. K.; Mishra, S. K., editores, *CSO (1)*, p. 731–734. IEEE Computer Society.
- [Xu *et al.*, 1992] Xu, L.; Krzyzak, A.; Suen, C. (1992). Methods for combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, v.24, n.3, p.418–435.
- [Zenobi & Cunningham, 2001] Zenobi, G.; Cunningham, P. (2001). Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error. *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, p. 576–587, London, UK. Springer-Verlag.