EUNELSON JOSÉ DA SILVA JÚNIOR

Método de Classificação em Cascata de Dois Níveis: uma alternativa para a redução do custo de sistemas baseados em múltiplos classificadores

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

Curitiba
2015

EUNELSON JOSÉ DA SILVA JÚNIOR

Método de Classificação em Cascata de Dois Níveis: uma alternativa para a redução do custo de sistemas baseados em múltiplos classificadores

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

Área de Concentração:
Reconhecimento de Padrões

Orientador:
Prof. Dr. Alceu de Souza Britto Jr.
Co-orientadores:
Prof. Dr. Luiz Eduardo Soares de Oliveira
Prof. Dr. Robert Sabourin

Curitiba
2015

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

S586m
2015

Silva Júnior, Eunelson José da
Método de classificação em cascata de dois níveis: uma alternativa para a redução do custo de sistemas baseados em múltiplos classificadores / Eunelson José da Silva Júnior; orientador Alceu de Souza Britto Jr.; co-orientadores, Luiz Eduardo Soares de Oliveira, Robert Sabourin. -- 2015
87 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2015
Bibliografia: f.64-68

1. Informática. 2. Algoritmos computacionais. 3. Classificação. 4. Base de dados. 5. Análise de sistemas. 6. Teoria dos conjuntos. I. Alceu de Souza Britto Jr., 1966-. II. Oliveira, Luiz Eduardo Soares de. III. Sabourin, Robert. IV. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. V. Título.

CDD 20. ed. – 004.068

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

PUCPR
GRUPO MARISTA

ATA DE DEFESA DE DISSERTAÇÃO DE MESTRADO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DEFESA DE DISSERTAÇÃO DE MESTRADO Nº 12/2015

Aos 14 dias do mês de Outubro de 2015 realizou-se a sessão pública de Defesa da Dissertação " **Método de Classificação em Cascata de Dois Níveis: Uma Alternativa para a Redução do Custo de Sistemas Baseados em Múltiplos Classificadores"** apresentado pelo aluno **Eunelson José da Silva Júnior**, como requisito parcial para a obtenção do título de Mestre em Informática, perante uma Banca Examinadora composta pelos seguintes membros:

Prof. Dr. Alceu de Souza Britto Jr
PUCPR (Orientador) _____ APROVADO
(assinatura) (Aprov/Reprov)

Prof. Dr. Luiz Eduardo Soares de Oliveira
UFPR (Co-orientador) _____ APROV
(assinatura) (Aprov/Reprov)

Prof. Dr. Robert Sabourin
ETS (Co-orientador) _____ APROVADO
(assinatura) (Aprov/Reprov)

Prof. Dr. Júlio Cesar Nievola
PUCPR _____ APROVADO
(assinatura) (Aprov/Reprov)

Prof. Dr. George Darmiton da Cunha Cavalcanti
UFPE _____ Aprovado
(assinatura) (Aprov/Reprov)

Conforme as normas regimentais do PPGIa e da PUCPR, o trabalho apresentado foi considerado APROVADO (*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.ª Dr.ª Andreia Malucelli.
Coordenadora do Programa de Pós-Graduação em Informática.

Programa de Pós-Graduação em Informática
PPGIa
PUCPR

Rua Imaculada Conceição, 1155 Prado Velho CEP 80215-901 Curitiba Paraná Brasil
Tel.: (41) 3271 1669 Fax.: (41) 3271 2321 www.ppgia.pucpr.br

Dedico à minha esposa Daniele,
aos meus pais, Eunelson e Aurora,
e a todos os meus familiares.

Agradecimentos

À Deus, meu refúgio e fortaleza, pelo dom da vida, pelo amparo e por diariamente me dar forças para continuar.

Ao meu orientador, professor Alceu de Souza Britto Jr. (PUCPR), pelo constante incentivo, pela paciência e companheirismo, pela dedicação e auxílio em todas etapas do mestrado, e por acreditar no meu potencial para o trabalho.

Aos meus co-orientadores, professor Luiz Eduardo S. Oliveira (UFPR) e professor Robert Sabourin (ÉTS/Canadá), pelas importantes contribuições ao projeto, pelas análises dos resultados e pelos diversos direcionamentos nos momentos de incertezas.

Aos professores convidados para as bancas de qualificação e de defesa, Alessandro L. Koerich (ÉTS/Canadá), George D. C. Cavalcanti (UFPE) e Júlio C. Nievola (PUCPR), pelas revisões, correções e contribuições para a melhoria do trabalho.

Aos professores, funcionários e colegas da PUCPR e UFPR, pelo apoio e pelo conhecimento compartilhado, em especial aos doutorandos Paulo R. L. Almeida, amigo de longa data, pela parceria na realização dos trabalhos e publicações, e André L. Brun, pela contribuição com os cálculos das métricas de complexidade.

À minha esposa, minha eterna namorada, Daniele Ruppel da Silva, pelo companheirismo, cumplicidade e dedicação, pelo suporte emocional e por seu amor incondicional.

Aos meus pais, minha eterna inspiração, Eunelson e Aurora, e minhas irmãs, Franciele e Pollyana, pelo amor e apoio sempre presentes, e por tantas vezes suportar minha ausência devido aos estudos.

Aos Institutos Lactec e aos colegas pesquisadores, pelas liberações para cumprir as atividades do mestrado e pelo incentivo à qualificação profissional.

À Pontifícia Universidade Católica do Paraná e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela oportunidade e pelo apoio financeiro.

A todos aqueles que acreditaram e torceram por esta conquista. Muito obrigado!

Sumário

Agradecimentos iv

Sumário v

Lista de Figuras viii

Lista de Tabelas x

Lista de Abreviações e Siglas xi

Resumo xiii

Abstract xiv

Capítulo 1

Introdução 1

1.1 Definição do problema 2

1.2 Objetivos 3

1.3 Desafios 4

1.4 Hipóteses 4

1.5 Estrutura do documento 4

Capítulo 2

Fundamentação Teórica 5

2.1 Classificadores monolíticos 7

2.1.1 K-Vizinhos Mais Próximos 8

2.1.2 Perceptron de Múltiplas Camadas 9

2.1.3 Árvore de Decisão 10

2.1.4 Naive Bayes 10

2.1.5 Máquinas de Vetores de Suporte 10
2.2 Sistemas de múltiplos classificadores 11
2.2.1 Geração do conjunto de classificadores 15
2.2.2 Seleção de classificadores 16
2.2.2.1 DCS-LA 16
2.2.2.2 KNORA 19
2.2.3 Fusão de classificadores 23
2.3 Rejeição 23
2.4 Classificação em cascata 24
2.5 Métricas de complexidade 25
2.5.1 Estimativa de complexidade dos dados 25
2.5.2 Estimativa de custo de classificação 27
2.6 Trabalhos relacionados 27
2.7 Considerações 32

Capítulo 3

Método Proposto 33
3.1 Primeiro nível da cascata 35
3.1.1 Limiar de rejeição 36
3.1.2 Estimativa de confiança na classificação 38
3.2 Segundo nível da cascata 39
3.2.1 Geração de conjuntos de classificadores 40
3.2.2 Sistemas de múltiplos classificadores 40
3.3 Métricas de avaliação de desempenho 41

Capítulo 4

Experimentos e Resultados 42
4.1 Bases de dados utilizadas 43
4.2 Primeiro nível da cascata - Classificador monolítico 47
4.3 Segundo nível da cascata - Sistema de múltiplos classificadores 53
4.4 Classificação em cascata 57

Capítulo 5

Conclusão 62

5.1 Trabalhos futuros 63

Referências Bibliográficas 64

Apêndice A

Resultados gerais dos experimentos 69

A.1 Análise da rejeição no classificador monolítico 69

A.2 Combinação de conjunto de classificadores 79

A.3 Seleção dinâmica de classificadores 82

A.4 Classificação em cascata 84

Lista de Figuras

Figura 2.1 Etapas de um sistema clássico de reconhecimento de padrões 6
Figura 2.2 A razão estatística para combinação de classificadores 12
Figura 2.3 A razão computacional para combinação de classificadores 13
Figura 2.4 A razão representacional para combinação de classificadores 14
Figura 2.5 Possíveis fases de um sistema de múltiplos classificadores 14
Figura 2.6 Exemplo do método OLA com três classificadores (c_1, c_2 e c_3) rotulando sete vizinhos da amostra de teste. O classificador c_3 é selecionado por apresentar maior taxa de acertos geral. 17
Figura 2.7 Exemplo do método LCA com três classificadores rotulando sete vizinhos da amostra de teste. As classificações onde os rótulos atribuidos aos vizinhos são diferente da amostra de teste, são ignoradas. O classificador c_2 é selecionado por apresentar maior taxa de acertos local. . . 19
Figura 2.8 Exemplo do método KNORA-E com quatro classificadores e sete vizinhos. A execução teve quatro iterações para selecionar os classificadores c_1 e c_4. 21
Figura 2.9 Exemplo do método KNORA-U com quatro classificadores e sete vizinhos. Todos os classificadores foram selecionados considerando a quantidade de votos (acertos) de cada um, obtendo $5c_1$, $4c_2$, $6c_3$ e $6c_4$. . . . 22
Figura 2.10 Esquema de classificação multiestágio proposto por Pudil et al. (1992) 28

Figura 3.1 Visão geral da abordagem em cascata de dois níveis 33
Figura 3.2 Fase de treinamento da abordagem em cascata 34

Figura 3.3 Gráfico de exemplo da relação entre a taxa de rejeições e a taxa de erros 37
Figura 3.4 Gráfico de exemplo do comportamento das taxas de rejeições e de erros com a variação do limiar de rejeição 38

Figura 4.1 Gráfico da relação entre a taxa de rejeições e a taxa de erros para a base de maior complexidade (LD) 50
Figura 4.2 Gráfico da relação entre a taxa de rejeições e a taxa de erros para a base de menor complexidade (IR) 50
Figura 4.3 Gráfico da relação entre a definição do limiar de rejeição e as taxas de erros e de rejeições para a base de maior complexidade (LD) 52
Figura 4.4 Gráfico da relação entre a definição do limiar de rejeição e as taxas de erros e de rejeições para a base de menor complexidade (IR) 52

Lista de Tabelas

Tabela 4.1 Descrição das bases de dados e separação das amostras 45
Tabela 4.2 Cálculo das medidas de complexidade das bases de dados 46
Tabela 4.3 Cálculo da posição média na complexidade das bases de dados . . . 47
Tabela 4.4 Melhores parâmetros de treinamento para os classificadores monolíticos em cada conjunto de dados 48
Tabela 4.5 Resultado da taxa de acertos dos classificadores monolíticos sem rejeição 49
Tabela 4.6 Limiar de rejeição do classificador SVM 51
Tabela 4.7 Resultado da classificação usando SVM com e sem rejeição 53
Tabela 4.8 Resultado da otimização da cardinalidade da técnica RSS 54
Tabela 4.9 Resumo dos melhores resultados obtidos para as combinações de conjuntos de classificadores 55
Tabela 4.10 Resumo dos melhores resultados obtidos para os métodos de seleção dinâmica de classificadores 57
Tabela 4.11 Resumo dos melhores resultados obtidos para os métodos propostos de classificação em cascata 58
Tabela 4.12 Análise do desempenho dos métodos de classificação em cascata que obtiveram os melhores resultados comparado com os experimentos utilizando seus respectivos SMCs fora da abordagem em cascata 59
Tabela 4.13 Desempenho do método de classificação em cascata proposto, baseado na acurácia (%), comparado com o classificador monolítico SVM e com os melhores resultados das abordagens de SMCs 61

Lista de Abreviações e Siglas

BAG	*Bagging*
BOO	*Boosting*
DCoL	*Data Complexity Library*
DCS-LA	Seleção Dinâmica de Classificadores baseada na Acurácia Local (*Dynamic Classifier Selection by Local Accuracy*)
F1	Taxa Máxima Discriminante de Fisher
KNN	K-Vizinhos Mais Próximos (*K-Nearest Neighbors*)
KNORA	Seleção Dinâmica de Classificadores baseada em K Oráculos Mais Próximos (*Dynamic Classifier Selection by K-Nearest-Oracles*)
LCA	Acurácia de Classe Local (*Local Class Accuracy*)
MLP	Perceptron de Múltiplas Camadas (*Multilayer Perceptron*)
MR	Posição Média (*Mean Rank*)
N2	Taxa da Distância Média Intra/inter Classes dos vizinhos mais próximos
N4	Não-linearidade do Classificador 1-vizinho mais próximo
NB	Naive Bayes
OLA	Acurácia Local Geral (*Overall Local Accuracy*)

RBF	Função de Base Radial (*Radial Basis Function*)
REL	Confiabilidade (*Reliability*)
RSS	*Random Subspace Selection*
SMC	Sistema de Múltiplos Classificadores
SVM	Máquinas de Vetores de Suporte (*Support Vector Machine*)
TA	Taxa de Acertos
TE	Taxa de Erros
TFV	Número Total de Características (*Total-number of Feature-Values*)
TR	Taxa de Rejeições

Resumo

Os métodos de combinação de classificadores, como a seleção dinâmica, são frequentemente criticados pelos seus custos computacionais, embora normalmente tenham taxas de acertos superiores aos métodos com um único classificador. O objetivo deste trabalho é propor um método de classificação em cascata de dois níveis. No primeiro nível será utilizado um classificador monolítico, que será responsável por rotular os exemplos mais fáceis, rejeitando aqueles que tiverem baixa confiança através da comparação com o limiar de rejeição. No segundo nível serão utilizados sistemas de múltiplos classificadores para rotular os exemplos rejeitados na fase anterior. Em outras palavras, a abordagem em cascata oferece uma estratégia interessante para conciliar os diferentes níveis de esforços necessários para lidar com padrões fáceis e difíceis encontrados em um problema de reconhecimento. Como esperado, os resultados experimentais mostraram que para alguns problemas mais de 93% das instâncias foram resolvidas logo no primeiro nível da cascata, evitando maiores esforços com a utilização de método de classificação mais complexo, com uma redução do custo de classificação de até 90%. Além disso, para a maioria dos problemas considerados difíceis até 97% das instâncias foram rejeitadas pelo primeiro nível, enquanto até 35% foram corretamente classificadas na segunda etapa. Em termos de precisão, a abordagem cascata tem mostrado ser capaz de melhorar até 15,2 pontos percentuais, enquanto ainda permite reduzir o esforço computacional na tarefa de classificação.

Palavras-chave: classificação em cascata; sistema de múltiplos classificadores; redução do custo de classificação

Abstract

The methods for combination of classifiers, as dynamic selection, are often criticized for its computational cost, but usually have a higher accuracy rate than methods with a single classifier. The objective of this work is to propose a two-level cascade classification method. In the first step of the cascade will be used a monolithic classifier, which is responsible for classifying the easiest patterns, rejecting those that have low confidence by comparison with the rejection threshold. In the second step multiple classifier systems are used to classify the patterns previously rejected. In other words, the cascade approach offers an interesting strategy to reconcile the different levels of effort to deal with easy and hard patterns found in a recognition problem. As expected, the experimental results showed that for some problems more than 93% of the instances can be processed in the first step of the cascade saving efforts by avoiding the use of a more complex classification method, with reduction in terms of cost of classification cost up to 90%. Moreover, for most difficult problems up to 97% of samples were rejected by the first step while up to 35% were correctly classified at the second step. In terms of accuracy, the cascade approach has shown to be able of improving it up to 15.2 percentage points while saving computational effort of the classification task.

Keywords: cascade classification; multiple classifier system; classification cost reduction

Capítulo 1

Introdução

Reconhecimento de padrões é uma área de estudo na qual a principal tarefa visa criar aplicações capazes de classificar objetos em classes previamente conhecidas. Estes objetos variam com a aplicação, podendo ser uma imagem, uma forma de onda (voz, música) ou qualquer outro objeto que possa ser caracterizado.

De cada objeto é extraído um conjunto de características (atributos) que o representa. Essas características são agrupadas em vetores, que serão usados tanto na criação do método de classificação quanto na etapa de reconhecimento de novas amostras.

Na abordagem de aprendizagem de máquina supervisionada, um conjunto de treinamento com objetos pré-classificados, ou seja, que já estão rotulados com uma classe conhecida, é utilizado para gerar um método de classificação que seja capaz de induzir a classe de uma amostra desconhecida. Existem diversos algoritmos que usam diferentes formas de aprendizagem e técnicas de classificação. Desta forma, um classificador pode ter diferentes capacidades e diferentes desempenhos dependendo da aplicação.

Uma estratégia de classificação para aumentar a taxa de reconhecimento é gerar um conjunto de classificadores fracos, capazes de reconhecer apenas parte das instâncias do problema, e depois combiná-los de modo a chegar a uma classificação final. Esta combinação pode ser realizada com diferentes técnicas que poderão utilizar a opinião (classificação) de todos os classificadores gerados, ou selecionar os mais promissores para cada exemplo.

A utilização de sistemas baseados em múltiplos classificadores tem sido defendida como uma alternativa para a difícil tarefa de construir um classificador monolítico capaz de absorver toda a variabilidade de um problema de classificação. A defesa está baseada

na observação de que diferentes classificadores podem gerar erros diferentes, ou seja, erros que apresentam baixa correlação. Assim, espera-se que um sistema composto de classificadores diversos deve ultrapassar o desempenho de um sistema baseado em um único classificador, uma vez que o primeiro sistema pode cobrir melhor a variabilidade do problema.

1.1 Definição do problema

A busca para alcançar maior acurácia na classificação pode muitas vezes levar à construção de sistemas mais complexos. Essa tendência no aumento da complexidade tem sido uma crítica frequente sobre o uso de múltiplos classificadores, principalmente quando o ganho, em termos de precisão, não é substancialmente suficiente para justificar a complexidade.

Uma discussão sobre a utilização de múltiplos classificadores é apresentada em Britto Jr., Sabourin e Oliveira (2014). Nesse trabalho foi comparado o desempenho de sistemas com seleção dinâmica de classificadores, contra um único classificador (o melhor disponível no conjunto de classificadores) e contra a combinação de todos os classificadores disponíveis. Os resultados demonstraram que, para alguns problemas de classificação, os métodos baseado em seleção dinâmica podem apresentar uma contribuição significativa no desempenho, mas este não é o caso geral. As análises mostraram que o desempenho dos métodos baseados em seleção dinâmica é dependente do problema, provavelmente relacionado à sua complexidade.

Outro aspecto a ser considerado, é que se pode observar diferentes níveis de dificuldade para discriminar entre as várias classes de um problema, ou seja, um problema de classificação é geralmente composto de padrões fáceis e difíceis. Uma instância do problema é dita como fácil quando ela é corretamente classificada, com uma confiança elevada, a uma única classe. Por outro lado, uma instância é considerada difícil quando o algoritmo de classificação não é capaz de atribuí-la a uma única classe, já que duas ou mais classes possuem altos níveis de confiança. Assim, sabe-se que um problema de classificação não é composto apenas de instâncias difíceis que sempre precisam de um esquema de classificação mais sofisticado. Há frequentes casos muito fáceis que podem ser resolvidos com regras muito simples.

1.2 Objetivos

Diversos autores defendem o uso da classificação em cascata para reduzir o risco de decisão (PUDIL et al., 1992), diminuir o custo computacional (ALPAYDIN; KAYNAK, 1998) e melhorar a precisão (ALIMOGLU; ALPAYDIN, 2001). Como mostrado em Fumera, Pillai e Roli (2004) e Oliveira, Britto Jr. e Sabourin (2005), um esquema em cascata é uma estratégia promissora para lidar com a classificação de padrões fáceis e difíceis. Nesse esquema, as instâncias rejeitadas pelo primeiro classificador são processadas pelo próximo nível da cascata, que utiliza um sistema de classificação mais complexo.

O objetivo deste trabalho é propor um método de classificação em cascata de dois níveis. A novidade do método é propor a utilização de classificadores monolíticos na primeira fase, enquanto que a segunda fase se baseia em um sistema de múltiplos classificadores. Tal abordagem em cascata é motivada pela observação de que, em muitos problemas de classificação, a maioria dos padrões pode ser explicada por uma regra simples, uma vez que eles são bem comportados, ou seja, um único classificador deve fazer o trabalho corretamente. No entanto, para os casos difíceis é necessário um esquema de classificação mais sofisticado.

Nesse sentido, no classificador proposto em cascata, os casos rejeitados pela primeira etapa, representada por um classificador monolítico, são tratados pela segunda etapa que utiliza um classificador mais sofisticado, representado por um sistema de múltiplos classificadores baseado em um conjunto onde os especialistas são criados com competência complementar. Neste trabalho, como um sistema de múltiplos classificadores, investigou-se a combinação de todos os classificadores dos conjuntos gerados através da utilização de diferentes técnicas de aprendizagem e também utilizando métodos de seleção dinâmica de classificadores.

Além disso, espera-se reduzir o custo computacional, em relação aos métodos tradicionais de combinação de classificadores, com a aplicação de um método mais sofisticado de classificação apenas nos casos mais difíceis.

Para o primeiro nível será configurado um limiar de rejeição, onde as amostras que não forem classificadas com o nível de confiança desejado serão rejeitadas, minimizando o erro nesta etapa. O primeiro estudo sobre a relação entre rejeição e erro foi feito por Chow (1970). Outra importante contribuição para a compreensão dos mecanismos de rejeição é dada em Fumera, Roli e Giacinto (2000). O segundo nível, por sua vez, receberá essas

rejeições e tentará classificá-las com um método baseado em múltiplos classificadores.

1.3 Desafios

O desafio do método proposto é conseguir reduzir o custo computacional utilizando uma abordagem em cascata como alternativa para os sistemas de múltiplos classificadores, porém sem perder precisão na tarefa de classificação.

Um dos pontos principais está na configuração do limiar de rejeição dos classificadores. Essa rejeição será responsável por separar os casos fáceis, aqueles que serão resolvidos no primeiro nível da cascata, dos casos difíceis, aqueles que serão rejeitados pelo classificador monolítico e serão então classificados por um método mais sofisticado, como um sistema baseado em múltiplos classificadores.

1.4 Hipóteses

A hipótese da pesquisa é que, através da combinação de um classificador monolítico com um sistema composto por diversos classificadores especialistas com uma abordagem em cascata, seja possível tratar adequadamente os problemas compostos de diferentes níveis de dificuldade, permitindo assim reduzir os esforços necessários para a tarefa de classificação. Em outras palavras, significa uma melhor relação entre precisão e complexidade, se comparado a um sistema de múltiplos classificadores, mantendo ao menos a mesma precisão na tarefa de classificação, principalmente para os problemas onde um sistema complexo não é necessário.

1.5 Estrutura do documento

Na Seção 2 é feita uma revisão da literatura sobre sistemas de classificação, rejeição, medidas de complexidade e trabalhos relacionados à classificação em cascata. O método proposto é descrito na Seção 3, enquanto a Seção 4 apresenta os experimentos e resultados obtidos. Finalmente, a Seção 5, traz a conclusão e sugestões de trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta uma revisão da literatura sobre sistemas de classificação em reconhecimento de padrões com foco em abordagens que tratam a opção de rejeição. Nas seções seguintes serão abordados métodos de reconhecimento como classificadores monolíticos, sistemas de múltiplos classificadores e classificação em cascata. Além disso, serão apresentadas métricas de avaliação de complexidade das bases de dados e dos métodos de classificação. Por fim, é feito uma revisão sobre os trabalhos relacionados à classificação em cascata.

Desde o surgimento dos computadores se pergunta se eles serão capazes de aprender, assim como os seres humanos. Os algoritmos existentes estão longe de terem um aprendizado tão eficiente quanto o das pessoas, porém se mostram efetivos para algumas tarefas de aprendizagem (MITCHELL, 1997). Devido à diversidade de aplicações, não existe um algoritmo perfeito que solucione qualquer problema, cada algoritmo oferece vantagens e desvantagens específicas.

Reconhecimento de padrões é uma área de aprendizagem de máquina que abrange o processamento de informações de problemas práticos, como reconhecimento de manuscrito, diagnóstico médico e categorização de músicas. Muitas vezes estes problemas são resolvidos por seres humanos sem muito esforço. No entanto, a solução utilizando computadores, em muitos casos, se revela extremamente difícil (BISHOP, 1995).

Um sistema clássico de reconhecimento de padrões pode ser dividido em seis principais etapas (Figura 2.1): aquisição do objeto de estudo, como uma imagem ou um sinal sonoro; pré-processamento, como a eliminação de ruídos; segmentação para isolar o objeto de interesse; caracterização do objeto através da extração de características;

classificação, obtendo uma pontuação para cada classe; e por fim, a decisão final que atribui uma classe ao objeto.

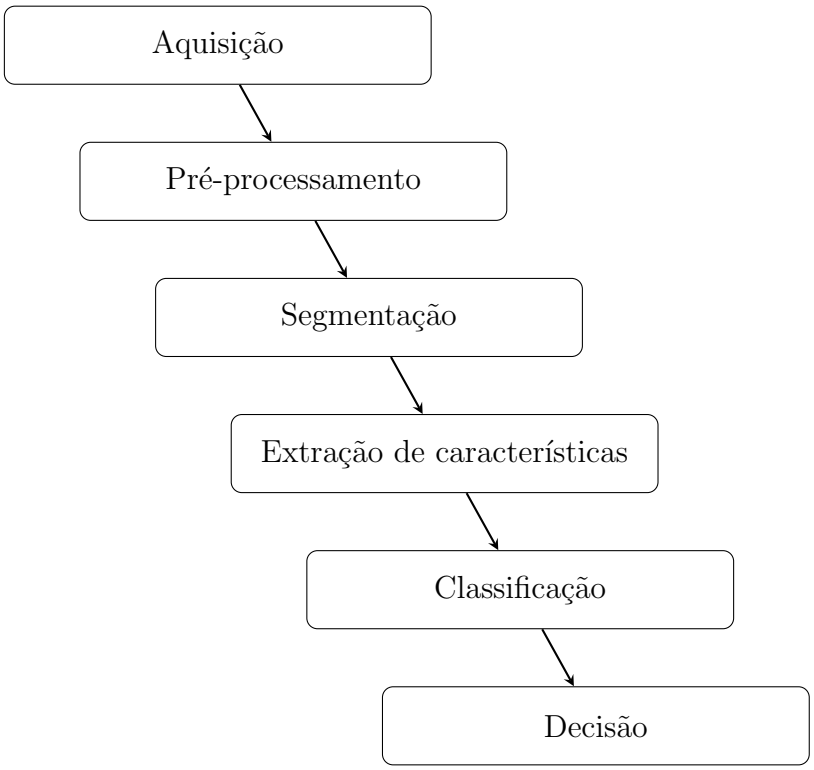

Figura 2.1: Etapas de um sistema clássico de reconhecimento de padrões

Existem dois modelos de aprendizagem de máquina, apresentados em Jain, Duin e Mao (2000): supervisionados e não supervisionados. O que os distingue é a presença ou não do atributo classe, que rotula os exemplos do conjunto de treinamento fornecido ao algoritmo. No aprendizado supervisionado, esse rótulo é conhecido, enquanto que no aprendizado não supervisionado os exemplos não estão previamente rotulados.

Neste trabalho serão abordados métodos de classificação supervisionados, que consiste em utilizar um conjunto de amostras conhecidas de cada uma das classes para criar um classificador de forma que ele seja capaz de posteriormente rotular amostras desconhecidas.

Para os estudos de reconhecimento de padrões é importante destacar algumas definições básicas:

- Amostras: são os exemplos ou instâncias de dados de um problema, também chamado de padrão.

- Atributos: também chamado de características, são os dados extraídos de uma amostra por meio de medida e/ou processamento.
- Classes: são os conjuntos de padrões que possuem características em comum. São categorias às quais as amostras podem pertencer.
- Classificação: é a ação de atribuir uma classe para as amostras com base em seus atributos.
- Ruídos: são as distorções, falhas ou imprecisões que podem ocorrer na aquisição de dados.

As bases de dados de um problema podem ser divididas em dois conjuntos: treinamento e testes. O conjunto de treinamento será utilizado para treinar o algoritmo, gerando um classificador. Já o conjunto de testes é utilizado na avaliação do desempenho do classificador criado. As amostras não podem se repetir nos dois conjuntos de dados, de forma que a intersecção do conjunto de treinamento e de testes deve ser vazio. Testar um classificador com amostras utilizadas no treinamento irá gerar uma avaliação falsa, já que o classificador pode se tornar tendencioso.

2.1 Classificadores monolíticos

Em um sistema de reconhecimento, cada amostra é um ponto em um espaço n-dimensional, onde n é o número de atributos que descrevem o padrão da amostra. Para classificar, divide-se este espaço de atributos utilizando um método de classificação, gerando limites de decisão de forma a separar os dados.

O treinamento é o processo de geração dos classificadores, que se baseia na análise das amostras cuja resposta correta (classe) já é conhecida. Na fase de treinamento, é desejável que o sistema tenha um bom poder de generalização, e ao receber amostras desconhecidas, seja capaz de classificá-las corretamente em uma das n classes pré-definidas pelo problema em questão.

O conjunto de treinamento utilizado pelos classificadores é formado por um conjunto de atributos extraídos de todas as amostras. Usualmente é representado por uma matriz onde as linhas são as amostras e as colunas os atributos, sendo que a última coluna traz a classe da amostra. Um exemplo de matriz é apresentado na Equação 2.1,

onde A é a matriz de atributos, com m sendo a quantidade de amostras, n a quantidade de atributos e y_i é a classe da amostra i.

$$A_{m,n+1} = \left[\begin{array}{cccc|c} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & y_1 \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} & y_m \end{array}\right] \tag{2.1}$$

Para rotular uma nova amostra, ou seja, atribuir o valor de y é preciso encontrar a função f, $y_i = f(x_i)$, onde x_i é um vetor de atributos $x_i = a_{1,1}$,$a_{1,2}$,$\cdots$,$a_{1,n}$, cujos componentes são valores discretos ou contínuos.

Dessa forma, dado um conjunto de treinamento, o algoritmo de classificação constrói uma hipótese que deve aproximar da verdadeira função f, tal que, dado um novo exemplo ele seja capaz de predizer sua classe.

O desempenho de um classificador é avaliado utilizando um conjunto de amostras rotuladas disjunto do conjunto de treinamento, o qual é denominado de conjunto de testes.

Os algoritmos supervisionados podem ser do tipo *eager* ou *lazy*. Os algoritmos *eager* usam o conjunto de treinamento para construir f e, após isso, descarta o conjunto, já que precisa apenas de f. Já os algoritmos *lazy* não constroem explicitamente uma hipótese e necessitam consultar as amostras de treinamento.

As próximas subseções apresentam os classificadores monolíticos que serão avaliados neste trabalho. São eles: K-Vizinhos Mais Próximos (KNN, *K-Nearest Neighbors*); Perceptron de Múltiplas Camadas (MLP, *Multilayer Perceptron*); Árvore de Decisão C4.5; Naive Bayes (NB) e Máquinas de Vetores de Suporte (SVM, *Support Vector Machine*);

2.1.1 K-Vizinhos Mais Próximos

O K-Vizinhos Mais Próximos (KNN, *K-Nearest Neighbors*) é um algoritmo de aprendizado supervisionado do tipo *lazy* (preguiçoso) baseado em instâncias. O aprendizado do KNN consiste em simplesmente armazenar o conjunto de treinamento. Quando uma nova amostra precisa ser classificada, o algoritmo recupera os K exemplos no conjunto de treinamento mais próximos da nova amostra, utilizando uma medida de distância, e com base no rótulo destes exemplos realiza a classificação (MITCHELL, 1997).

O algoritmo requer pouco esforço durante o treinamento, já que simplesmente armazena os exemplos. A desvantagem é que possui alto custo computacional na tarefa de classificação, por armazenar e examinar todos os dados de treinamento a cada classificação (MICHIE et al., 1994; MITCHELL, 1997).

A utilização do KNN envolve a escolha de uma métrica de distância e a otimização do número de vizinhos K. Em Mitchell (1997) utiliza-se a distância Euclidiana para encontrar os vizinhos mais próximos de uma amostra, que é uma das métricas mais simples e mais utilizadas. Dado que uma instância arbitrária x seja descrita pelo vetor

$$\{a_1(x),a_2(x),...a_n(x)\} \tag{2.2}$$

onde $a_r(x)$ indica o valor do atributo r para a amostra x, então a distância entre duas instâncias x_i e x_j é definida como $d(x_i,x_j)$, onde

$$d(x_i,x_j) = \sqrt{\sum_{r=1}^{n} \Big(a_r(x_i) - a_r(x_j)\Big)^2} \tag{2.3}$$

Já a escolha do valor de K pode ser feita por experimentos utilizando métodos de validação cruzada, dividindo o conjunto de treinamento e utilizando uma parte para ser classificada pelo algoritmo (MICHIE et al., 1994).

2.1.2 Perceptron de Múltiplas Camadas

O algoritmo Perceptron de Múltiplas Camadas (MLP, *Multilayer Perceptron*) é inspirado no sistema nervoso humano, onde as informações são processadas através de neurônios interconectados (BISHOP, 1995). O MLP é uma rede onde a informação se propaga da entrada para a saída, passando por múltiplas camadas intermediárias, que são geralmente utilizadas para fazer um gargalo, forçando a rede a gerar um modelo simples que seja capaz de generalizar os padrões desconhecidos (MICHIE et al., 1994).

As entradas são alimentadas com valores de cada característica e as saídas fornecem o valor da classe. Com uma camada de neurônios, a saída é uma combinação linear ponderada das entradas. A fase de treinamento consiste na otimização iterativa dos pesos que ligam os neurônios através da minimização da média quadrada da taxa de erros (DEPEURSINGE et al., 2010).

2.1.3 Árvore de Decisão

A Árvore de Decisão C4.5 é um algoritmo que divide o espaço de características sucessivamente, escolhendo a cada passo a característica com maior capacidade de discriminação para um novo nível na árvore. Este método é robusto para características ruidosas, já que apenas os atributos com alto ganho de informação são usados, no entanto, é sensível a variabilidade de dados (DEPEURSINGE et al., 2010).

A cada nó de uma árvore de decisão é realizado um teste lógico, feito a partir de uma característica extraída da amostra, que define qual nó filho continuará a execução. A decisão ocorre nos nós folhas que representam as classes do problema.

2.1.4 Naive Bayes

O algoritmo Naive Bayes (NB) classifica uma amostra baseado na probabilidade máxima *a posteriori* considerando seu conjunto de características. A probabilidade $P(\omega_i|\vec{a}_x)$ da classe ω_i, dado um vetor de atributos $\vec{a}$ da amostra x, é determinada utilizando o teorema de Bayes, definido como:

$$P(\omega_i|\vec{a}_x) = \frac{P(\vec{a}_x|\omega_i)P(\omega_i)}{P(\vec{a}_x)} \tag{2.4}$$

Segundo Michie et al. (1994), o algoritmo é melhor se os atributos são condicionalmente independentes dado uma classe, ou seja, a informação de um evento não é informativa sobre nenhum outro. Ele é recomendado para conjuntos de treinamento grande ou moderado.

2.1.5 Máquinas de Vetores de Suporte

O algoritmo Máquinas de Vetores de Suporte (SVM, *Support Vector Machine*) é uma técnica de classificação supervisionada e não paramétrica, já que não armazena os exemplos de treinamento para classificar novas amostras. Além disso, o SVM é um classificador binário (duas classes).

O SVM constrói um hiperplano, como superfície de decisão, que maximiza a margem de separação entre os exemplos positivos e negativos. Quando um problema não é linear, é necessário mapear o conjunto de treinamento de seu espaço original para

um novo espaço de maior dimensão, denominado espaço de características, que é linear. Para isso é preciso encontrar uma transformação não linear.

Para realizar a transformação de espaço é empregada uma função *Kernel*. As quatros funções básicas citadas em Hsu, Chang e Lin (2010) são apresentadas a seguir. Considere γ, r e d como parâmetros específicos de cada *Kernel*.

- Linear: $K(x_i,x_j) = x_i^T x_j$;
- Polinomial: $K(x_i,x_j) = (\gamma x_i^T x_j + r)^d$, $\gamma > 0$;
- Função de Base Radial: $K(x_i,x_j) = exp(-\gamma||x_i - x_j||^2)$, $\gamma > 0$;
- Sigmóide: $K(x_i,x_j) = tanh(\gamma x_i^T x_j + r)$.

A Função de Base Radial (RBF, *Radial Basis Function*) é considerada uma escolha razoável em Hsu, Chang e Lin (2010). Ao contrário do *kernel* linear, o RBF pode tratar os casos em que a relação entre os rótulos de classes e atributos não são lineares. Além disso, apresenta desempenho parecido com os *kernels* linear e sigmóide, e possui um número menor de hiperparâmetros do que o *kernel* polinomial.

O algoritmo SVM foi desenvolvido para problemas de reconhecimento de padrões binários, ou seja, que possui apenas duas classes. Uma abordagem clássica para resolver problemas de reconhecimento de padrões multiclasses é considerar o problema como uma coleção de problemas de classificação binária (WESTON; WATKINS, 1999). Há duas formas de combinar os múltiplos casos: um-contra-todos e um-contra-um. O método um-contra-todos treina M classificadores ("um" positivo e o "resto" negativo) para um problema de M classes, e obtém uma classe para uma amostra que corresponda a maior distância positiva.

Já o método um-contra-um treina um classificador para cada par de classes, ou seja, um problema de M classes terá $M(M-1)/2$ classificadores. Um método de combinação por votação pode ser utilizado para se obter uma decisão final.

2.2 Sistemas de múltiplos classificadores

Quando há um problema de classificação difícil, com poucos dados ou com muito ruído, o uso de apenas um classificador pode não ser suficiente. Criar um

classificador monolítico para cobrir toda a variabilidade inerente à maioria dos problemas de reconhecimento de padrões é impraticável (BRITTO JR.; SABOURIN; OLIVEIRA, 2014). Os sistemas de múltiplos classificadores surgem para aproveitar as vantagens de diversos esquemas de classificação e aumentar o desempenho na tarefa de reconhecimento.

Um *ensemble* é um conjunto de classificadores cujas decisões individuais são combinadas de alguma forma para classificar novas amostras. *Ensembles* são frequentemente mais precisos do que os classificadores individuais (DIETTERICH, 2000).

Dietterich (2000) aponta três razões fundamentais pelas quais é possível criar conjuntos de classificadores que sejam melhores do que um único classificador. A primeira razão é estatística. Considerando ter um conjunto de amostras de treinamento Tr e diferentes classificadores com bom desempenho em Tr. Se for adotado um único classificador como a solução do problema, corre-se o risco de fazer uma má escolha. Cada um desses classificadores pode possuir um desempenho de generalização diferente, portanto escolher um único classificador para resolver o problema não é uma boa saída. A solução mais adequada é utilizar todos os classificadores para resolver o problema e considerar como resposta a "média" das respostas de todos os classificadores (DIETTERICH, 2000).

A Figura 2.2 ilustra essa situação. D^* indica o melhor dos classificadores para o problema. A curva externa denota o espaço onde se encontram todos os classificadores, enquanto a região sombreada interna denota os "bons" classificadores. Ao se combinar os classificadores espera-se que o classificador resultante fique o mais próximo possível de D^* (KUNCHEVA, 2004).

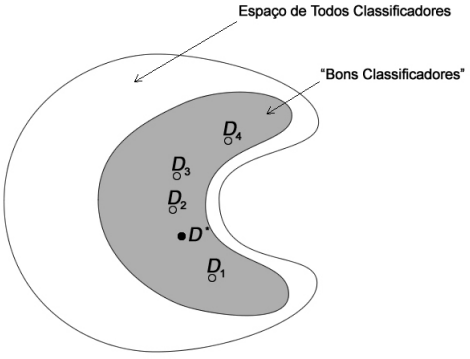

Figura 2.2: A razão estatística para combinação de classificadores

A segunda razão é computacional. Um classificador inicia em algum lugar no espaço dos possíveis classificadores e através de um processo de treinamento ele termina ficando mais próximo de um classificador ideal D^*. Esse comportamento está ilustrado na Figura 2.3, onde D^* indica o melhor dos classificadores para o problema, o espaço fechado mostra o espaço de todos os classificadores e as linhas tracejadas são as trajetórias hipotéticas para os classificadores durante o treinamento (KUNCHEVA, 2004).

A agregação de classificadores pode levar a um novo classificador que é ainda mais próximo de D^* do que qualquer um dos classificadores individuais (DIETTERICH, 2000).

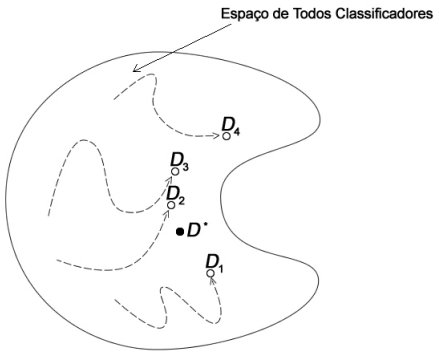

Figura 2.3: A razão computacional para combinação de classificadores

A terceira razão é representacional. Muitas vezes as divisões entre as classes de determinado problema são muito complexas ou até mesmo não podem ser implementadas em determinados classificadores (DIETTERICH, 2000).

A Figura 2.4 mostra o caso em que o classificador ideal D^* está fora do espaço dos classificadores possíveis, onde D^* indica o melhor dos classificadores para o problema e o espaço fechado mostra o espaço de todos os classificadores (KUNCHEVA, 2004).

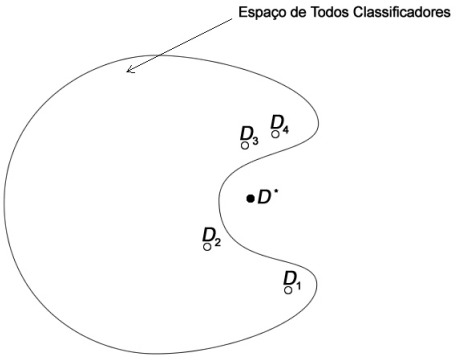

Figura 2.4: A razão representacional para combinação de classificadores

Os sistemas de múltiplos classificadores podem ser divididos em três fases, representadas na Figura 2.5: geração do conjunto (*pool*) de classificadores; seleção de classificadores; e, integração. As duas últimas fases são facultativas, podendo não estar presentes em algumas abordagens.

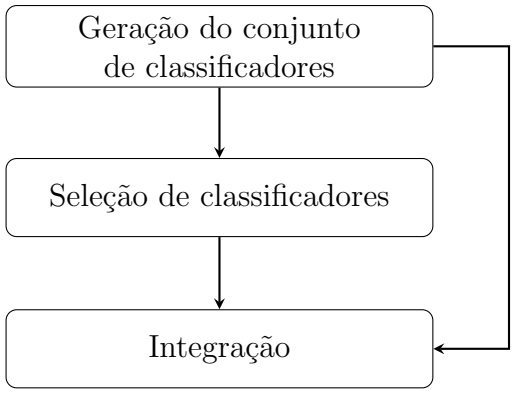

Figura 2.5: Possíveis fases de um sistema de múltiplos classificadores

A primeira fase pode ser formada por um conjunto de classificadores gerados individualmente ou utilizar alguma técnica de aprendizagem para gerar um conjunto diversificado de classificadores.

Na fase de seleção, o Sistema de Múltiplos Classificadores (SMC) pode selecionar um único classificador ou selecionar um subconjunto dos classificadores mais promissores. Essa fase não está presente quando utilizada a combinação de todos os classificadores, já que não há necessidade da seleção.

Por fim, na fase de integração, uma decisão final é tomada baseada na predição de cada um dos classificadores selecionados, utilizando alguma estratégia de fusão para

as respostas. Em sistemas que selecionam um único classificador, por exemplo, essa fase não é necessária.

2.2.1 Geração do conjunto de classificadores

A primeira fase de um SMC é responsável por gerar um *pool* (conjunto) de classificadores base. Para isso, é considerada alguma estratégia para criar especialistas diversificados e acurados. A ideia é gerar classificadores que cometam erros diferentes e que, consequentemente, mostrem alguma complementariedade (BRITTO JR.; SABOURIN; OLIVEIRA, 2014).

Para obter a variedade desejada, pode-se variar as informações utilizadas na criação dos classificadores, como alterações nos parâmetros iniciais, utilização de diferentes subconjuntos de treinamento, ou usando diferentes subespaços de características. Nesta abordagem, quanto maior a diversidade presente no conjunto, melhor será o resultado final. Normalmente ela apresenta melhor desempenho do que um classificador único.

Os algoritmos mais comumente utilizados na literatura são: *Bagging* (BAG), *Boosting* (BOO) e *Random Subspace Selection* (RSS).

A técnica *Bagging*, um acrônimo para *Bootstrap AGGreatING*, foi proposto por Breiman (1996). Essa técnica seleciona aleatoriamente diferentes subconjuntos de amostras de treinamento. É esperado que uma mesma amostra esteja presente em vários subconjuntos. Cada subconjunto será utilizado para treinar um único classificador. A hipótese é de que classificadores treinados por diferentes subconjuntos de amostras apresentem diferentes comportamentos.

O método *Boosting* apresentada em Avnimelech e Intrator (1999) é similar ao *Bagging*. Ele também usa subconjuntos de amostras para treinar classificadores, mas ao invés de escolher aleatoriamente, ele se baseia na dificuldade de cada amostra, onde amostras difíceis tem uma maior probabilidade de serem selecionadas para o treinamento do que as fáceis. Em ambos os métodos, a quantidade de amostras utilizadas para treinar cada classificador é definida como uma percentagem da base de treinamento.

Já a técnica *Random Subspace Selection*, proposta por Ho (1998), gera subespaços aleatórios de características (atributos). Então, cada subespaço é utilizado para treinar um classificador diferente. A quantidade de características de cada subconjunto é definida pela cardinalidade do método.

2.2.2 Seleção de classificadores

A segunda fase de um SMC é a seleção de classificadores, que pode ocorrer de duas maneiras: estática ou dinâmica. Na abordagem estática, todas as instâncias de teste serão avaliadas pelo mesmo subconjunto de classificadores, que serão escolhidos em um momento anterior à fase de teste. Já na abordagem dinâmica, os classificadores são selecionados se baseando em características ou regiões de decisão da instânica de teste, podendo ser um único classificador ou *ensembles* (subconjuntos) de classificadores. Dessa forma, no momento da classificação serão escolhidos um ou mais classificadores que são considerados os mais promissores para cada amostra a ser classificada.

Quando a seleção dinâmica retornar um *ensemble*, cada um dos classificadores que o compõe irá classificar paralelamente a amostra de teste. Os resultados obtidos são enviados para o método de fusão que irá combinar as respostas chegando à classificação final.

Britto Jr., Sabourin e Oliveira (2014) faz uma revisão da literatura citando os principais métodos de seleção dinâmica de classificadores. Os métodos utilizam diferentes abordagens com base no ranking, na acurácia, na probabilidade, no comportamento e no oráculo.

As próximas subseções apresentam os quatro métodos de seleção que foram considerados para este trabalho.

2.2.2.1 DCS-LA

No método de Seleção Dinâmica de Classificadores baseada na Acurácia Local (DCS-LA, *Dynamic Classifier Selection by Local Accuracy*) a principal característica é estimar a acurácia do classificador como uma simples porcentagem das amostras classificadas corretamente. Duas variações do método são propostas por Woods, Kegelmeyer Jr. e Bowyer (1997): Acurácia Local Geral (OLA, *Overall Local Accuracy*) e Acurácia de Classe Local (LCA, *Local Class Accuracy*).

A variação OLA calcula a acurácia local geral dos classificadores na região local do espaço de características próximas à amostra desconhecida no conjunto de dados de treinamento (ver Algoritmo 1). Um exemplo didático do processo de seleção utilizando OLA é apresentado na Figura 2.6.

Algoritmo 1 DCS-LA OLA - Seleção dinâmica baseada na acurácia local geral

Entrada: conjunto de classificadores C; subconjunto de treinamento Tr; amostra de teste t; e o número de vizinhos K;

Saída: classificação da amostra t na classe ω;

Submeta t para todos os classificadores em C;

se todos classificadores concordam com a classe ω para a amostra t **então**

retorne a classe ω para a amostra t;

else

Encontre Ψ como os K vizinhos mais próximos da amostra t em Tr;

para cada classificador c_i em C **faça**

Calcule OLA_i como a porcentagem de classificações corretas de c_i em Ψ;

fim para

Selecione o melhor classificador para t como $c^* = arg\ max_i(OLA_i)$;

$\omega = c^*(t)$, a classificação de c^* para a amostra t;

retorne a classe ω para a amostra t;

fim se

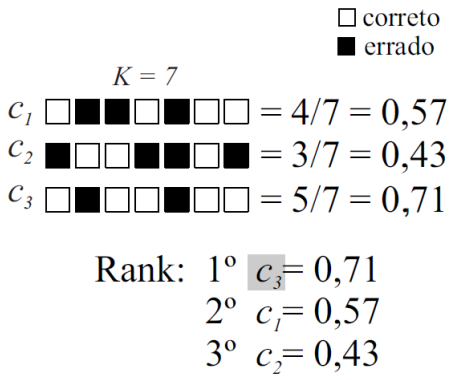

Figura 2.6: Exemplo do método OLA com três classificadores (c_1, c_2 e c_3) rotulando sete vizinhos da amostra de teste. O classificador c_3 é selecionado por apresentar maior taxa de acertos geral.

Fonte: Almeida (2014)

No método DCS-LA baseado na Acurácia de Classe Local, calcula-se LCA para cada classificador como a porcentagem de classificações corretas na região local, no entanto, considerando apenas os exemplos onde o classificador obteve a mesma classe

obtida para a amostra desconhecida (ver Algoritmo 2). Um exemplo didático do processo de seleção utilizando LCA é apresentado na Figura 2.7.

Algoritmo 2 DCS-LA LCA - Seleção dinâmica baseada na acurácia de classe local

Entrada: conjunto de classificadores C; subconjunto de treinamento Tr; amostra de teste t; e o número de vizinhos K;

Saída: classificação da amostra t na classe ω;

Submeta t para todos os classificadores em C;

se todos classificadores concordam com a classe ω para a amostra t **então**

retorne a classe ω para a amostra t;

else

Encontre Ψ como os K vizinhos mais próximos da amostra t em Tr;

para cada classificador c_i em C **faça**

$\omega_{i,t} = c_i(t)$, a classificação de c_i para a amostra t;

Calcule LCA_i como a porcentagem de classificações corretas de c_i em Ψ somente quando $\omega_{i,k} = \omega_{i,t}$;

fim para

Selecione o melhor classificador para t como $c^* = arg\ max_i(LCA_i)$;

$\omega = c^*(t)$, a classificação de c^* para a amostra t;

retorne a classe ω para a amostra t;

fim se

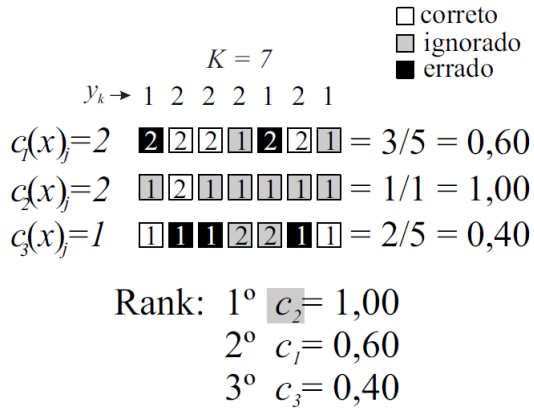

Figura 2.7: Exemplo do método LCA com três classificadores rotulando sete vizinhos da amostra de teste. As classificações onde os rótulos atribuidos aos vizinhos são diferente da amostra de teste, são ignoradas. O classificador c_2 é selecionado por apresentar maior taxa de acertos local.

Fonte: Almeida (2014)

2.2.2.2 KNORA

Diferente da maioria dos métodos de seleção dinâmica, o método Seleção Dinâmica de Classificadores baseada em K Oráculos Mais Próximos (KNORA, *Dynamic Classifier Selection by K-Nearest-Oracles*), proposto por Ko, Sabourin e Britto Jr. (2008), não é projetado para encontrar o classificador com maior probabilidade de sucesso. No entanto, o método seleciona um conjunto de classificadores mais adequados para cada amostra. Neste trabalho será avaliado dois esquemas diferentes usando KNORA: Eliminate e Union.

Para classificar uma nova amostra t, o método KNORA-Eliminate localiza as K instâncias presentes no conjunto de treinamento mais próximas a t. O algoritmo então busca os classificadores do *pool* que classificam corretamente todas as K amostras.

Caso não encontre nenhum classificador capaz de classificar todas as K amostras, o valor de K é decrementado em 1 e o processo é repetido do início (ver Algoritmo 3). Um exemplo didático do processo de seleção utilizando KNORA-Eliminate é apresentado na Figura 2.8.

Algoritmo 3 KNORA-Eliminate - Seleção dinâmica baseada em k oráculos mais próximos com estratégia *Eliminate*

Entrada: conjunto de classificadores C; meta-espaço sVa onde para cada amostra de treinamento é relacionado os classificadores que a reconhecem corretamente; amostra de teste t; e o número de vizinhos K;

Saída: EoC^* como o conjunto de classificadores mais promissores para a amostra t;

enquanto $K > 0$ **faça**

Encontre Ψ como os K vizinhos mais próximos da amostra t em sVa;

para cada classificador c_i em C **faça**

se c_i classifica corretamente todas amostras de Ψ **então**

$EoC^* = EoC^* \cup c_i$;

fim se

fim para

se $EoC^* == \emptyset$ **então**

$K = K - 1$;

else

break;

fim se

fim enquanto

se $EoC^* == \emptyset$ **então**

Encontre o classificador c_i que reconheça corretamente mais amostras em Ψ;

Obtenha EoC^* como os classificadores capazes de reconhecer a mesma quantidade de amostras de c_i;

fim se

retorne EoC^*;

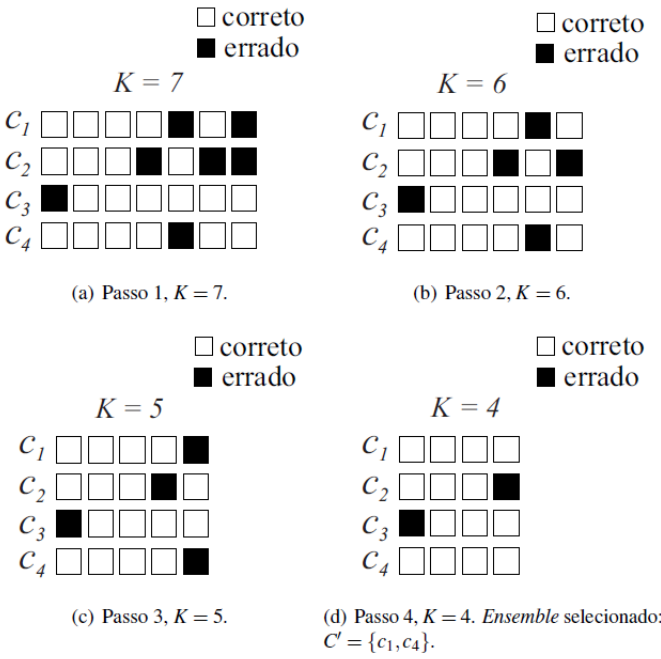

Figura 2.8: Exemplo do método KNORA-E com quatro classificadores e sete vizinhos. A execução teve quatro iterações para selecionar os classificadores c_1 e c_4.

Fonte: Almeida (2014)

Para classificar uma nova amostra t, o método KNORA-Union, de forma semelhante ao método KNORA-Eliminate, também localiza as K amostras presentes no conjunto de treinamento mais próximas a t. O algoritmo então busca os classificadores do *pool* que classificam corretamente algumas das K amostras.

A classificação é feita pela combinação de todos os classificadores que acertaram ao menos uma das K amostras. Se determinado classificador acertar N das K amostras mais próximas de t, então ele tem direito a N votos no esquema de combinação (ver Algoritmo 4). Um exemplo didático do processo de seleção utilizando KNORA-Union é apresentado na Figura 2.9.

Algoritmo 4 KNORA-Union - Seleção dinâmica baseada em k oráculos mais próximos com estratégia *Union*

Entrada: conjunto de classificadores C; meta-espaço sVa onde para cada amostra de treinamento é relacionado os classificadores que a reconhecem corretamente; amostra de teste t; e o número de vizinhos K;

Saída: EoC^* como o conjunto de classificadores mais promissores para a amostra t;

```
Encontre Ψ como os K vizinhos mais próximos da amostra t em sVa;
para cada amostra ψ_i em Ψ faça
    para cada classificador c_j em C faça
        se c_j classifica corretamente ψ_i então
            EoC* = EoC* ⊎ c_j;
        fim se
    fim para
fim para
retorne EoC*;
```

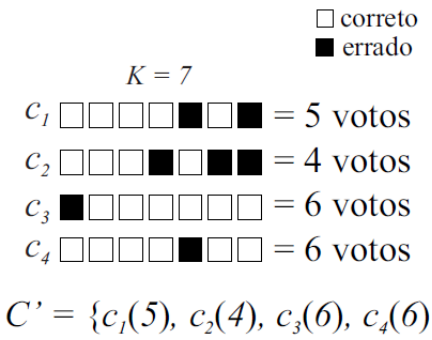

Figura 2.9: Exemplo do método KNORA-U com quatro classificadores e sete vizinhos. Todos os classificadores foram selecionados considerando a quantidade de votos (acertos) de cada um, obtendo $5c_1$, $4c_2$, $6c_3$ e $6c_4$.

Fonte: Almeida (2014)

Os classificadores selecionados pelo KNORA são combinados utilizando a votação majoritária. Outros métodos de fusão de classificadores serão abordados na próxima subseção.

2.2.3 Fusão de classificadores

A terceira fase de um SMC consiste em aplicar os classificadores selecionados para reconhecer a amostra de teste. Nos casos onde todos os classificadores são usados sem seleção, ou quando um *ensemble* de classificadores é selecionado, uma estratégia de fusão para combinar os resultados de cada classificador se torna necessária para gerar a resposta final.

Em Kittler (1998) e Kuncheva (2004) são apresentados diversos métodos de fusão, como as regras de combinações baseadas na: Média, Produto, Soma, Máximo e Mínimo. Um dos métodos de fusão mais utilizados e simples de implementar é o Voto Majoritário (KUNCHEVA, 2004). Nele, a saída de cada classificador é considerada um voto, e a classe mais votada é atribuída à amostra na classificação final.

2.3 Rejeição

Em reconhecimento de padrões, a rejeição foi introduzida para evitar erros excessivos (CHOW, 1970; PUDIL et al., 1992). Em problemas binários (que possuem apenas duas classes), por exemplo, a rejeição pode ser considerada como uma terceira saída deste problema, um espécie de pseudoclasse (PUDIL et al., 1992).

A rejeição tem como objetivo diminuir as classificações incorretas de um sistema de classificação, aumentando a confiabilidade do resultado. A meta de rejeição ideal é ser capaz de recusar todas as instâncias classificadas erroneamente e aceitar as instâncias classificadas corretamente. A recusa pode ocorrer por falta de evidência suficiente para tomar uma decisão, por nenhuma hipótese parecer adequada ou por mais de uma hipótese parecer adequada (KOERICH, 2004).

O desempenho de um sistema de reconhecimento de padrões pode ser caracterizado pela taxa de erros e taxa de rejeição. Quando o padrão de uma classe é identificado como de uma classe diferente, ocorre um erro ou uma classificação falsa. A rejeição ocorre quando o classificador retém a sua decisão e o padrão é rejeitado para uma manipulação excepcional, tais como reanálise ou inspeção manual (CHOW, 1970).

Quando se inclui a rejeição em um problema, é importante observar que alguns casos que seriam corretamente classificados podem ser transformados em rejeição (CHOW,

1970). Chow descreve pela primeira vez uma regra de rejeição ideal. Considerando que $m(v)$ seja o máximo da probabilidade a posteriori de uma classe para o padrão (v). Nesta regra, um padrão é considerado aceito sempre que

$$m(v) \geq 1 - t, \tag{2.5}$$

e é considerado rejeitado sempre que

$$m(v) < 1 - t. \tag{2.6}$$

O parâmetro t na regra de decisão pode ser chamado de limiar de rejeição, tal que $0 \leq t \leq 1$. A regra é rejeitar os padrões sempre que o máximo da probabilidade a posteriori seja menor do que o limiar de rejeição $(1 - t)$.

Em Fumera, Roli e Giacinto (2000) é proposto o uso de múltiplos limiares de rejeição para as diferentes classes de dados. Limiares definidos por classe podem fornecer uma melhor relação rejeição-erro do que um único limiar global. Oliveira, Britto Jr. e Sabourin (2005) defende que essa relação pode ser melhorada se um algoritmo apropriado é utilizado para encontrar esses limites por classe.

2.4 Classificação em cascata

Os métodos baseados em múltiplos classificadores vistos até aqui são combinados paralelamente. Todos os classificadores selecionados rotulam a amostra desconhecida de forma paralela, e todas essas decisões são enviadas para um método de fusão, que produzirá o resultado final.

Uma alternativa para a combinação paralela é a combinação em série, que pode ser em duas abordagens: redução de classe ou reavaliação. Na redução de classes, a cada estágio do sistema o número de classes candidatas é reduzido. Na reavaliação os padrões rejeitados em níveis anteriores são avaliados novamente. Se o nível de confiança do classificador é baixo, a amostra deverá ser rejeitada novamente.

A classificação em cascata pode ser considerada então uma combinação em série de múltiplos classificadores, onde a cada estágio existe um classificador, ou um outro sistema de múltiplos classificadores, atuando no sistema. A amostra não precisa percorrer todos

os estágios, podendo ser classificada em qualquer etapa. O objetivo é reduzir o custo computacional, classificando as amostras mais fáceis no primeiro nível, que possui um método de classificação menos sofisticado.

Na literatura a abordagem em cascata pode ser encontrada com outros termos, como classificação multiestágio. Quando utilizado este último termo, não confundir com métodos de dois ou mais estágios em que a rejeição não é considerada em nenhuma etapa, ou seja, todas as amostras de testes devem percorrer todos os estágios para serem classificadas.

Quando se fala em rejeições, uma das questões que surge é sobre o que fazer com os casos rejeitados. Para Pudil et al. (1992) existem basicamente dois caminhos possíveis. O primeiro é quando a rejeição é uma decisão aceitável. Nesse caso, pode-se sugerir uma ação alternativa para aqueles padrões rejeitados. O segundo é quando a rejeição não é aceita como resultado final. Neste caso, as instâncias rejeitadas podem ser processadas por um estágio avançado, constituído de um sistema de reconhecimento de padrões que utiliza mais informações.

Uma revisão da literatura mais aprofundada sobre classificação em cascata é apresentado na Seção 2.6.

2.5 Métricas de complexidade

Nas subseções seguintes serão abordadas algumas métricas utilizadas no presente trabalho para avaliar o desempenho do método de classificação proposto, estimando a complexidade das bases de dados utilizadas e, calculando a estimativa de redução de custo computacional com a utilização da classificação em cascata.

2.5.1 Estimativa de complexidade dos dados

Uma das hipóteses apresentadas na introdução deste trabalho é de que o método de classificação em cascata proposto é promissor por tratar adequadamente os padrões fáceis logo no primeiro nível da cascata, com um custo computacional reduzido, e os padrões difíceis serão encaminhados para o segundo nível da cascata, que possui um esquema de classificação mais sofisticado, embora tenha um maior custo computacional.

Para avaliar o nível de dificuldade dos problemas de classificação utilizados nesta pesquisa, foram selecionadas três medidas de complexidade amplamente difundidas na literatura (HO; BASU, 2002; SANCHEZ; MOLLINEDA; SOTOCA, 2007).

A taxa máxima discriminante de Fisher, F1, demonstrada na Equação 2.7, é uma métrica bem conhecida de sobreposição de classe que é calculada ao longo de cada dimensão de característica única como indicado na Equação 2.7, onde M é o número de classes e μ é a média global. Os valores n_i, μ_i e s_j^i são, respectivamente, o número de amostras, a média e a amostra de índice j da classe i. Nesta definição de F1, δ é a distância Euclidiana. Um valor elevado de F1 indica a presença de características discriminantes e, portanto, representa um problema de classificação mais fácil.

$$F1 = \frac{\sum_{i=1}^{M} n_i \cdot \delta(\mu, \mu_i)}{\sum_{i=1}^{M} \sum_{j=1}^{n_i} \delta(s_j^i, \mu_i)} \tag{2.7}$$

A medida N2, taxa da distância média intra/inter classes dos vizinhos mais próximos, é baseada em uma separabilidade não-paramétrica de classes. Ela compara a dispersão dentro da classe (intra) com a separabilidade entre classes (inter), como demonstra a Equação 2.8. Para isto, $\eta_1^{intra}(s_i)$ e $\eta_1^{inter}(s_i)$ representam o número de vizinhos mais próximos da amostra s_i inter e intraclasses, enquanto δ é a distância Euclidiana. Um valor baixo de N2 indica uma alta separabilidade e, consequentemente, representa um problema de classificação mais fácil.

$$N2 = \frac{\sum_{i=1}^{n} \delta\big(\eta_1^{intra}(s_i), s_i\big)}{\sum_{i=1}^{n} \delta\big(\eta_1^{inter}(s_i), s_i\big)} \tag{2.8}$$

A última medida, N4 - não-linearidade do classificador 1-vizinho mais próximo (NN), representada na Equação 2.9, cria um conjunto de testes a partir de uma base de treinamento usando interpolação linear entre pares sorteados de pontos da mesma classe. Em seguida, a taxa de erros do classificador NN sobre este conjunto de teste é mensurada. Em outras palavras, N4 utiliza a taxa de erros do NN com uma base de treinamento para descrever a não-linearidade do classificador. Um valor baixo de N4 indica alta separabilidade e, consequentemente, representa um problema de classificação mais fácil.

$$N4 = taxa\_erros\big(NN(conj\_treinamento)\big) \tag{2.9}$$

2.5.2 Estimativa de custo de classificação

Uma das medidas possíveis para comparar métodos de classificação é o custo computacional exigido. O método pode ser avaliado numa determinada base de dados, avaliando diretamente o tempo real de execução. Porém, o tempo de execução não é uma medida confiável, uma vez que envolve diversos fatores como poder de processamento, memória e recursos disponíveis no momento da execução.

Como sugerido em Last, Bunke e Kandel (2002), para avaliar sistema de classificação com multiestágios pode-se utilizar como alternativa para estimar o custo de classificação uma medida chamada Número Total de Características (TFV, *Total-number of Feature-Values*), que é dada por

$$TFV = \sum_{i=1}^{n} m_i x_i \tag{2.10}$$

onde n é a quantidade de classificadores utilizados, m é a quantidade de características utilizadas pelo classificador i, e x é a quantidade de amostras processadas pelo classificador i.

2.6 Trabalhos relacionados

Um dos primeiros trabalhos a apresentar um modelo de classificação em cascata, ou multiestágio, foi Pudil et al. (1992). A ideia do sistema proposto é que a cada estágio de classificação, mais informações sejam adicionadas para aprimorar o sistema de classificação. O primeiro estágio, utilizaria menos informações, porém tem um custo menor, e a cada estágio novas informações são adicionadas, aumentando o custo do sistema. Todos estágios possuem rejeição, exceto o último. A Figura 2.10 ilustra o método proposto.

Para simplificar o modelo, foi considerado um sistema de reconhecimento de padrões de dois estágios, embora os conceitos apresentados podem ser estendidos para qualquer quantidade de estágios sem dificuldades. Considerou-se também um problema de reconhecimento de duas classes.

Uma análise matemática de redução de risco de decisão é apresentada. No modelo proposto, no primeiro estágio os padrões são classificados usando uma regra de decisão

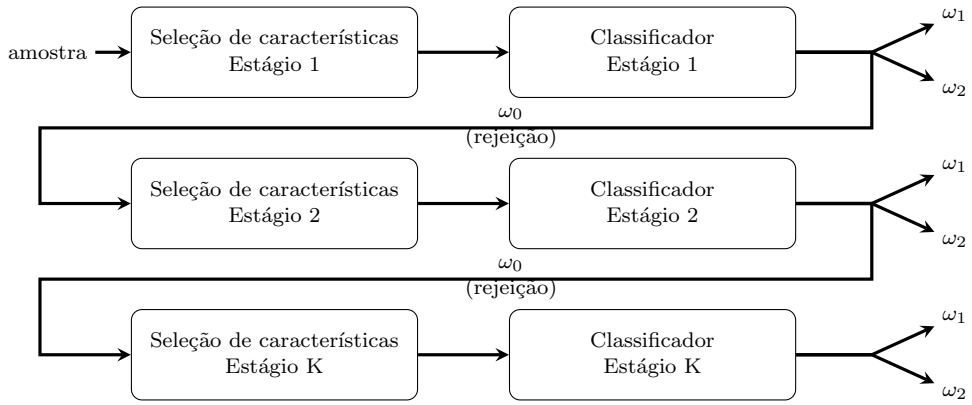

Figura 2.10: Esquema de classificação multiestágio proposto por Pudil et al. (1992)

de Bayes com rejeição $d^{(R1)}$ baseado em um vetor de características x_1. Se o padrão for rejeitado, então um segundo classificador atribui uma das duas classes usando uma regra de decisão de Bayes $d^{(2)}$ baseado em vetor de características x_2. A conclusão obtida, representada na desigualdade da Equação 2.11, é que o sistema de reconhecimento de padrões de dois estágios ($d^{(II)}$) possui um risco médio decisão inferior ao padrão de um único classificador.

$$R(d^{(II)}) < R(d^{(1)}) \tag{2.11}$$

O princípio proposto por Pudil et al. (1992) pode ser explorado mesmo nos casos em que todas os atributos estão incialmente disponíveis para um classificador de fase única. Pode-se considerar dividir o conjunto de atributos original em alguns subconjuntos de baixo custo e um subconjunto de custo mais alto (mais informativo). Utiliza-se o último subconjunto apenas para os padrões rejeitados durante a classificação nas primeiras fases. Obviamente que não existe apenas uma forma de separar o conjunto original de características entre os estágios, assim como não é trivial determinar o número de estágios necessários. O objetivo final é minimizar o risco médio da decisão para uma tarefa específica.

O trabalho de Alpaydin e Kaynak (1998) defende que uma grande porcentagem dos casos de treinamento em muitas aplicações podem ser explicadas com uma simples regra, com um número pequeno de exceções. Pensando nisso, um método de reconhecimento multiestágio é proposto. O esquema é composto de um modelo paramétrico linear e um classificador não paramétrico K-Vizinhos Mais Próximos (KNN, *K-Nearest Neighbors*). Basicamente o modelo linear aprende a "regra" e o KNN aprende as "exceções"

rejeitadas pela "regra". Como o modelo linear classifica uma grande porcentagem dos exemplos usando uma regra simples, somente um pequeno subconjunto de treinamento é armazenado como exceções durante o treinamento. Da mesma forma, durante os testes muitos padrões são classificados pelo modelo linear e poucos pelo KNN, causando assim somente um pequeno acréscimo na memória e processamento. O KNN é lento para encontrar os k vizinhos mais próximos, mas ele só será usado nos casos rejeitados pelo modelo linear, e quando usado, terá que procurar os vizinhos em um pequeno conjunto de dados. Para avaliar o desempenho do método utilizou-se uma base de dados de dígitos manuscritos (NIST). O valor de k do classificador KNN foi definido com 3 e para o limiar de rejeição foram testados os valores {0.70,0.80,0.90,0.95,0.99,1.00}. Os resultados analisados mostram que em média foi armazenado 7% dos dados de treinamento e apenas 18% do conjunto de testes usou o KNN. Alpaydin e Kaynak (1998) avalia que o método de cascata é uma abordagem melhor do que combinação de classificadores onde todos classificadores são usados para todos os casos. A definição do valor do limiar de rejeição, segundo o autor, é uma escolha entre velocidade e precisão. Para casos onde uma alta taxa de precisão é requerida, o limiar de rejeição deverá ser alto e utilizará mais o segundo estágio, ficando mais lento e usando mais memória.

Em Alimoglu e Alpaydin (2001) é feita uma investigação de técnicas para combinar múltiplas representação de dígitos manuscritos para aumentar a precisão de classificação sem elevar significativamente a complexidade do sistema ou tempo de reconhecimento. Os experimentos foram realizados em uma base de dígitos manuscritos com duas representações diferentes de cada dígito. Uma representação é dinâmica, que é o movimento da caneta de como o dígito é escrito em um *tablet*. A outra representação é estática, que é uma imagem gerada como resultado do movimento da caneta. Duas redes neurais Perceptron de Múltiplas Camadas (MLP, *Multilayer Perceptron*) foram utilizadas nos testes preliminares, uma para cada representação, e verificou-se que elas cometem erros de classificação em diferentes padrões, e que portanto, uma combinação adequada dos dois classificadores pode elevar a precisão. Para realizar a combinação dos classificadores foram avaliados os métodos de votação, mistura de especialistas, empilhamento e cascata.

Os resultados obtidos em Alimoglu e Alpaydin (2001) mostram que a combinação de classificadores foi capaz de melhorar a precisão. O método de cascata, seguindo a abordagem de Alpaydin e Kaynak (1998), se destacou por que além de melhorar o

desempenho, utiliza o segundo classificador, que é mais complexo, apenas em uma pequena porcentagem dos casos de teste. Em 70% dos casos de teste não foi necessário considerar as imagens da segunda representação.

Em Last, Bunke e Kandel (2002) a abordagem em cascata foi adotada com preocupação em melhorar a eficiência computacional através da combinação serial de classificadores usando reavaliação. É proposto então um método de classificação de dois estágios utilizando classificador KNN. O primeiro classificador é treinado com um subconjunto de características selecionado por um algoritmo específico. Já o segundo classificador é treinado com todas as características disponíveis. Na análise dos resultados o método se mostrou capaz de reduzir em até 50% o esforço computacional mantendo a acurácia no mesmo nível de um classificador um único estágio, ou até melhorando esta métrica.

Em Fumera, Pillai e Roli (2004) se investiga a utilidade da rejeição em sistemas de classificação de texto. A rejeição é a possibilidade de um classificador de texto reter a decisão da classificação caso considere que não seja suficientemente confiável. A princípio, essas rejeições deveriam ser tratadas manualmente. Para resolver automaticamente essas rejeições, é proposto uma arquitetura de classificador em duas fases, onde os documentos rejeitados na primeira fase são automaticamente classificados na segunda fase, de modo que não haja mais rejeição. Todos os estágios, exceto o último, podem classificar ou rejeitar um padrão. Os padrões rejeitados são encaminhados para o próximo estágio. O modelo de cascata proposto utiliza um classificador "global" e rápido - rede neural MLP ou NB - no primeiro estágio e um classificador "local" e lento - KNN - no segundo estágio. O desempenho do método é avaliado em uma tarefa de classificação de texto real, usando a base de dados Reuters.

Analisando os resultados obtidos em Fumera, Pillai e Roli (2004), o método de cascata em alguns casos supera o classificador KNN sem rejeição. Além disso, os resultados mostram que um classificador KNN no segundo estágio treinado com a mesma base utilizada no primeiro estágio tem desempenho inferior do que o KNN treinado com as rejeições do primeiro estágio.

Em Oliveira, Britto Jr. e Sabourin (2005) um sistema de classificação em cascata é utilizado para avaliar um algoritmo de otimização de limiar de rejeição por classe, melhorando a relação entre rejeição e erro. Para avaliar o algoritmo, ele foi aplicado

para otimizar os limiares de um sistema de classificação em cascata para reconhecer dígitos manuscritos. Nos experimentos, o classificador base foi uma rede neural MLP, o erro máximo permitido foi de 2% e foram gerados cinco classificadores com diferentes quantidades de características.

Nos resultados de Oliveira, Britto Jr. e Sabourin (2005) foi observado uma redução de custo de classificação, usando TFV, de aproximadamente 75% e um ganho na taxa de reconhecimento de 0,2%. Embora o ganho não seja significante, deve-se considerar que a classificação em cascata diminuiu a complexidade mantendo o mesmo nível de desempenho do classificador base.

Em Nguyen, Nguyen e Pham (2013) é apresentado um sistema de classificação para Análise de Sentimento[1] utilizando a abordagem de cascata. Na primeira fase da cascata encontra-se um classificador Naive Bayes, enquanto na segunda fase tem um classificador Máquinas de Vetores de Suporte. Os experimentos foram realizados com uma base de dados com 2000 resenhas de filmes, positivas e negativas. São extraídos três conjuntos de características dos comentários da base de dados. O primeiro nível da cascata utiliza apenas um conjunto de características, enquanto o segundo nível utiliza todos os três conjuntos para o treinamento, sendo então um classificador mais caro, conforme sugerido por Pudil et al. (1992). A base de dados foi dividida em dez subconjuntos, cada um com 100 comentários de cada classe. Os experimentos foram realizados utilizando validação cruzada, com nove subconjuntos para treinamento e um para teste. Para os classificadores testados individualmente, sem rejeição, a taxa de reconhecimento foi de 70% e 87,7% para o NB e SVM, respectivamente. Já com a cascata, a taxa de reconhecimento foi de 87,75%, onde o primeiro nível classificou corretamente 11,8% e rejeitou 87,55% e, o segundo nível classificou corretamente 86,75% dos casos rejeitados. Comparando a taxa de reconhecimento da cascata com o classificador SVM, o ganho é insignificante, estatisticamente falando, porém há de se considerar a redução do conjunto de características utilizadas e a não necessidade do uso do SVM (classificador mais complexo) para 12,45% dos casos.

[1]Refere-se ao uso de processamento de linguagem natural, análise de texto e linguística computacional para identificar e extrair informações subjetiva em documentos fontes

2.7 Considerações

Neste capítulo foi apresentado uma revisão de métodos de classificação em reconhecimento de padrões, como os classificadores base e, a geração, seleção e fusão de classificadores nos SMCs. O objetivo foi descrever as principais características dos métodos que serão utilizados nos próximos capítulos.

Também foram apresentadas as métricas de complexidade que serão utilizadas na análise do método proposto, como as estimativas de complexidade das bases de dados e a estimativa de custo computacional associado a tarefa de classificação.

Por fim, foram apresentados os principais trabalhos da literatura relacionados à classificação em cascata. Foi observado que a utilização da abordagem em cascata tem diferentes motivações nos diversos trabalhos. Por exemplo, em Pudil et al. (1992) o foco do trabalho é a redução do risco de decisão. Já em Alpaydin e Kaynak (1998) o objetivo é diminuir o custo computacional com o uso de memória e processamento. E em Alimoglu e Alpaydin (2001) pretende-se melhorar a precisão sem aumentar significativamente a complexidade do sistema.

Em todos esses casos, o método de classificação em cascata foi capaz de atingir os objetivos esperados, mostrando ser uma abordagem eficiente e promissora. Outro ponto a se destacar é que não foram observados trabalhos que consideram a rejeição no último nível da cascata, assim como não foi detectado a utilização de sistemas de múltiplos classificadores compondo um nível da cascata.

Capítulo 3

Método Proposto

O método de classificação em cascata de dois níveis proposto neste trabalho é uma combinação de um classificador monolítico (C), no primeiro nível da cascata, com um Sistema de Múltiplos Classificadores (E) no segundo nível.

A Figura 3.1 mostra uma visão geral da fase operacional da metodologia proposta. Neste esquema, uma amostra de classe desconhecida é apresentada ao primeiro nível da cascata (C), descrito na Seção 3.1, que irá classificar essa nova amostra ou rejeitá-la. No caso de rejeição, essa amostra é encaminhada para um segundo sistema de classificação (E), mais robusto que o primeiro (C). Esse segundo nível, detalhado na Seção 3.2, poderá por sua vez classificar a amostra ou também rejeitá-la.

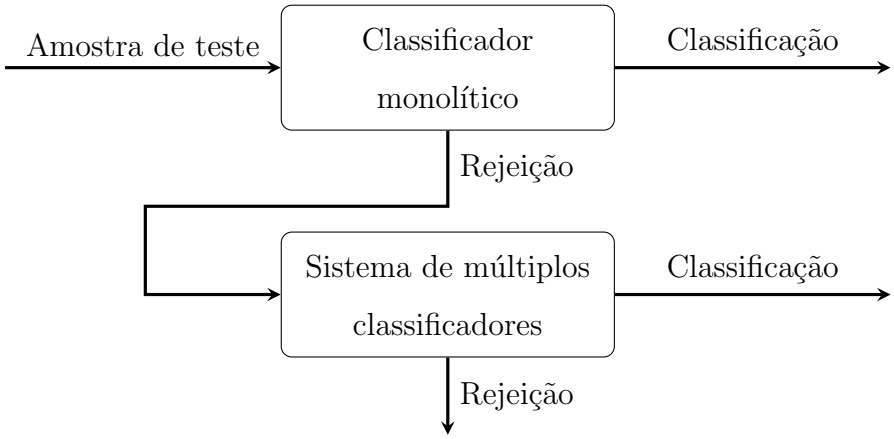

Figura 3.1: Visão geral da abordagem em cascata de dois níveis

O custo de um erro normalmente é mais elevado do que o custo de uma rejeição (CHOW, 1970). Dessa forma, com o objetivo de ter uma classificação mais confiável e

minimizar a taxa de erros, o resultado final do sistema pode ser uma rejeição da amostra, o que significa que o sistema de classificação não foi capaz de atribuir uma classe a essa nova amostra, respeitando a confiança mínima estabelecida pelo limiar de rejeição.

A fase de treinamento dos classificadores da cascata é ilustrada na Figura 3.2. Todas as amostras do conjunto de treinamento são utilizadas para treinar os classificadores dos dois níveis da cascata. No segundo nível, são utilizadas diferentes técnicas para geração de subconjuntos de dados (Seção 3.2.1) que servirão para treinar os classificadores que irão compor o conjunto E.

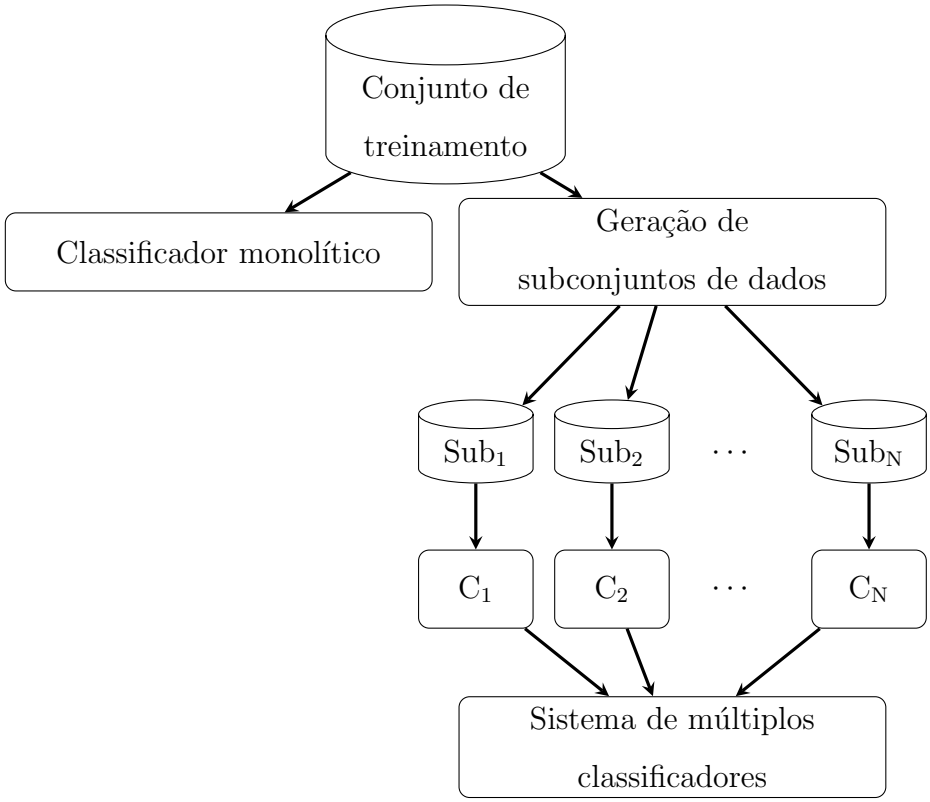

Figura 3.2: Fase de treinamento da abordagem em cascata

Para avaliar o desempenho dos métodos de classificação, cada amostra dos conjuntos de testes será submetida aos classificadores e será calculado então, a taxa de acertos (TA), a taxa de erros (TE) e a taxa de rejeições (TR), definidas como

$$TA = \frac{N_{\text{acertos}}}{N_{\text{testes}}} \times 100 \tag{3.1}$$

$$TE = \frac{N_{\text{erros}}}{N_{\text{testes}}} \times 100 \tag{3.2}$$

$$TR = \frac{N_{\text{rejeições}}}{N_{\text{testes}}} \times 100 \tag{3.3}$$

onde N_{testes} é a quantidade total de amostras de um conjunto de testes, N_{acertos} é a quantidade de amostras classificadas corretamente, N_{erros} é a quantidade de amostras classificadas erroneamente e $N_{\text{rejeições}}$ é a quantidade de amostras rejeitadas após a classificação. Dado que

$$N_{\text{testes}} = N_{\text{acertos}} + N_{\text{erros}} + N_{\text{rejeições}} \tag{3.4}$$

então,

$$TA + TE + TR = 100\% \tag{3.5}$$

ou seja, as taxas são complementares. Para o segundo nível (E) da cascata, o conjunto de testes é formado pelas amostras rejeitadas pelo primeiro nível (C), logo $N_{\text{testes},E} = N_{\text{rejeições},C}$.

3.1 Primeiro nível da cascata

O primeiro nível do método de classificação em cascata proposto possui apenas um classificador monolítico. Ele será responsável por resolver os casos de classificação mais fáceis, ou seja, os mais intuitivos, e rejeitar os casos que considerar difíceis, aqueles que precisarão ser submetidos a um método de reconhecimento mais robusto.

A definição do algoritmo base de classificação será realizada a partir de experimentos realizados nas bases de dados. Os cinco algoritmos descritos na fundamentação teórica na Seção 2.1 (KNN, MLP, NB, C4.5 e SVM) terão seus desempenhos avaliados, sem rejeição. Aquele que apresentar maior acurácia média para os problemas testados será selecionado como classificador monolítico do método cascata para os próximos experimentos.

Para utilizar alguns classificadores é necessário definir alguns parâmetros que influenciam diretamente no resultado da classificação. Para utilizar o algoritmo KNN é necessário primeiramente definir o número de vizinhos mais próximos (K). Para otimizar este valor, serão testadas variações com $1 \leq K \leq 30$, com incremento de 1 e com validação cruzada de quatro subconjuntos. Obtido os resultados, será selecionado o valor de K que

maximize a taxa de acertos para cada base de dados.

No MLP serão otimizados dois parâmetros principais, a taxa de aprendizagem (L), que controla os ajustes dos pesos durante a fase de treinamento, e a quantidade de neurônios na camada oculta (H) (DEPEURSINGE et al., 2010). Para o parâmetro L serão testados os valores $\{0{,}3; 0{,}6; 0{,}9\}$ e para H serão calculados quatro valores utilizando como parâmetro as características de cada problema de classificação, são eles: (n^o *de atributos*+ n^o *de classes*)/2, n^o *de atributos*, n^o *de classes* e n^o *de atributos* + n^o *de classes*.

Dois principais parâmetros influenciam no desempenho da árvore de decisão C4.5, o número mínimo de instâncias por folha (M), que determina o tamanho da árvore, e o limite de confiança (P) utilizado para a poda da árvore, que consiste na retirada de ramos com pouco ou nenhum ganho em precisão (DEPEURSINGE et al., 2010). Os valores testados serão $M = \{1; 2; 3; 4; 5\}$ e $P = \{0{,}05; 0{,}10; 0{,}15; 0{,}20; 0{,}25; 0{,}30\}$.

Já no algoritmo SVM (BURGES, 1998), será utilizada a função kernel RBF proposta por Hsu, Chang e Lin (2010). Esta função é dependente de dois parâmetros específicos, *Cost* (C) e *Gamma* (γ). Para otimizar estes parâmetros, Hsu, Chang e Lin (2010) recomendam a utilização da busca em grade *grid search* com validação cruzada. Várias combinações de (C, γ) serão testadas e aquela com maior precisão será escolhida. Para C assume-se os valores $\{1{,}2{,}3, \cdots, 15{,}16\}$ e para γ, $\{10^{-5}{,}10^{-4}{,}10^{-3}{,}10^{-2}{,}10^{-1}{,}10^{0}{,}10^{1}{,}10^{2}\}$, totalizando 128 diferentes combinações de parâmetros.

Já o classificador Naive Bayes não necessita de calibração (DEPEURSINGE et al., 2010).

3.1.1 Limiar de rejeição

Um dos pontos principais a ser considerado na classificação do primeiro nível é a definição do limiar de rejeição. Quando a probabilidade de um exemplo ser classificado corretamente pelo monolítico for menor que este limite, o exemplo deve ser rejeitado, ou seja, encaminhado para o segundo nível da cascata.

De acordo com a regra de rejeição de Chow (1970), para um problema de N classes uma amostra x é considerada aceita e rotulada com a classe ω_i se

$$\max_{k=1,\cdots,N} P(\omega_k|x) = P(\omega_i|x) \geq T, \tag{3.6}$$

onde $T \in [0,1]$. Por outro lado, a amostra x é rejeitada se

$$\max_{k=1,\cdots,N} P(\omega_k|x) = P(\omega_i|x) < T. \tag{3.7}$$

Este limiar T, $0 \leq T \leq 1$ é responsável por separar as amostras fáceis das difíceis. Se $T = 0$, nenhuma amostra será rejeitada, enquanto se $T = 1$, todas as amostras serão rejeitadas. Quanto maior o limite, maior também será o índice de amostras rejeitadas e menor deverá ser a taxa de erros, porém, em alguns casos que seriam corretamente classificados podem ser transformados em rejeição (CHOW, 1970), diminuindo a taxa de acertos.

Para os experimentos deste trabalho, a tolerância de erro considerada será $\leq 1\%$. O objetivo da otimização do limiar de rejeição é encontrar o valor mais adequado que maximize a taxa de acertos mantendo a tolerância de erro. Para isso, as probabilidades atribuidas as saídas dos classificadores serão utilizadas como guia para estabelecer um limiar de rejeição.

O impacto da inclusão da tolerância de erro pode ser observado no exemplo na Figura 3.3. A marcação destacada em 1% da taxa de erros indica a tolerância desejada. Quando não há rejeição, a taxa de erros é de aproximadamente 21,5%, logo a taxa de acertos é de 78,5%. Para que a taxa de erros seja $\leq 1\%$, neste caso, o sistema tem que rejeitar mais de 93% dos testes.

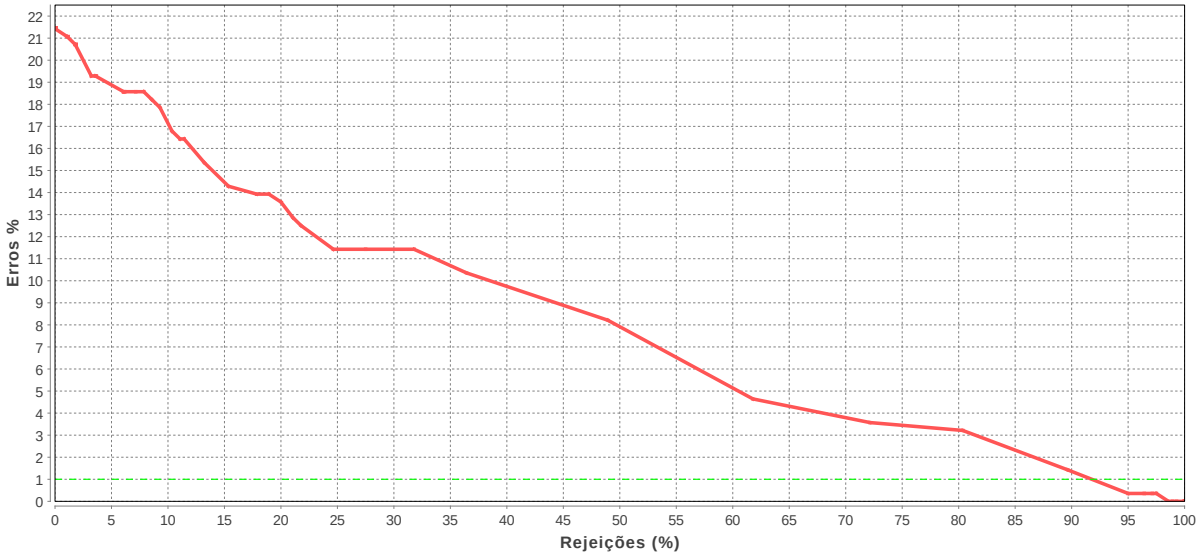

Figura 3.3: Gráfico de exemplo da relação entre a taxa de rejeições e a taxa de erros

A busca pelo limiar de rejeição será efetuada utilizando validação cruzada para cada uma das bases de dados. Dentro de um laço de repetição, os valores do limiar irão variar de 100% a 0%, com decremento de 1%. A cada passo, será mensurada a probabilidade de acerto para cada amostra do subconjunto de validação. Esse valor deverá ser comparado ao limiar de rejeição. Caso seja inferior, a amostra será rejeitada, caso contrário, o resultado da classificação será confrontado com a classe original da amostra, contabilizando como acerto ou erro. Quando a taxa de erros ultrapassar a tolerância definida, interrompe-se a busca e assume-se o valor do limiar de rejeição.

O reflexo da variação do valor limiar de rejeição T na taxa de erros e na taxa de rejeição pode ser visto no gráfico de exemplo da Figura 3.4. No exemplo, quando não há rejeição, o erro é de 12%. Para obter uma taxa de erros $\leq 1\%$, a rejeição será de pelo menos 35%.

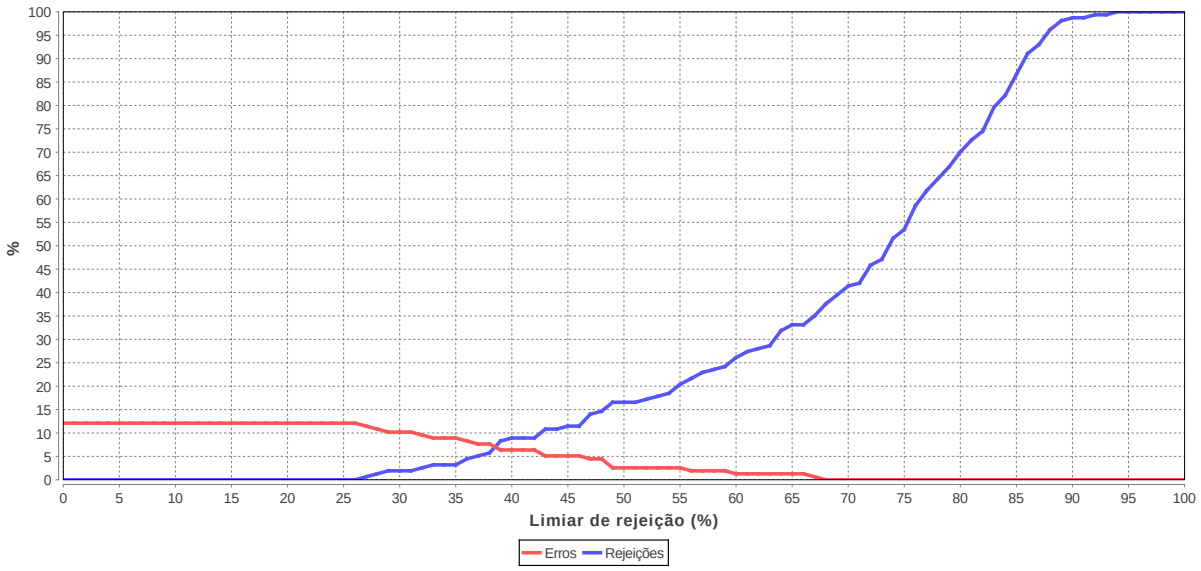

Figura 3.4: Gráfico de exemplo do comportamento das taxas de rejeições e de erros com a variação do limiar de rejeição

Um dos benefícios da rejeição é o possível ganho de confiança na classificação. Na próxima subseção será apresentado como estimar essa medida.

3.1.2 Estimativa de confiança na classificação

Em reconhecimento de padrões é improvável afirmar que um sistema de classificação possa fornecer 100% de acurácia. Muitos resultados publicados na literatura

apresentam o desempenho de seus métodos com nível zero de rejeição, ou seja, não consideram os possíveis casos de rejeição em um sistema de reconhecimento. Em problemas reais, geralmente a confiança no resultado é mais importante do que a taxa de reconhecimento propriamente dita (NIU; SUEN, 2012).

Além da taxa de reconhecimento, pode-se utilizar a confiabilidade (REL, *Reliability*) como uma medida de avaliação de um sistema de classificação. A Equação 3.8 define o cálculo da confiabilidade de um classificador relacionando a taxa de reconhecimento com a taxa de erros.

$$REL = \frac{TA}{(TA + TE)} \tag{3.8}$$

Em sistemas que não consideram a rejeição, os resultados se resumem em acertos e erros. Logo, $TA + TE = 100\%$. Substituindo os termos na Equação 3.8, obtém-se que $REL = TA$. Ou seja, quando não há rejeição, a confiança de uma classificação é sua própria taxa de reconhecimento.

Aplicando a Equação 3.8 no exemplo da Figura 3.3, no instante em que a taxa de erros passa a ser aceitável (taxa de erro = 1% e taxa de rejeição = 93%), obtém-se $REL = 85{,}7$. Pode-se concluir que no exemplo há um ganho de confiabilidade de 7,2 pontos percentuais, quando comparado a taxa de acertos sem rejeição, com um redução da taxa de erros de 21,5% para 1%. E ainda, na metodologia proposta, os 93% dos casos rejeitados no exemplo poderão ser recuperados pelo segundo nível da cascata, assunto da próxima seção.

3.2 Segundo nível da cascata

O segundo nível é projetado para processar apenas os casos rejeitados no primeiro nível. Ao configurar adequadamente a rejeição limiar do classificador C, deixa-se para E apenas os casos difíceis para os quais é necessário um esquema de classificação mais sofisticado. O conjunto de classificadores em E será gerado utilizando as mesmas amostras de treinamento usados para treinar os classificadores monolíticos utilizados no primeiro nível.

3.2.1 Geração de conjuntos de classificadores

Como bases para gerar conjuntos de classificadores, considerou-se os algoritmos KNN e SVM. A motivação é que o primeiro é normalmente escolhido quando se quer gerar um conjunto de classificadores fracos, e o segundo é geralmente escolhido quando se precisa de um classificador robusto. Dessa forma, é possível obter uma diversidade de classificadores para serem avaliados na cascata.

Os conjuntos de classificadores E serão geradas a partir de três diferentes técnicas de aprendizagem capaz de gerar diversos classificadores baseado em diferentes subconjuntos de dados: *Bagging*, *Boosting* e *Random Subspace Selection*.

Além disso, será avaliado também a técnica *Boosting* inicializada com instâncias de diferentes pesos (*BoostingW*). Nesta variação, as amostras rejeitadas ou mal classificadas pela primeira etapa receberão um peso maior do que aquelas que foram classificadas corretamente. A lógica por trás disso é aumentar a probabilidade das instâncias rejeitadas pelo primeiro nível serem selecionadas no treinamento dos conjuntos de classificadores.

Para cada uma das técnicas, um conjunto com 10 classificadores deverá ser gerado para cada problema. Um algoritmo de busca deverá ser utilizado para otimizar a cardinalidade da técnica RSS, que é o número de características selecionadas em cada subgrupo. A busca será feita com testes exaustivos variando o número de atributos entre 1 e a quantidade máxima de atributos disponíveis.

3.2.2 Sistemas de múltiplos classificadores

Duas abordagens diferentes serão avaliadas no segundo nível do método de classificação em cascata proposto. Em primeiro lugar o uso de conjuntos de classificadores sem qualquer mecanismo de seleção, em que todos os classificadores disponíveis no conjunto gerado são combinados usando a regra da votação por maioria. Em segundo lugar, o uso de métodos de seleção dinâmica de classificadores. Em ambos os casos, serão utilizados os mesmos conjuntos de classificadores.

Os métodos de seleção dinâmica de classificadores que serão analisados nos experimentos são: DSC-LA OLA e DSC-LA LCA, propostos por Woods, Kegelmeyer Jr. e Bowyer (1997); e, KNORA-Elimine e KNORA-Union, propostos por Ko, Sabourin e Britto Jr. (2008).

Assim como foi feito para o classificador monolítico no primeiro nível, o limiar de rejeição será definido para os sistemas de múltiplos classificadores considerando uma taxa de erro $\leq 1\%$ em um conjunto de validação.

Durante a fase operacional, uma amostra desconhecida é enviada para o classificador C. Espera-se que os casos fáceis sejam classificados no primeiro nível do método, enquanto os difíceis sejam rejeitados. Quando rejeitados, os casos são submetidos ao segundo nível E, onde poderá ser novamente rejeitada. O limiar de rejeição do segundo nível é otimizado também para uma taxa de erro $\leq 1\%$ em um conjunto de dados de validação.

3.3 Métricas de avaliação de desempenho

Para avaliar o desempenho do método proposto em comparação aos sistemas de múltiplos classificadores serão utilizadas três métricas. A primeira é a mais utilizada na literatura, a acurácia, que é mensurada pela taxa de acertos definida na Equação 3.1. Porém, para a classificação em cascata o cálculo deve ser adaptado para considerar as rejeições do primeiro nível (C) submetidas para o segundo nível (E). Assim, a taxa de acertos final é calculada de acordo com a Equação 3.9.

$$TA^* = TA_C + (TA_E \times TR_C) \tag{3.9}$$

Além da acurácia, a segunda métrica calcula a confiança na classificação, conforme Seção 3.1.2. A confiabilidade está relacionada a capacidade de um sistema de reconhecimento não aceitar uma classificação falsa e não rejeitar uma classificação correta.

Por fim, a terceira métrica para avaliar o desempenho é a redução de custo de classificação, apresentado na Seção 2.5.2. Para obter a estimativa de redução de custo (RC), calcula-se a relação entre o TFV (Equação 2.10) do método em cascata e o TFV do sistema de múltiplos classificadores executado isolado da cascata, conforme Equação 3.10.

$$RC = \left(1 - \frac{TFV_{\text{Cascata}}}{TFV_{\text{SMC}}}\right) \times 100 \tag{3.10}$$

Capítulo 4

Experimentos e Resultados

Esta seção apresenta os experimentos realizados para avaliar o método de classificação em cascata proposto. O principal objetivo é investigar se a abordagem em cascata pode reduzir os esforços necessários para a tarefa de classificação, quando comparado aos SMCs, enquanto mantém uma acurácia similar.

Começa-se apresentando as bases de dados utilizadas nos experimentos descrevendo suas principais características, dando uma noção sobre seu nível de dificuldade. Em seguida os experimentos são divididos em três conjuntos.

No primeiro conjunto de experimentos, comparou-se o desempenho dos classificadores monolíticos para o primeiro nível da abordagem em cascata. Os testes foram realizados de duas formas: sem rejeição e com rejeição. O objetivo é selecionar o classificador mais promissor para assumir o primeiro nível da cascata e mostrar o impacto da rejeição em um mecanismo com meta de taxa de erro $\leq 1\%$.

O segundo conjunto de experimentos avalia dois modelos de sistemas de múltiplos classificadores para compor o segundo nível da cascata utilizando um *ensemble*: a combinação de todos os classificadores disponíveis; e, quatro métodos baseado na seleção dinâmica de classificadores.

E por fim, no último conjunto de experimentos, avalia-se a abordagem em cascata considerando diferentes configurações possíveis. Para realizar esta avaliação, considerou-se além da taxa de reconhecimento, a redução de custo em termos de classificação, que é estimada usando o TFV, e a confiança (REL) dos métodos de classificação.

4.1 Bases de dados utilizadas

Para validar a metodologia proposta e proporcionar futuras comparações com outros métodos de classificação, testaram-se as abordagens em 12 diferentes problemas de reconhecimento de padrão. A maioria das bases de dados foram extraídas de um repositório de dados de aprendizagem de máquina, o UCI[1] (LICHMAN, 2013), com exceção da base Forest Species (FS)[2] apresentada por Martins et al. (2013).

As bases de dados utilizadas são apresentadas e descritas no Quadro 4.1. As amostras dessas bases foram separadas em conjuntos de treinamento e de teste. Em geral, adotou-se a proporção de 50% para cada uma das finalidades, exceto para as bases de dados IS e FS, onde os conjuntos de treinamento e de teste foram previamente recomendados pelos autores das bases. Para os métodos de classificação que necessitam de validação, separou-se 70% do conjunto de treinamento para realizar o treinamento do classificador e os 30% restantes para realizar a validação.

A Tabela 4.1 apresenta as principais características de cada base de dados, como a quantidade de classes e atributos, a distribuição das amostras em conjunto de treinamento e de teste, e o nível de desbalanceamento. Uma base é dita balanceada quando todas as classes possuem a mesma quantidade de amostras. O nível de desbalanceamento entre as classes é também um indicador de complexidade da base, ele pode ser medido pela razão entre o número de instâncias da classe majoritária e o número da classe minoritária.

[1]Repositório disponível em: http://archive.ics.uci.edu/ml/datasets.html

[2]Base de dados disponível em: http://web.inf.ufpr.br/vri/image-and-videos-databases/forest-species-database

Quadro 4.1: Apresentação das bases de dados utilizadas

Base de dados	**Descrição**
Liver Disorders (LD)	Utiliza dados de testes de sangue que são pensados para ser sensível a doenças do fígado que podem surgir a partir do consumo excessivo de álcool. Cada amostra constitui o registro de um único indivíduo do sexo masculino. Possui atributos contínuos e discretos.
Haberman (HB)	O conjunto de dados contém casos de um estudo sobre a sobrevivência de pacientes que haviam sido submetidos a cirurgia para câncer de mama. Todos os atributos são discretos.
Blood (BD)	Utiliza dados da frequência de doação de sangue por doador extraídas aleatoriamente de um banco de dados de doadores. Todos os atributos são discretos.
Pima Indians Diabetes (PD)	Utiliza amostras de uma população de pacientes do sexo feminino, acima de 21 anos, para detectar sinais de diabetes. Possui atributos contínuos e discretos.
Vehicle (VE)	Utiliza um conjunto de características retirados da silhueta de quatro tipos de veículos. As medidas foram calculadas considerando diferentes ângulos de visão do carro. Todos os atributos são contínuos.
Sonar (SO)	O conjunto de dados contém sinais sonoros obtidos a partir de cilindros de metais e de rochas em condições semelhantes. Todos os atributos são contínuos.
Ionosphere (IO)	Utiliza antenas de alta frequência com alvo nos elétrons livres da ionosfera. Todos os atributos são contínuos.
Forest Species (FS)	Utiliza dados extraídos de imagens microscópicas de diferentes espécies florestais. Todos os atributos são discretos.
Wine (WI)	Utiliza análise química para determinar a origem do vinho. Os dados são de vinhos produzidos na mesma região da Itália, mas derivadas de três cultivares diferentes. Todos os atributos são contínuos.
Wisconsin Breast-Cancer (WC)	Utiliza dados extraídos de uma imagem digitalizada que descrevem as características dos núcleos celulares da mama, caracterizando-os como benigno ou maligno. Possui atributos contínuos e discretos.
Image Segmentation (IS)	Utiliza características de imagens segmentadas a mão para criar uma classificação para cada pixel, rotulando como céu, janela, cimento, entre outras classes. Cada instância é uma região de 3x3. Possui atributos contínuos e discretos.
Iris (IR)	O conjunto de dados contém 50 casos de cada uma das três classes existentes, que se referem a um tipo de planta. Apenas uma das classes é linearmente separável das demais. Todos os atributos são contínuos.

Tabela 4.1: Descrição das bases de dados e separação das amostras

Base	Classes	Atributos	Treinamento	Teste	Desbalanceamento*
LD	2	6	172	173	1,38
HB	2	3	153	153	2,78
BD	2	4	374	374	3,20
PD	2	8	384	384	1,87
VE	4	18	423	423	1,10
SO	2	60	104	104	1,14
IO	2	34	175	176	1,79
FS	41	1352	11768	35304	2,68
WI	3	13	89	89	1,48
WC	2	30	284	285	1,68
IS	7	19	210	2100	1,00
IR	3	4	75	75	1,00

* Quanto maior o valor do indicador, maior é o desbalanceamento, sendo que o valor mínimo 1,0 indica que as classes estão balanceadas.

Para avaliar o nível de dificuldade que cada problema de classificação apresenta, foram usados nos experimentos três medidas de complexidade, descritas na Seção 2.5.1. De acordo com as definições em Ho e Basu (2002) e Sanchez, Mollineda e Sotoca (2007), foi selecionada uma medida da sobreposição entre valores de uma única característica (F1), uma medida de separabilidade não-paramétrica de classes (N2) e uma medida relacionada com a geometria e densidade de classes (N4). Todas as medidas foram calculadas usando a biblioteca em C++ *Data Complexity Library* (DCoL)[3] descrita em Orriols-Puig, Macia e Ho (2010). A Tabela 4.2 mostra os valores obtidos das três medidas de complexidade para cada uma das bases de dados.

[3]Software disponível em: http://dcol.sourceforge.net/

Tabela 4.2: Cálculo das medidas de complexidade das bases de dados

Base	**F1**	**N2**	**N4**
LD	0,055	0,91	0,342
HB	0,189	0,76	0,364
BD	0,298	0,63	0,396
PD	0,577	0,84	0,274
VE	0,451	0,62	0,299
SO	0,466	0,74	0,094
IO	0,614	0,63	0,159
FS	1,854	0,79	0,103
WI	3,831	0,54	0,131
WC	3,405	0,56	0,013
IS	8,809	0,05	0,111
IR	7,097	0,13	0,081

Para efeito comparativo da complexidade entre conjuntos de dados, pode-se analisar sua Posição Média (MR, *Mean Rank*) para as medidas F1, N2 e N4. A Equação 4.1 calcula a posição média para cada base de dados b usando o número de sua posição (R) nos resultados ordenados do cálculo de cada uma das medidas de complexidade c.

$$MR_b = \frac{\sum_{i=1}^{c} R_i^b}{c} \tag{4.1}$$

A Tabela 4.3 mostra o *ranking* das bases de dados para cada uma das medidas, assim como sua posição média MR que será considerada, para efeito comparativo, como a medida final de complexidade. Os registros da tabela estão ordenados da menor posição média (maior complexidade) para a maior posição (menor complexidade). Como se pode ver, a base de dados LD apresenta a maior complexidade, enquanto IR é o problema mais fácil em função das medidas estimadas.

Tabela 4.3: Cálculo da posição média na complexidade das bases de dados

Base	Posição F1	Posição N2	Posição N4	Posição Média (MR)
LD	1	1	3	1,7
HB	2	4	2	2,7
BD	3	6	1	3,3
PD	6	2	5	4,3
VE	4	8	4	5,3
SO	5	5	10	6,7
IO	7	7	6	6,7
FS	8	3	9	6,7
WI	10	10	7	9,0
WC	9	9	12	10,0
IS	12	12	8	10,7
IR	11	11	11	11,0

O indicador MR é baseada na ordenação dos resultados das medidas de complexidade, logo o resultado é um valor que simboliza a complexidade de uma base de dados em relação as demais. O MR ordena as bases de dados da mais complexa para a menos complexa. Assim, as bases que obtiveram as primeiras colocações na ordenação, menor valor de MR, são as que apresentam maior complexidade.

Espera-se observar que os problemas mais fáceis sejam classificados pelo primeiro nível da cascata proposta, enquanto os problemas difíceis sejam rejeitados e encaminhados para o segundo nível que possui um sistema de classificação mais robusto. De fato, o objetivo final não é apenas mostrar algum ganho em termos de precisão, mas reduzir o esforço de classificar os casos de teste sem perder precisão quando comparado a um sistema de múltiplos classificadores.

4.2 Primeiro nível da cascata - Classificador monolítico

Para definir o classificador monolítico que será utilizado no primeiro nível da cascata nos próximos experimentos, foram avaliados os classificadores KNN, MLP, C4.5, NB e SVM, a fim de selecionar aquele que demonstrasse melhor desempenho. O primeiro passo é a otimização dos algoritmos. A Tabela 4.4 mostra os melhores parâmetros de cada classificador para cada conjunto de dados. Para o classificador KNN, apresenta-se

o número de vizinhos (K), e para o MLP, a taxa de aprendizagem (L) e o número de neurônios na camada oculta (H). Já para o C4.5, mostra-se o mínimo de instâncias por folha (M) e o limite de confiança (P), enquanto para o SVM, os valores otimizados de *Cost* (C) e *Gamma* (γ). O classificador NB não necessita de otimização.

Tabela 4.4: Melhores parâmetros de treinamento para os classificadores monolíticos em cada conjunto de dados

Base	**KNN**	**MLP**		**C4.5**		**SVM**	
	K	**L**	**H**	**M**	**P**	γ	**C**
LD	9	0,3	6	0,25	2	1,00	14
HB	9	0,3	2	0,25	5	1,00	16
BD	10	0,3	2	0,05	2	10,00	8
PD	5	0,6	8	0,20	2	0,10	16
VE	1	0,3	4	0,15	1	1,00	16
SO	1	0,6	31	0,25	4	0,10	16
IO	5	0,3	34	0,30	5	1,00	16
FS	1	0,3	1393	0,20	5	0,01	16
WI	14	0,3	13	0,25	4	0,10	16
WC	8	0,9	16	0,20	1	1,00	16
IS	1	0,3	26	0,25	5	1,00	14
IR	3	0,3	4	0,25	2	0,10	16

A Tabela 4.5 contém os primeiros resultados dos experimentos com os desempenhos dos classificadores nas bases de testes, medidos pela taxa de acertos, sem considerar a rejeição. Nota-se que o algoritmo SVM teve desempenho igual ou superior em 10 das 12 bases do experimento, com reconhecimento médio de 85,01%. Dessa forma, o classificador monolítico SVM é selecionado para representar o primeiro nível da cascata.

Tabela 4.5: Resultado da taxa de acertos dos classificadores monolíticos sem rejeição

Base	KNN	MLP	C4.5	NB	SVM
LD	63,58	**69,36**	56,65	51,45	**69,36**
HB	**75,82**	**75,82**	73,86	**75,82**	75,16
BD	79,14	**79,41**	75,67	75,13	78,34
PD	73,44	74,22	67,71	76,30	**77,86**
VE	70,69	77,30	74,23	40,66	**79,67**
SO	84,62	78,85	70,19	60,58	**86,54**
IO	85,80	84,09	88,64	84,09	**94,89**
FS	57,82	43,20	35,05	40,48	**71,64**
WI	97,75	98,88	93,26	96,63	**100,00**
WC	97,89	97,19	94,04	93,68	**98,60**
IS	91,62	89,14	91,00	79,95	**92,10**
IR	**96,00**	94,67	94,67	**96,00**	**96,00**

Nota: os melhores resultados de reconhecimento para cada base de dados estão destacados com negrito.

A definição do limiar de rejeição do classificador monolítico é um ponto crucial para o bom desempenho do método proposto. Ele é responsável por separar os padrões fáceis, que são as amostras que o classificador monolítico tem capacidade de rotular corretamente, e os padrões difíceis, que são as amostras que necessitam de um sistema mais robusto. Além disso, a rejeição é adotada para estabelecer uma meta de tolerância de erro, de acordo com o problema envolvido, que neste caso é erro $\leq 1\%$.

A Figura 4.1 apresenta um gráfico com a relação entre a taxa de rejeição e a taxa de erro do classificador monolítico SVM para a base LD, que é o problema de maior complexidade do conjunto de experimentos, conforme relacionado anteriormente na Tabela 4.3. Os resultados foram obtidos usando validação cruzada no conjunto de treinamento. Observa-se que para reduzir a taxa de erros para 1% (valor destacado no gráfico), como consequência é rejeitado mais de 97% das amostras de teste. Quando o crescimento da rejeição tem proporção maior do que a redução da taxa de erros, significa que amostras que seriam classificadas corretamente estão sendo rejeitadas. Em um sistema ideal, ao incluir a opção de rejeição teria $TE = 0$, $TR = TE_{\text{sem rejeição}}$ e $TA_{\text{com rejeição}} = TA_{\text{sem rejeição}}$.

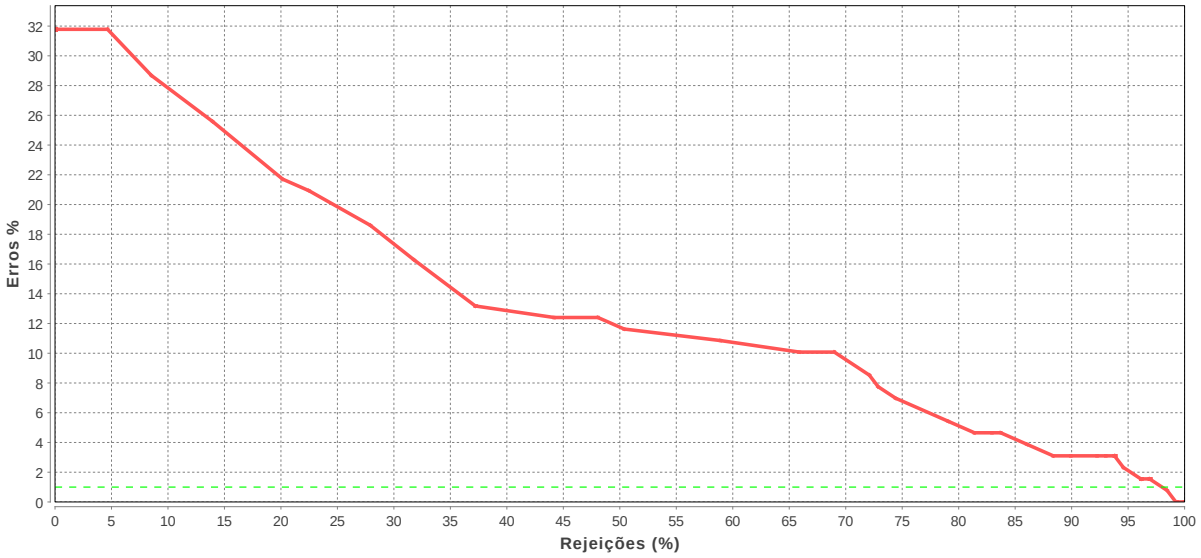

Figura 4.1: Gráfico da relação entre a taxa de rejeições e a taxa de erros para a base de maior complexidade (LD)

Em contrapartida, a Figura 4.2 apresenta o mesmo gráfico da figura anterior, mas agora para o problema de menor complexidade, a base IR. Neste caso, observa-se que o reflexo da redução da taxa de erros para 1%, impacta numa rejeição a partir de 7%, sendo assim, a taxa de acertos na fase de validação para o classificador monolítico seria de até 92%.

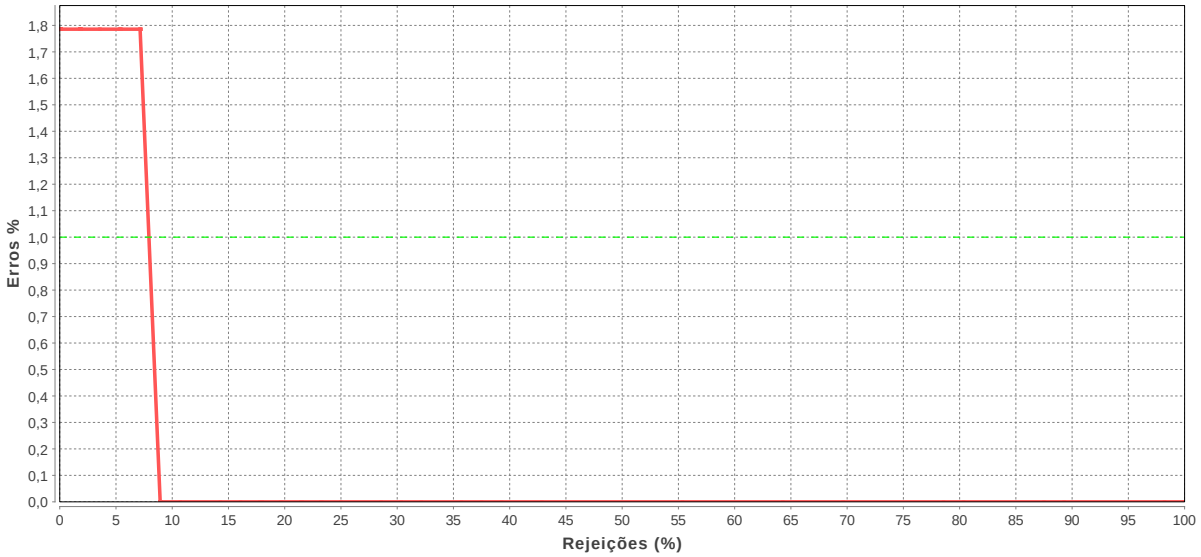

Figura 4.2: Gráfico da relação entre a taxa de rejeições e a taxa de erros para a base de menor complexidade (IR)

O limiar de rejeição estabelece a probabilidade mínima requerida na tarefa de classificação para aceitar ou recusar a decisão de um classificador ou de um sistema de múltiplos classificadores. Um limiar de rejeição inadequado poderá comprometer o desempenho do sistema. Se muito alto, amostras que seriam corretamente classificadas no primeiro nível da cascata serão rejeitadas, demandando uma reanálise no segundo nível, aumentando o custo da classificação. Do mesmo modo, para um limiar de rejeição muito baixo, amostras que deveriam ser rejeitadas pela baixa confiança na classificação serão aceitas pelo primeiro nível, podendo comprometer a meta da taxa de erros $\leq 1\%$.

A otimização do limiar de rejeição foi então realizada com foco em obter uma taxa de erro $\leq 1\%$ sobre a base de validação. Os limiares obtidos para o classificador monolítico selecionado no passo anterior, o SVM, foram obtidos para cada uma das bases de dados utilizando validação cruzada. Os resultados são apresentados na Tabela 4.6.

Tabela 4.6: Limiar de rejeição do classificador SVM

Base	**SVM**
LD	0,93
HB	0,83
BD	0,85
PD	0,93
VE	0,80
SO	0,79
IO	0,95
FS	0,73
WI	0,44
WC	0,81
IS	0,66
IR	0,62

Para analisar o impacto do limiar de rejeição no desempenho de um sistema de classificação, duas representações gráficas dos experimentos utilizados na definição do limiar de rejeição são apresentadas na Figura 4.3 para a base LD (maior complexidade) e na Figura 4.4 para a base IR (menor complexidade). Os gráficos obtidos para as demais bases estão disponíveis no Apêndice A.1.

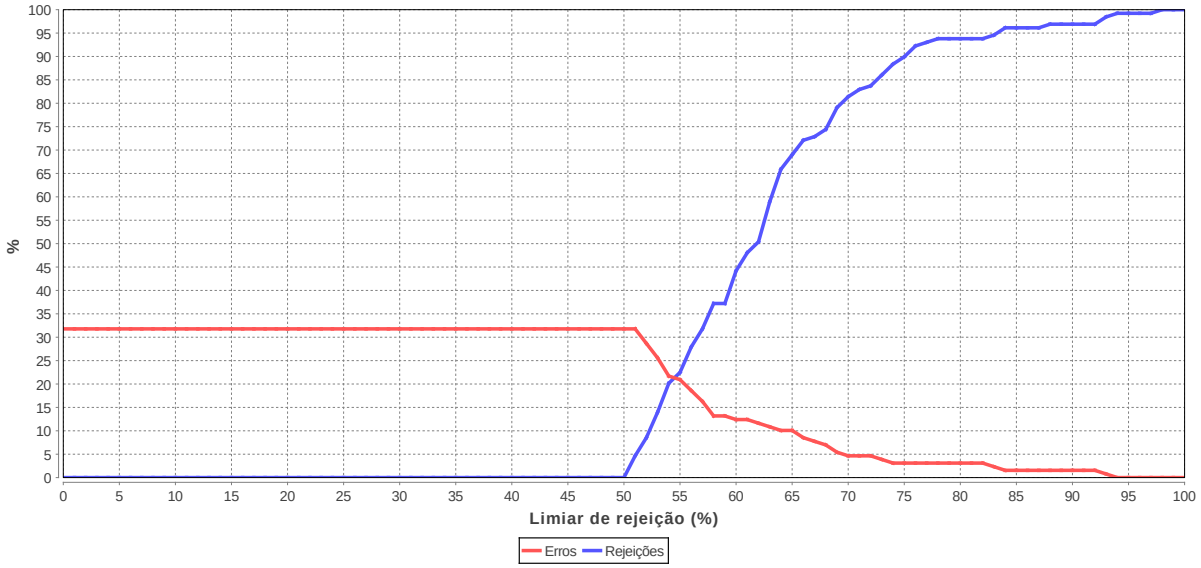

Figura 4.3: Gráfico da relação entre a definição do limiar de rejeição e as taxas de erros e de rejeições para a base de maior complexidade (LD)

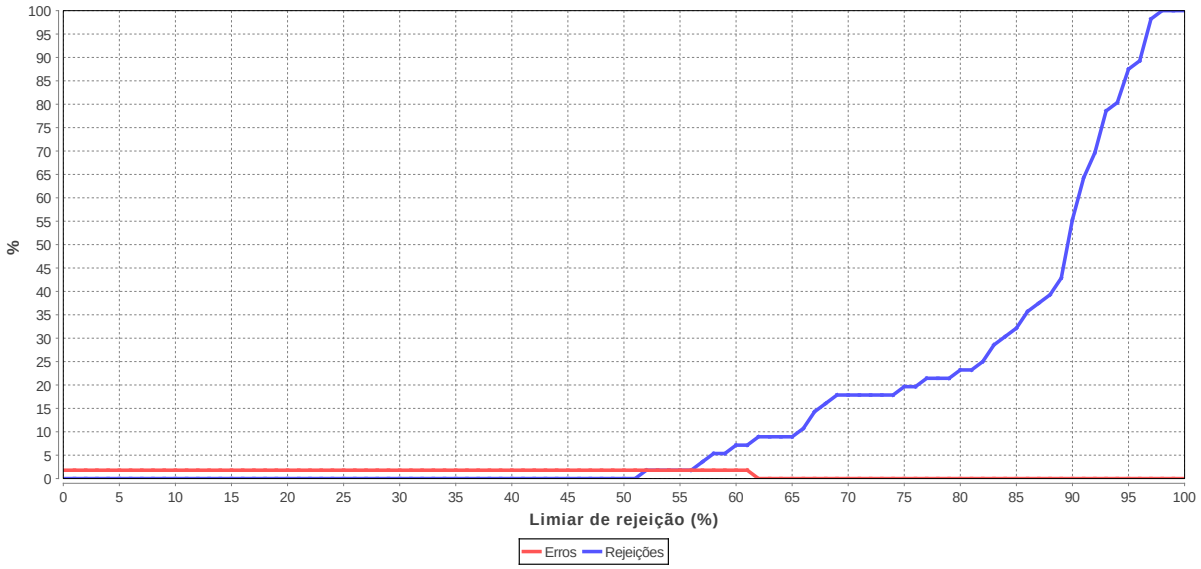

Figura 4.4: Gráfico da relação entre a definição do limiar de rejeição e as taxas de erros e de rejeições para a base de menor complexidade (IR)

Quando utiliza-se o classificador SVM considerando o mecanismo de rejeição, a porcentagem média de acertos no primeiro nível é de 49,74%, conforme Tabela 4.7, enquanto 48,59% das amostras são rejeitadas e serão processadas pelo próximo nível.

Tabela 4.7: Resultado da classificação usando SVM com e sem rejeição

Base	Sem rejeição		Com rejeição			
	acertos	**erros**	**acertos**	**erros**	**rejeição**	**REL**
LD	69,36	30,64	5,78	1,16	93,06	83,33
HB	75,16	24,84	9,15	1,31	89,54	87,50
BD	78,34	21,66	2,14	0,27	97,59	88,89
PD	77,86	22,14	8,59	0,52	90,89	94,29
VE	79,67	20,33	53,19	1,89	44,92	96,57
SO	86,54	13,46	46,15	4,81	49,04	90,57
IO	94,89	5,11	55,11	0,57	44,32	98,98
FS	71,64	28,36	49,09	5,87	45,05	89,33
WI	100,00	0,00	100,00	0,00	0,00	100,00
WC	98,60	1,40	93,33	0,70	5,96	99,25
IS	92,10	7,90	81,00	1,62	17,38	98,04
IR	96,00	4,00	93,33	1,33	5,33	98,59

Pode-se observar que o primeiro nível da cascata pode reconhecer muitos padrões, mesmo quando a rejeição é considerada. Este é o caso das bases IR, WI e WC, que são consideradas fáceis como mostrado na Tabela 4.3. Por outro lado, bases como LD, HB, BD e PD, que são consideradas complexas, apresentaram um alto número de rejeições para o classificador monolítico.

É possível observar que os problemas mais difíceis da Tabela 4.3 (LD, HB, BD, PD, por exemplo) tem uma alta taxa de rejeição na primeira etapa. Por outro lado, os problemas fáceis, como WI, WC e IR, demonstraram baixa taxa de rejeição. Tais observações já confirmam que uma quantidade significativa de casos pode ser resolvida pelo primeiro classificador do método de cascata proposto. No entanto, a questão é quantas instâncias rejeitadas poderão ser classificadas na segunda etapa. Este é o foco dos experimentos das próximas seções.

4.3 Segundo nível da cascata - Sistema de múltiplos classificadores

Neste ponto, a primeira tarefa era gerar um *pool* de classificadores diversos. Conforme descrito na metodologia foram utilizadas três diferentes técnicas de

aprendizagem capazes de gerar conjunto de classificadores: *Bagging*, *Boosting* e *Random Subspace Selection*. Além disso, uma variação da técnica *Boosting*, aqui chamada de *BoostingW*, foi utilizada adotando pesos diferenciados para as amostras rejeitadas ou mal classificadas.

Em cada técnica foram gerados *pools* de 10 classificadores para cada problema, com exceção para alguns problemas usando RSS em que o número de características não é suficiente para gerar 10 diferentes subconjuntos. Para *Bagging* e *Boosting*, 66% das amostras de treinamento foram usados para gerar os subconjuntos para treinamento de cada classificador. A otimização da cardinalidade (número de atributos a ser selecionado) do RSS, assim como a quantidade de classificadores gerados é mostrado na Tabela 4.8.

Tabela 4.8: Resultado da otimização da cardinalidade da técnica RSS

Base	**Cardinalidade subconjunto**	**Qtde. de classificadores**
LD	3	10
HB	2	3
BD	2	6
PD	4	10
VE	6	10
SO	12	10
IO	8	10
FS	100	10
WI	6	10
WC	5	10
IS	4	10
IR	2	6

Conforme descrito anteriormente, as diferentes variantes de duas abordagens possíveis foram avaliadas para a segunda etapa do método de cascata. Em primeiro lugar, o uso de *ensembles* sem qualquer mecanismo de seleção, em que todos os classificadores disponíveis no conjunto gerado são combinados usando a regra da votação majoritária. Em segundo lugar, o uso de métodos de seleção dinâmica de classificadores. Em ambos os casos, os mesmos *pools* de classificadores foram utilizados.

Tal como feito para o classificador monolítico do primeiro nível, o limiar de rejeição de cada SMCs foi definido considerando uma taxa de erro $\leq 1\%$ em um conjunto com

validação cruzada.

Na combinação de classificadores foram gerados seis *ensembles* a serem avaliados para cada problema. Para cada uma das três técnica, *Bagging*, *Boosting* e RSS, foram considerados dois classificadores base: KNN e SVM. A motivação da escolha é que o KNN é geralmente indicado para gerar um *pool* de classificadores fracos, enquanto o SVM é normalmente escolhido quando se necessita de um classificador mais robusto, conforme demonstrado nos resultados dos experimentos anteriores. Dessa forma, considerando as 12 bases de dados, foram realizados 72 experimentos com a combinação de classificadores.

A Tabela 4.9 resume os resultados de todos as combinações de conjuntos de classificadores avaliadas para o segundo nível da cascata. Ela apresenta os melhores desempenhos para cada problema. É importante notar que estes resultados não consideram o regime de cascata. No entanto, eles são importantes, pois permitirão uma comparação com o desempenho do método em cascata aqui proposto. Os resultados gerais obtidos nestes experimentos estão disponíveis no Apêndice A.2.

Tabela 4.9: Resumo dos melhores resultados obtidos para as combinações de conjuntos de classificadores

Base	**Técnica**	**Classifi.**	**Sem rejeição**		**Com rejeição**			
		base	**acertos**	**erros**	**acertos**	**erros**	**rejeição**	**REL**
LD	RSS	SVM	60,12	39,88	16,18	5,20	78,61	75,68
HB	Boosting	KNN	75,82	24,18	13,07	3,27	83,66	80,00
BD	RSS	SVM	76,20	23,80	35,56	4,01	60,43	89,86
PD	Bagging	SVM	78,39	21,61	27,86	2,08	70,05	93,04
VE	Bagging	SVM	79,20	20,80	53,90	2,13	43,97	96,20
SO	Boosting	KNN	84,62	15,38	84,62	15,38	0,00	84,62
IO	RSS	SVM	94,32	5,68	86,36	2,84	10,80	96,82
FS	Bagging	SVM	70,56	29,44	42,51	4,19	53,30	91,02
WI	Bagging	SVM	98,88	1,12	98,88	1,12	0,00	98,88
WC	Bagging	SVM	98,25	1,75	97,19	1,40	1,40	98,58
IS	Boosting	KNN	91,62	8,38	91,62	8,38	0,00	91,62
IR	Bagging	KNN	98,67	1,33	97,33	1,33	1,33	98,65

Outra opção avaliada para o segundo nível da abordagem em cascata é a utilização de métodos baseados na seleção dinâmica de classificadores. Foram considerados quatro

métodos de seleção dinâmica (DS-OLA, DS-LCA, KNORA-Eliminate, e KNORA-Union) apresentados na Seção 2.2.2.

Considerando as quatro técnicas de geração de subconjunto de classificadores (*Bagging*, *Boosting*, *BoostingW* e RSS) e os dois classificadores base (KNN e SVM), tem-se 32 possíveis configurações para serem avaliadas em cada base de dados, além dos experimentos de combinação já citados.

Vale salientar que o objetivo é mostrar que utilizando o esquema de cascata proposto, é possível poupar esforços na tarefa de classificação ao tratar os casos fáceis no primeiro nível da cascata. Não é o caso de comparar o desempenho de todas as configurações possíveis quando se utiliza o método seleção dinâmica. Com isto em mente algumas escolhas foram feitas para o conjunto de experimentos, como: o KNN foi selecionado como classificador base. Este método é geralmente a escolha na literatura para geração de conjunto de classificadores fracos; a técnica *Bagging* foi selecionada para gerar o subconjunto de 10 classificadores.

A Tabela 4.10 resume os resultados de todos os métodos de seleção dinâmica avaliados para o segundo nível. Ela mostra o melhor resultado observado para cada problema. É importante notar que estes resultados ainda não consideram a abordagem em cascata. No entanto, eles são importantes para serem comparados posteriormente com o desempenho da cascata. Como se pode ver, os métodos de seleção dinâmica superaram o classificador monolítico em 6 das 12 bases, enquanto que o desempenho geral foi pior do que a combinação de todos os classificadores. Os resultados gerais obtidos nestes experimentos estão disponíveis no Apêndice A.3.

Tabela 4.10: Resumo dos melhores resultados obtidos para os métodos de seleção dinâmica de classificadores

Base	Seleção	Sem rejeição		Com rejeição			
	dinâmica	acertos	erros	acertos	erros	rejeição	REL
LD	DS-OLA	58,96	41,04	13,29	7,51	79,19	63,89
HB	KNORA-U	73,20	26,80	10,46	1,96	87,58	84,21
BD	KNORA-E	75,67	24,33	10,96	1,07	87,97	91,11
PD	KNORA-U	70,83	29,17	17,19	1,04	81,77	94,29
VE	*	–	–	0,00	0,00	100,00	–
SO	DS-LCA	69,23	30,77	14,42	7,69	77,88	65,22
IO	*	–	–	0,00	0,00	100,00	–
FS	KNORA-U	57,14	42,86	16,55	0,91	82,54	94,79
WI	KNORA-U	96,63	3,37	95,51	1,12	3,37	98,84
WC	DS-OLA	96,49	3,51	94,39	1,40	4,21	98,53
IS	DS-LCA	86,71	13,29	13,57	4,00	82,43	77,24
IR	KNORA-U	97,33	2,67	97,33	2,67	0,00	97,33

Resultados marcados com "*" são padrão para todos os métodos.
Nestes casos não foram analisadas as classificações sem rejeição.

4.4 Classificação em cascata

Os melhores resultados obtidos com método de classificação em cascata proposto neste trabalho estão sumarizados na Tabela 4.11. Além das taxas de acertos, erros e rejeições, a tabela indica qual configuração de cascata foi responsável pelo melhor desempenho em cada base. A variação das configurações da cascata estão no segundo nível, que pode assumir diferentes SMCs avaliados nos experimentos. São apresentados então, os resultados individuais do primeiro e segundo nível da cascata, e o resultado final do método, medido pela acurácia (acertos) e confiabilidade (REL). É importante salientar que todos os resultados apresentados consideram a opção de rejeição. Os resultados gerais obtidos nestes experimentos estão disponíveis no Apêndice A.4.

Tabela 4.11: Resumo dos melhores resultados obtidos para os métodos propostos de classificação em cascata

Base	1º nível			2º nível						Cascata	
	acertos	erros	rejeição	Classif.	Técnica	Seleção	acertos	erros	rejeição	acertos	REL
LD	5,78	1,16	93,06	KNN	Bagging	DS-OLA	13,66	8,70	77,64	18,50	66,67
HB	9,15	1,31	89,54	KNN	Boosting	-	10,22	2,92	86,86	18,30	82,35
BD	2,14	0,27	97,59	SVM	RSS	-	35,34	3,84	60,82	36,63	90,13
PD	8,59	0,52	90,89	SVM	Bagging	-	21,20	1,72	77,08	27,86	93,04
VE	53,19	1,89	44,92	SVM	Boosting	-	14,21	10,53	75,26	59,57	90,00
SO	46,15	4,81	49,04	KNN	Boosting	-	82,35	17,65	0,00	86,54	86,54
IO	55,11	0,57	44,32	SVM	RSS	-	70,51	5,13	24,36	86,36	96,82
FS	49,09	5,87	45,05	KNN	Boosting	-	8,68	15,85	75,46	53,00	80,29
WI	100,00	0,00	0,00	-	-	-	-	-	-	100,00	100,00
WC	93,33	0,70	5,96	SVM	Bagging	-	64,71	11,76	23,53	97,19	98,58
IS	81,00	1,62	17,38	KNN	Boosting	-	65,75	34,25	0,00	92,43	92,43
IR	93,33	1,33	5,33	SVM	BoostingW	-	100,00	0,00	0,00	98,67	98,67

Por exemplo, para um problema considerado difícil, como a base PD, apenas 9,11% dos casos foram classificados no primeiro nível. Deste total, 8,59% foram classificados corretamente e 0,52% foram classificados erroneamente, com uma rejeição de 90,89% utilizando o classificador monolítico SVM. No segundo nível, a partir da quantidade total de casos rejeitados, 22,92% foram classificados, sendo 21,2% corretamente e 1,72% erroneamente. O segundo nível rejeitou 77,08% dos casos já rejeitados pelo primeiro nível. Finalmente, observa-se que a taxa de reconhecimento para a base PD aumentou de 8,59% para 27,89%.

No outro lado, considerando um problema fácil, como a base de dados WC, apenas 94,03% dos casos foram classificados no primeiro nível. Deste total, 93,33% foram classificados corretamente e 0,70% foram classificados erroneamente, com uma rejeição de 5,96% utilizando o classificador monolítico SVM. No segundo nível, a partir da quantidade total de casos rejeitados, 76,47% foram classificados, sendo 64,71% corretamente e 11,76% erroneamente. O segundo nível rejeitou 23,53% dos casos já rejeitados pelo primeiro nível. Finalmente, observa-se que a taxa de reconhecimento para a base WC aumentou de 93,33% para 97,19%.

A Tabela 4.12 é um complemento da tabela anterior que analisa o desempenho dos métodos em cascata que obtiveram os melhores resultados em comparação ao seu SMC correspondente (fora da abordagem de cascata). Nessa tabela são apresentados primeiramente os resultados da acurácia e confiabilidade do SMC e do método em cascata para os mesmos conjuntos de testes, e então, são apresentados os resultados das análises de redução de custo (utilizando TFV), ganho de confiabilidade e ganho de acurácia. Para

obter a redução de custo de classificação, calculou-se a razão entre o TFV para o método de cascata proposto e o TFV considerando que o problema seja sempre tratado por um SMC. A mesma relação foi utilizada para calcular os ganhos de confiabilidade e acurácia.

Tabela 4.12: Análise do desempenho dos métodos de classificação em cascata que obtiveram os melhores resultados comparado com os experimentos utilizando seus respectivos SMCs fora da abordagem em cascata

Base	**SMC**		**Cascata**		**Redução**	**Ganho de**	**Ganho de**
	acertos	**REL**	**acertos**	**REL**	**de custo**	**confiabilidade**	**acurácia**
LD	13,29	63,89	18,50	66,67	-3,06	4,35	39,13
HB	13,07	80,00	18,30	82,35	0,46	2,94	40,00
BD	35,56	89,86	36,63	90,13	-30,92	0,30	3,01
PD	27,86	93,04	27,86	93,04	-0,89	0,00	0,00
VE	29,79	84,00	59,57	90,00	45,08	7,14	100,00
SO	84,62	84,62	86,54	86,54	40,96	2,27	2,27
IO	86,36	96,82	86,36	96,82	13,18	0,00	0,00
FS	19,50	66,05	53,00	80,29	44,95	21,57	171,80
WI	-	-	100,00	100,00	-	-	-
WC	97,19	98,58	97,19	98,58	84,04	0,00	0,00
IS	91,62	91,62	92,43	92,43	72,62	0,88	0,88
IR	97,33	97,33	98,67	98,67	84,67	1,37	1,37

Como esperado, para problemas de classificação fáceis, percebe-se uma significativa redução de custos utilizando o método proposto quando comparado a um SMC, enquanto que para problemas difíceis, o custo computacional pode às vezes ter um pequeno aumento em relação ao SMC. Para um exemplo de problema de menor complexidade, como a base WC, o impacto sobre o custo computacional foi positivo, isto é, o custo foi reduzido em 84,04%. Já para um problema de maior complexidade, como a base PD, embora pequeno, o impacto sobre o custo computacional foi negativo, isto é, o custo foi aumentado em 0,89%.

É importante notar que, para as bases de dados em que a técnica RSS apresentou os melhores resultados (BD e IR), o custo para utilizar o segundo nível é baixo já que a quantidade de atributos consideradas para treinamento é menor do que a utilizada para treinar o classificador monolítico do primeiro nível.

O método de classificação em cascata mostrou ser capaz de reduzir o custo de classificação em 31,92%, em média. Para os problemas de menor complexidade, como as bases WC, IS e IR, a redução de custo foi maior que 70%. Na análise da confiabilidade, embora com valores discretos, o método proposto teve um ganho médio de 3,71%. Já na acurácia, o ganho foi significativo, principalmente nos problemas de maior complexidade, com média de 32,59%.

Em uma visão geral, como esperado, o melhor classificador monolítico nos experimentos foi o SVM. A partir das 12 bases de dados, quatro delas (IR, IS, WC e WI) tinham mais de 80% dos casos classificados no primeiro nível. Três bases de dados (IR, WI e WC) apresentaram menos de 10% dos casos rejeitados no primeiro nível. A base de dados WI não apresentou rejeições, ou seja, todas as instâncias foram classificadas pelo monolítico. Por outro lado, quatro conjuntos de dados (BD, HB, LD e PD) tinham mais de 89% dos casos rejeitados no primeiro nível. Todas estas observações tem alta correlação com os resultados de complexidade apresentados na Tabela 4.3.

Ao usar o segundo nível, a precisão foi melhorada em até 40,39 pontos percentuais quando comparado com o primeiro nível correspondente. O segundo nível foi capaz de reconhecer até 100% dos casos rejeitados pelo primeiro nível. Na maioria das vezes os conjuntos com base nas técnicas *Bagging* e *Boosting* obtiveram os melhores resultados na abordagem em cascata e os métodos seleção dinâmica apresentaram os melhores resultados apenas para a base de dados LD.

A partir da Tabela 4.13 é possível comparar os resultados da melhor classificação obtida a partir do classificador monolítico e dos sistemas de múltiplos classificadores, com os melhores resultados obtidos com a abordagem em cascata. Como podemos ver, a precisão fornecida pela abordagem em cascata na maioria das vezes ultrapassa a precisão de todas as outras abordagens.

Tabela 4.13: Desempenho do método de classificação em cascata proposto, baseado na acurácia (%), comparado com o classificador monolítico SVM e com os melhores resultados das abordagens de SMCs

Base	SVM	SMCs	Cascata
LD	5,78	16,18	**18,50**
HB	9,15	13,07	**18,30**
BD	2,14	35,56	**36,63**
PD	8,59	**27,86**	**27,86**
VE	53,19	53,90	**59,57**
SO	46,15	84,62	**86,54**
IO	55,11	**86,36**	**86,36**
FS	49,09	42,51	**53,00**
WI	**100,00**	98,88	**100,00**
WC	93,33	**97,19**	**97,19**
IS	81,00	91,62	**92,43**
IR	93,33	97,33	**98,67**

Todos resultados consideram rejeição.
Os melhores resultados estão destacadas em negrito.

É importante notar que em todos os casos de SMCs apresentados na Tabela 4.13 foi a combinação de todos os classificadores que apresentou os melhores resultados. Como mencionado anteriormente, aumentar a precisão não era o objetivo principal do método proposto, no entanto, isso se mostrou possível.

Os experimentos demonstraram que o método de cascata pode contribuir para reduzir o custo da tarefa de classificação. Para os problemas fáceis, a redução foi significativa, em 8 de 12 conjuntos foram observadas alguma redução, quatro delas sendo superior a 70%. E finalmente, pode-se dizer que, conforme observado, a redução do custo é dependente do problema e está relacionada com o nível de complexidade.

Capítulo 5

Conclusão

A proposta do presente trabalho foi um método de classificação em cascata em que um classificador monolítico é combinado com um sistema de múltiplos classificadores. Conforme observado nos resultados experimentais, tal combinação é uma estratégia interessante para lidar com casos fáceis e difíceis normalmente encontrados em um problema de classificação, apresentando uma boa relação entre a precisão e a utilização de métodos complexos na classificação.

Nos experimentos para definição do classificador monolítico para representar o primeiro nível da cascata, o SVM teve desempenho igual ou superior aos classificadores KNN, MLP, C4.5 e NB, para 10 das 12 bases utilizadas, com reconhecimento médio de 85,01%, ainda sem considerar a rejeição.

O primeiro nível da cascata, com classificador monolítico, foi capaz de classificar corretamente até 100% dos casos de teste, conforme experimentos com a base WI. Comportamento parecido foi observado nas bases mais fáceis, de acordo com o cálculo da complexidade, onde poucas instâncias eram rejeitadas para o segundo nível.

Já no segundo nível, que foi testado com diferentes sistemas de múltiplos classificadores, para as bases mais complexas, como a BD, até 97,59% das amostras foram rejeitadas pelo classificador monolítico. Trabalhando na rejeição da primeira etapa, para alguns problemas os sistemas de múltiplos classificadores puderam classificar até 100% dos casos, como visto na base IR.

Em termos de precisão, o segundo nível da cascata foi capaz de recuperar amostras rejeitadas pelo primeiro nível aumentando em até 40,38 pontos percentuais. A precisão final da cascata será sempre igual ou superior ao classificador monolítico utilizado

separadamente, já que a cascata trata as rejeições deste classificador. Além disso, os experimentos mostram que a precisão final da cascata ultrapassou a utilização dos sistemas de múltiplos classificadores, quando utilizados separadamente, em até 15,2 pontos percentuais.

Na análise dos experimentos, a configuração de cascata que se destacou pela taxa de reconhecimentos utilizava um SVM no primeiro nível e a combinação de um conjunto de classificadores KNN gerados com a técnica *Boosting*. Essa combinação na cascata teve os melhores resultados nos experimentos realizados com as bases HB, SO, FS e IS.

Outro ponto a ser destacado nos resultados da cascata é o impacto da utilização da técnica *Random Subspace Selection* no cálculo da redução de custo TFV. O RSS gera um conjunto de classificadores utilizando diferentes subconjuntos de atributos, e como o número de atributos é um dos parâmetros para calcular o TFV, o seu custo será inferior as demais técnicas utilizadas (*Bagging* e *Boosting*).

E por fim, com os experimentos realizados observar-se que, apesar das estratégias de combinação e seleção dinâmica de classificadores serem bastante populares, nem sempre devem ser utilizadas isoladamente em problemas que necessitem de rejeição. A abordagem em cascata mostrou-se um método promissor capaz de melhorar a precisão enquanto reduz o custo de classificação.

5.1 Trabalhos futuros

Outros trabalhos podem ser feitos para investigar melhor a relação da complexidade do problema e o desempenho do método de classificação em cascata proposto. Um ponto importante que pode ser otimizado de outras maneiras é o limiar de rejeição, que poderia ser definido para cada classe, por exemplo. Uma alternativa que pode ser inserida nesta abordagem de cascata está na estratégia utilizada para gerar o conjunto de classificadores utilizado no segundo nível, usando apenas as rejeições do primeiro nível para treinamento. A criação de classificadores especialistas pode ser feita com foco na região do domínio do problema, com competência complementar para cobrir uma região específica definida pelas rejeições de primeiro nível. É possível ainda fazer novos experimentos com outros métodos de seleção de classificadores, assim como se pode avaliar a inclusão de novos níveis para a cascata.

Referências Bibliográficas

ALIMOGLU, F.; ALPAYDIN, E. Combining multiple representations for pen-based handwritten digit recognition. *Turk J Elec Eng and Comp Sci*, v. 9, n. 1, p. 1–12, 2001.

ALMEIDA, H. A. de M. S. *Seleção dinâmica de classificadores baseada em filtragem e em distância adaptativa*. 2014. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal de Pernambuco, Recife.

ALPAYDIN, E.; KAYNAK, C. Cascading classifiers. *Kybernetika*, v. 34, p. 369–374, 1998.

AVNIMELECH, R.; INTRATOR, N. Boosting regression estimators. *Neural Comput.*, MIT Press, Cambridge, MA, USA, v. 11, n. 2, p. 499–520, fev. 1999. ISSN 0899-7667. Disponível em: ⟨http://dx.doi.org/10.1162/089976699300016746⟩.

BISHOP, C. M. *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford University Press, Inc., 1995. ISBN 0198538642.

BREIMAN, L. Bagging predictors. *Machine Learning*, Kluwer Academic Publishers, Hingham, MA, USA, v. 24, n. 2, p. 123–140, ago. 1996. ISSN 0885-6125. Disponível em: ⟨http://dx.doi.org/10.1023/A:1018054314350⟩.

BRITTO JR., A. S.; SABOURIN, R.; OLIVEIRA, L. E. S. Dynamic selection of classifiers-a comprehensive review. *Pattern Recogn.*, Elsevier Science Inc., New York, NY, USA, v. 47, n. 11, p. 3665–3680, nov. 2014. ISSN 0031-3203. Disponível em: ⟨http://dx.doi.org/10.1016/j.patcog.2014.05.003⟩.

BURGES, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 2, n. 2, p. 121–167, jun. 1998. ISSN 1384-5810. Disponível em: ⟨http://dx.doi.org/10.1023/A:1009715923555⟩.

CHOW, C. On optimum recognition error and reject tradeoff. *IEEE Trans. Inf. Theor.*, IEEE Press, Piscataway, NJ, USA, v. 16, n. 1, p. 41–46, set. 1970. ISSN 0018-9448. Disponível em: ⟨http://dx.doi.org/10.1109/TIT.1970.1054406⟩.

DEPEURSINGE, A.; IAVINDRASANA, J.; HIDKI, A.; COHEN, G.; GEISSBUHLER, A.; PLATON, A.; POLETTI, P.-A.; MULLER, H. Comparative performance analysis of state-of-the-art classification algorithms applied to lung tissue categorization. *J. Digital Imaging*, v. 23, n. 1, p. 18–30, 2010. Disponível em: ⟨http://dblp.uni-trier.de/db/journals/jdi/jdi23.html\#DepeursingeIHCGPPM10⟩.

DIETTERICH, T. G. Ensemble methods in machine learning. In: *Proceedings of the First International Workshop on Multiple Classifier Systems*. London, UK, UK: Springer-Verlag, 2000. (MCS '00), p. 1–15. ISBN 3-540-67704-6. Disponível em: ⟨http://dl.acm.org/citation.cfm?id=648054.743935⟩.

FUMERA, G.; PILLAI, I.; ROLI, F. A two-stage classifier with reject option for text categorisation. In: FRED, A.; CAELLI, T.; DUIN, R.; CAMPILHO, A.; RIDDER, D. de (Ed.). *Structural, Syntactic, and Statistical Pattern Recognition*. Springer Berlin Heidelberg, 2004, (Lecture Notes in Computer Science, v. 3138). p. 771–779. ISBN 978-3-540-22570-6. Disponível em: ⟨http://dx.doi.org/10.1007/978-3-540-27868-9_84⟩.

FUMERA, G.; ROLI, F.; GIACINTO, G. Reject option with multiple thresholds. *Pattern Recognition*, v. 33, p. 2099–2101, 2000.

HO, T. K. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 20, n. 8, p. 832–844, ago. 1998. ISSN 0162-8828. Disponível em: ⟨http://dx.doi.org/10.1109/34.709601⟩.

HO, T. K.; BASU, M. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, p. 289–300, 2002.

HSU, C.-W.; CHANG, C.-C.; LIN, C.-J. *A Practical Guide to Support Vector Classification*. [S.l.], 2010. Disponível em: ⟨http://www.csie.ntu.edu.tw/~cjlin/papers.html⟩.

JAIN, A. K.; DUIN, R. P. W.; MAO, J. Statistical pattern recognition: A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 22, n. 1, p. 4–37, jan. 2000. ISSN 0162-8828. Disponível em: ⟨http://dx.doi.org/10.1109/34.824819⟩.

KITTLER, J. Combining classifiers: A theoretical framework. *Pattern Analysis and Applications*, Springer-Verlag, v. 1, n. 1, p. 18–27, 1998. ISSN 1433-7541. Disponível em: ⟨http://dx.doi.org/10.1007/BF01238023⟩.

KO, A. H. R.; SABOURIN, R.; BRITTO JR., A. S. From dynamic classifier selection to dynamic ensemble selection. *Pattern Recogn.*, Elsevier Science Inc., New York, NY, USA, v. 41, n. 5, p. 1718–1731, maio 2008. ISSN 0031-3203. Disponível em: ⟨http://dx.doi.org/10.1016/j.patcog.2007.10.015⟩.

KOERICH, A. L. Rejection strategies for handwritten word recognition. In: *Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition*. Washington, DC, USA: IEEE Computer Society, 2004. (IWFHR '04), p. 479–484. ISBN 0-7695-2187-8. Disponível em: ⟨http://dx.doi.org/10.1109/IWFHR.2004.88⟩.

KUNCHEVA, L. I. *Combining Pattern Classifiers: Methods and Algorithms*. [S.l.]: Wiley-Interscience, 2004. ISBN 0471210781.

LAST, M.; BUNKE, H.; KANDEL, A. A feature-based serial approach to classifier combination. *Pattern Analysis and Applications*, Springer-Verlag London Limited, v. 5, n. 4, p. 385–398, 2002. ISSN 1433-7541. Disponível em: ⟨http://dx.doi.org/10.1007/s100440200034⟩.

LICHMAN, M. *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences, 2013. Disponível em: ⟨http://archive.ics.uci.edu/ml⟩.

MARTINS, J.; OLIVEIRA, L. S.; NISGOSKI, S.; SABOURIN, R. A database for automatic classification of forest species. *Machine Vision and Applications*, Springer-Verlag, v. 24, n. 3, p. 567–578, 2013. ISSN 0932-8092. Disponível em: ⟨http://dx.doi.org/10.1007/s00138-012-0417-5⟩.

MICHIE, D.; SPIEGELHALTER, D. J.; TAYLOR, C. C.; CAMPBELL, J. (Ed.). *Machine Learning, Neural and Statistical Classification*. Upper Saddle River, NJ, USA: Ellis Horwood, 1994. ISBN 0-13-106360-X.

MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072.

NGUYEN, D. Q.; NGUYEN, D. Q.; PHAM, S. B. A two-stage classifier for sentiment analysis. In: . In press: [s.n.], 2013.

NIU, X.-X.; SUEN, C. Y. A novel hybrid cnn-svm classifier for recognizing handwritten digits. *Pattern Recogn.*, Elsevier Science Inc., New York, NY, USA, v. 45, n. 4, p. 1318–1325, abr. 2012. ISSN 0031-3203. Disponível em: ⟨http://dx.doi.org/10.1016/j.patcog.2011.09.021⟩.

OLIVEIRA, L. S.; BRITTO JR., A. S. J.; SABOURIN, R. Improving cascading classifiers with particle swarm optimization. In: *Proceedings of the Eighth International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE Computer Society, 2005. (ICDAR '05), p. 570–574. ISBN 0-7695-2420-6. Disponível em: ⟨http://dx.doi.org/10.1109/ICDAR.2005.138⟩.

ORRIOLS-PUIG, A.; MACIA, N.; HO, T. K. *Documentation for the data complexity library in C++*. [S.l.]: Tech. Rep., La Salle - Universitat Ramon Llull, 2010.

PUDIL, P.; NOVOVICOVA, J.; BLAHA, S.; KITTLER, J. Multistage pattern recognition with reject option. In: *Pattern Recognition, 1992. Vol.II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*. [S.l.: s.n.], 1992. p. 92–95.

SANCHEZ, J. S.; MOLLINEDA, R. A.; SOTOCA, J. M. An analysis of how training data complexity affects the nearest neighbor classifiers. *Pattern Anal. Appl.*, Springer-Verlag, London, UK, v. 10, n. 3, p. 189–201, jul. 2007. ISSN 1433-7541. Disponível em: ⟨http://dx.doi.org/10.1007/s10044-007-0061-2⟩.

WESTON, J.; WATKINS, C. Support vector machines for multi-class pattern recognition. In: *ESANN 1999, 7th European Symposium on Artificial Neural Networks, Bruges,*

Belgium, April 21-23, 1999, Proceedings. [s.n.], 1999. p. 219–224. Disponível em: ⟨https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es1999-461.pdf⟩.

WOODS, K.; KEGELMEYER JR., W. P.; BOWYER, K. Combination of multiple classifiers using local accuracy estimates. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 19, n. 4, p. 405–410, abr. 1997. ISSN 0162-8828. Disponível em: ⟨http://dx.doi.org/10.1109/34.588027⟩.

Apêndice A

Resultados gerais dos experimentos

No Capítulo 4 foi apresentado os diversos experimentos realizados neste trabalho, como avaliação dos classificadores monolíticos, combinação de classificadores, métodos de seleção dinâmica e método de cascata. Os resultados apresentados no capítulo de experimentos estão sumarizados com o objetivo de avaliar as hipóteses levantadas na definição do trabalho.

Neste apêndice serão apresentados os resultados gerais obtidos na realização dos principais experimentos. Estes dados poderão ser úteis para realizar outras análises e comparações do desempenho do método de classificação em cascata proposto neste trabalho.

A.1 Análise da rejeição no classificador monolítico

A definição do limiar de rejeição do classificador monolítico é um ponto crucial para o bom desempenho do método proposto. O classificador monolítico é o primeiro nível da cascata. As amostras de testes rejeitadas por ele, serão submetidas a um sistema de classificação mais complexo e custoso.

Logo, a rejeição neste ponto é responsável por separar os padrões fáceis, que são as amostras que o classificador monolítico tem capacidade de rotular corretamente, e os padrões difíceis, que são as amostras que necessitam de um sistema mais robusto. Além disso, a rejeição é adotada para estabelecer uma meta de tolerância de erro, de acordo com o problema envolvido, que neste caso é erro $\leq 1\%$.

Os gráficos anexados na sequência mostram a relação entre a taxa de rejeição e a taxa de erro para o classificador monolítico SVM obtidos na fase de validação. São 12 gráficos, um para cada base de dados utilizada no experimento, apresentados na ordem de complexidade, de acordo com a Tabela 4.3, começando pelos mais complexos. Pode-se observar que para reduzir a taxa de erros para 1% (valor destacado nos gráficos), em alguns casos, rejeita-se quase toda as amostras de teste. Quando o crescimento da rejeição tem proporção maior do que a redução da taxa de erros, significa que amostras que seriam classificadas corretamente estão sendo rejeitadas. Em um sistema ideal, ao incluir a opção de rejeição teria $TE = 0$, $TR = TE_{\text{sem rejeição}}$ e $TA_{\text{com rejeição}} = TA_{\text{sem rejeição}}$.

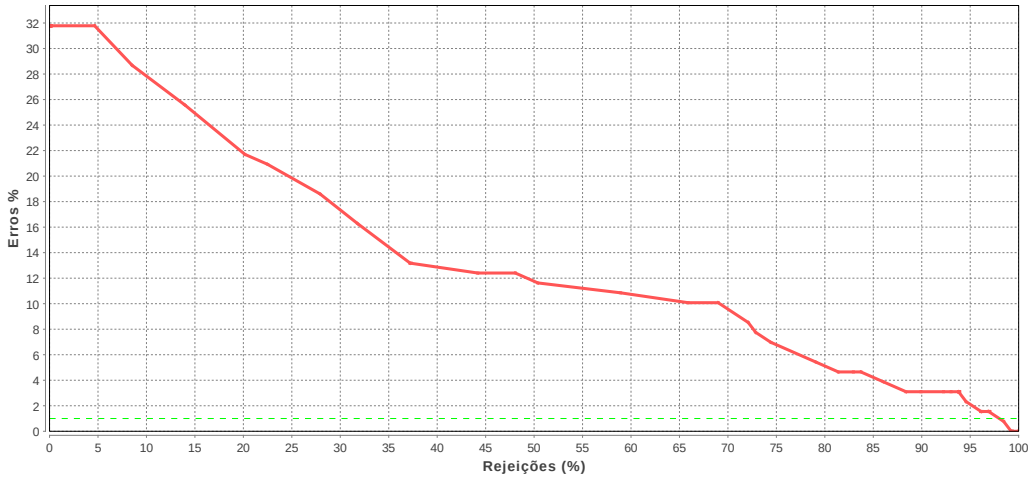

Figura A.1: Gráfico de análise da relação rejeição-erro do SVM para a base LD

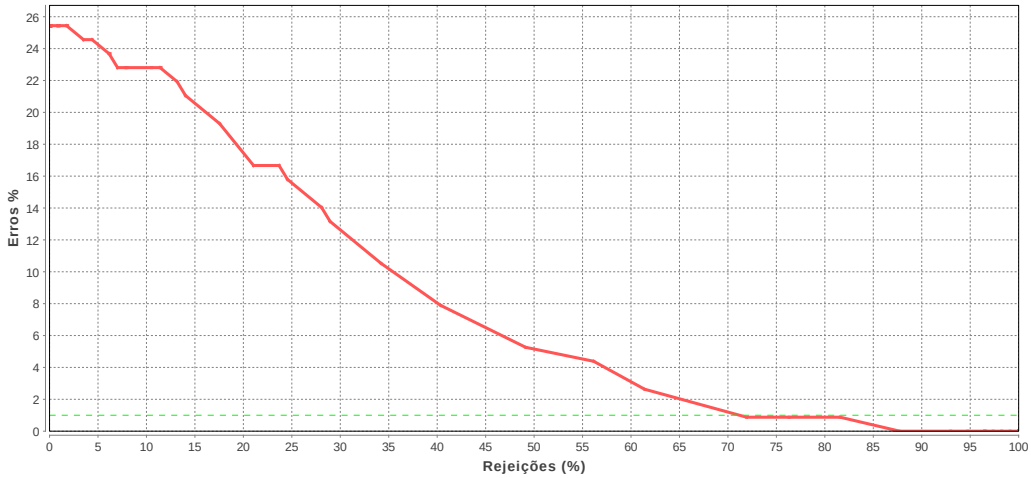

Figura A.2: Gráfico de análise da relação rejeição-erro do SVM para a base HS

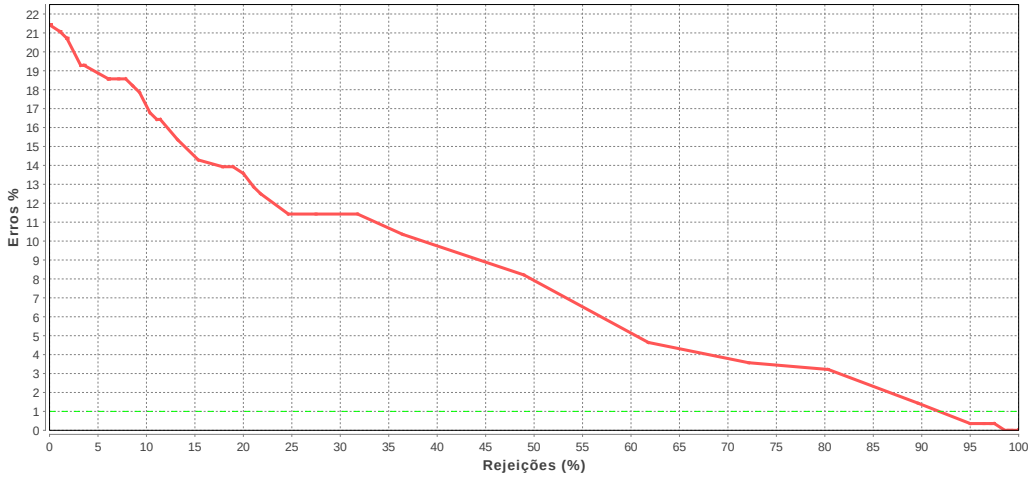

Figura A.3: Gráfico de análise da relação rejeição-erro do SVM para a base BT

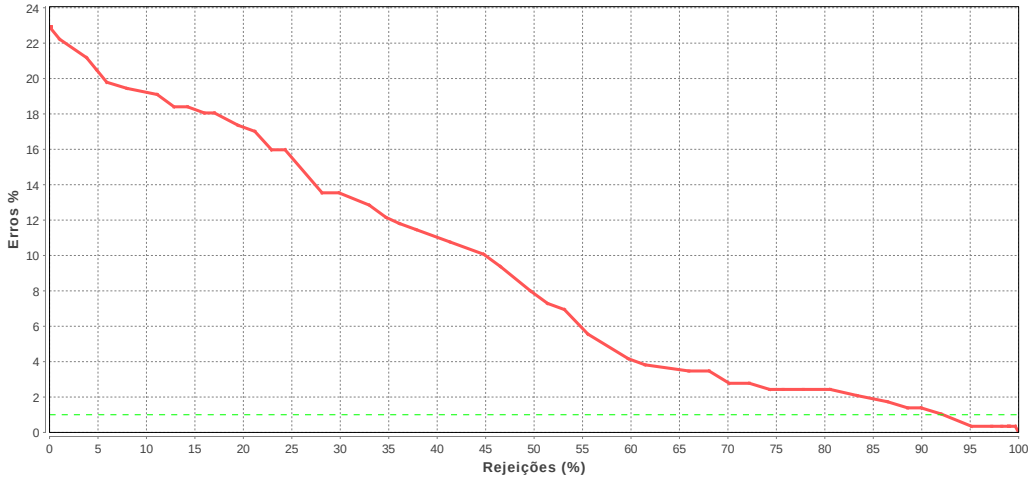

Figura A.4: Gráfico de análise da relação rejeição-erro do SVM para a base PD

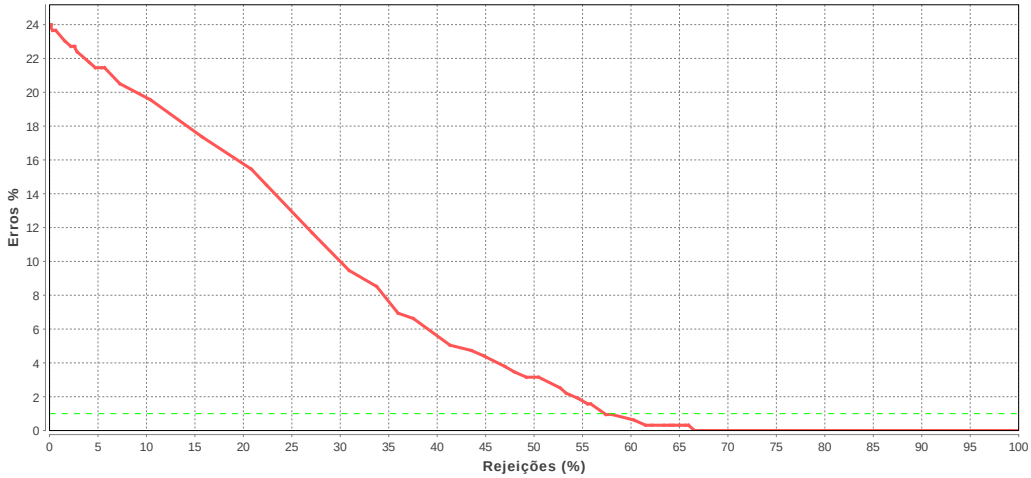

Figura A.5: Gráfico de análise da relação rejeição-erro do SVM para a base VS

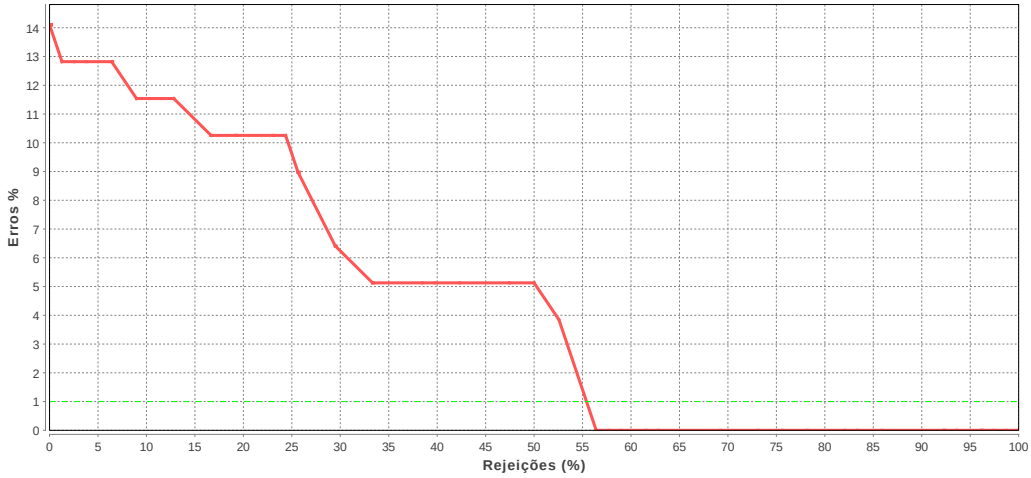

Figura A.6: Gráfico de análise da relação rejeição-erro do SVM para a base SO

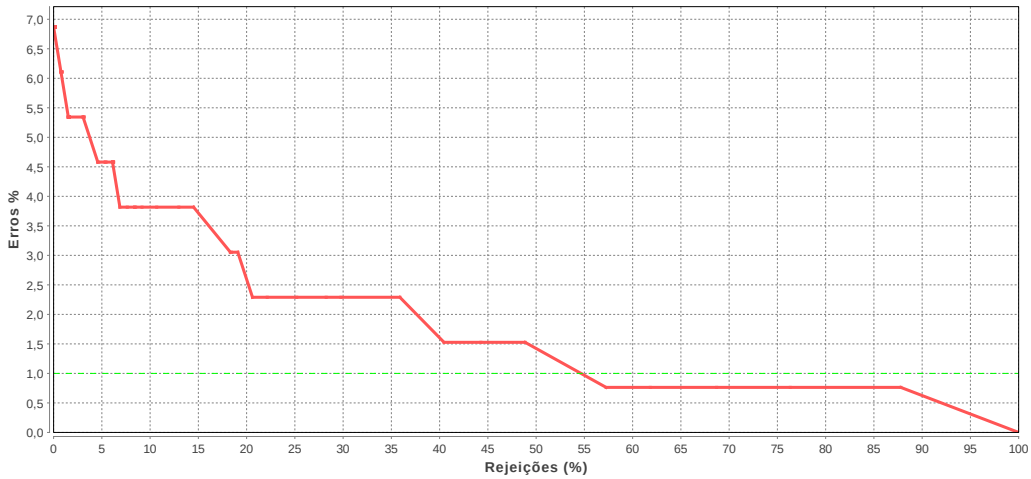

Figura A.7: Gráfico de análise da relação rejeição-erro do SVM para a base IO

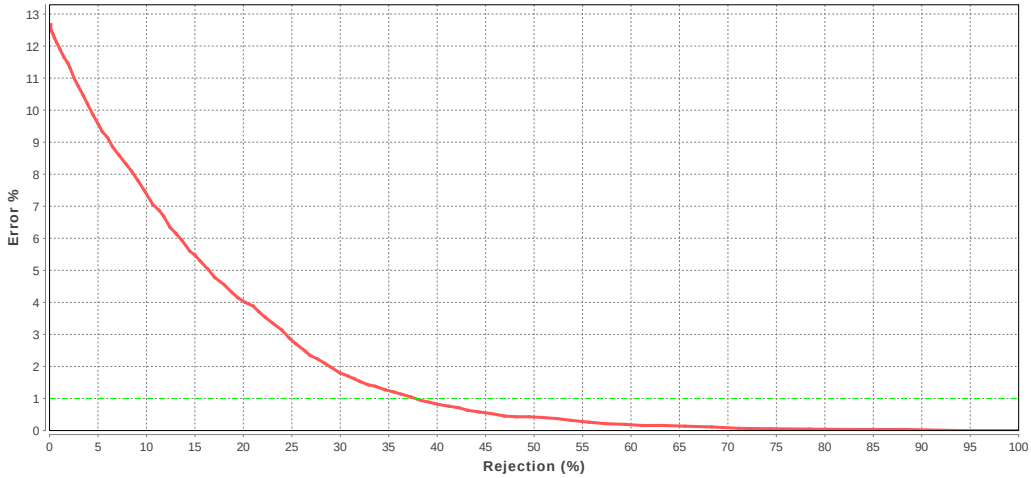

Figura A.8: Gráfico de análise da relação rejeição-erro do SVM para a base FS

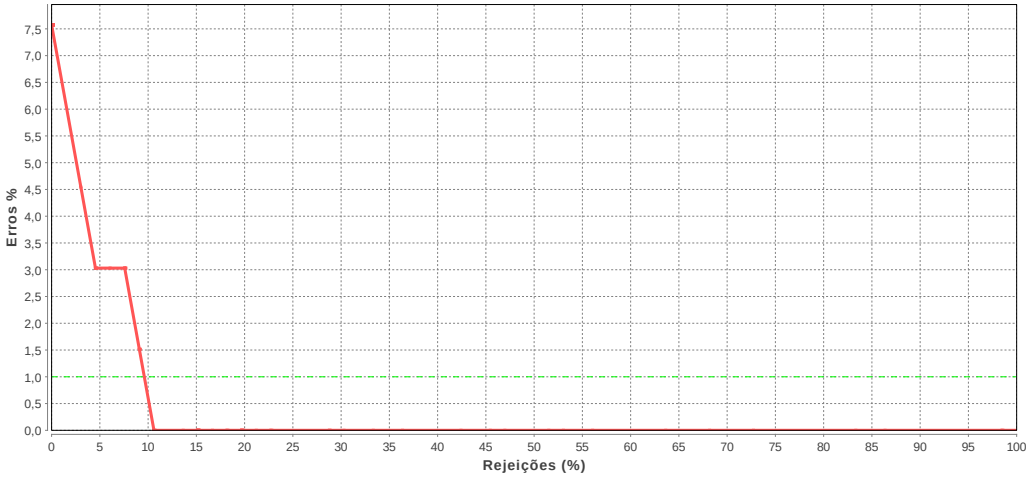

Figura A.9: Gráfico de análise da relação rejeição-erro do SVM para a base WI

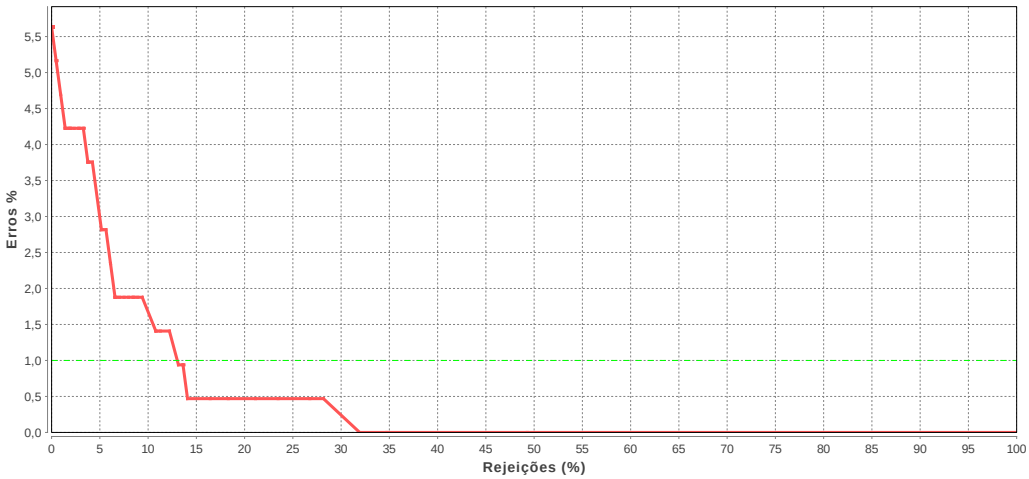

Figura A.10: Gráfico de análise da relação rejeição-erro do SVM para a base WC

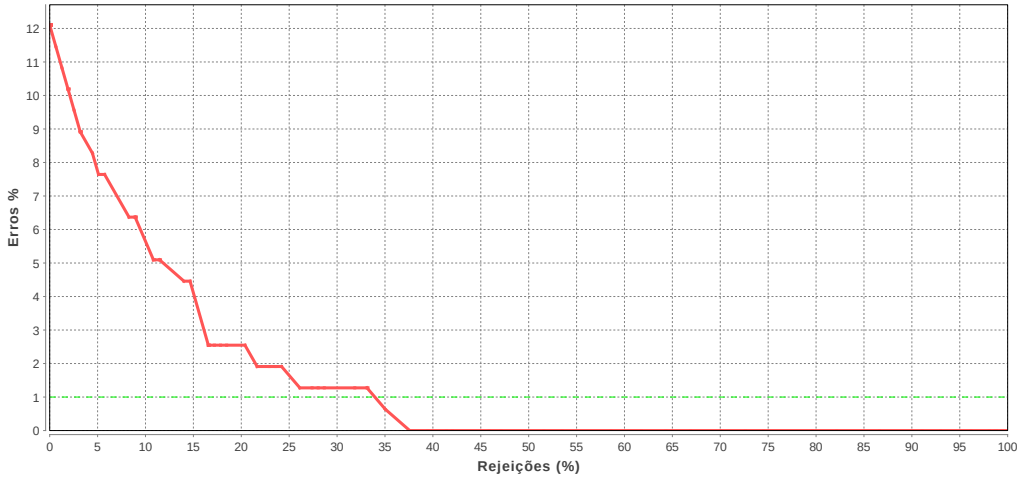

Figura A.11: Gráfico de análise da relação rejeição-erro do SVM para a base IS

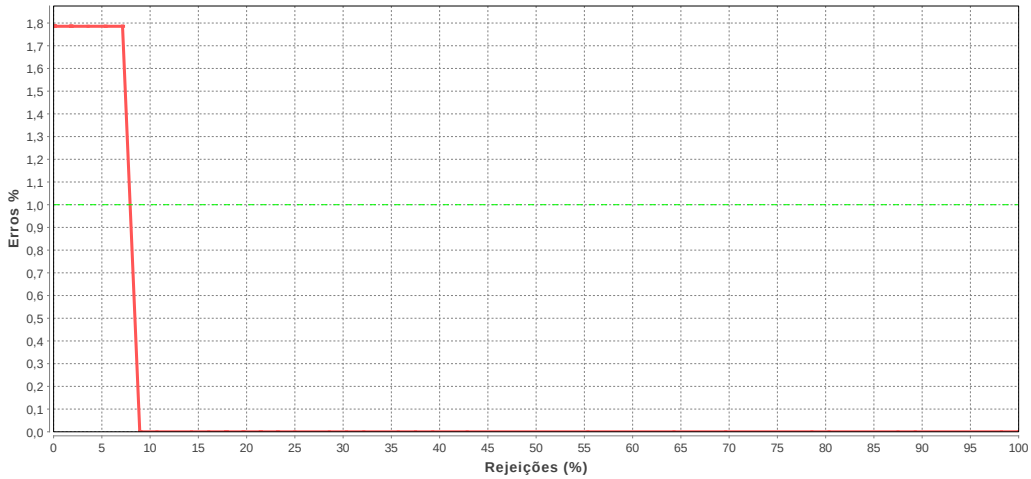

Figura A.12: Gráfico de análise da relação rejeição-erro do SVM para a base IR

O limiar de rejeição estabelece a probabilidade mínima requerida na tarefa de classificação para aceitar ou recusar a decisão de um classificador ou de um sistema de múltiplos classificadores. Um limiar de rejeição inadequado poderá comprometer o desempenho do sistema. Se muito alto, amostras que seriam corretamente classificadas no primeiro nível da cascata serão rejeitadas, demandando uma reanálise no segundo nível, aumentando o custo da classificação. Do mesmo modo, para um limiar de rejeição muito baixo, amostras que deveriam ser rejeitadas pela baixa confiança na classificação serão aceitas pelo primeiro nível, podendo comprometer a meta da taxa de erros $\leq 1\%$.

Os gráficos anexados na sequência mostram uma outra visão da relação rejeição-erro, apresentado a taxa de rejeição, taxa de erro e cálculo da confiabilidade do classificador monolítico SVM obtidos na fase de validação. Estes gráficos foram obtidos experimentalmente variando o limiar de rejeição de 0% até 100%, calculando a cada ponto cada uma das taxas mencionadas. São 12 gráficos, um para cada base de dados utilizada no experimento, apresentados na ordem de complexidade, de acordo com a Tabela 4.3, começando pelos mais complexos.

Observa-se novamente que para reduzir a taxa de erros em algumas bases, a taxa de rejeição torna-se muito alta. Em alguns trechos dos gráficos é possível notar que mesmo quando a taxa de erros permanece constante, há um crescimento da taxa de rejeições. Como as taxas são complementares, logo se conclui que está havendo claramente uma redução da taxa de acertos.

Outro item a ser observado é o reflexo da confiabilidade de acordo com a variação

das taxas de erros e de rejeições. Em alguns pontos a medida de confiabilidade é decrescente. Isso é consequência da observação anterior, onde a taxa de rejeição aumenta sem diminuir a taxa de erros, ou seja, está diminuindo a taxa de acertos, afetando diretamente a confiabilidade do classificador.

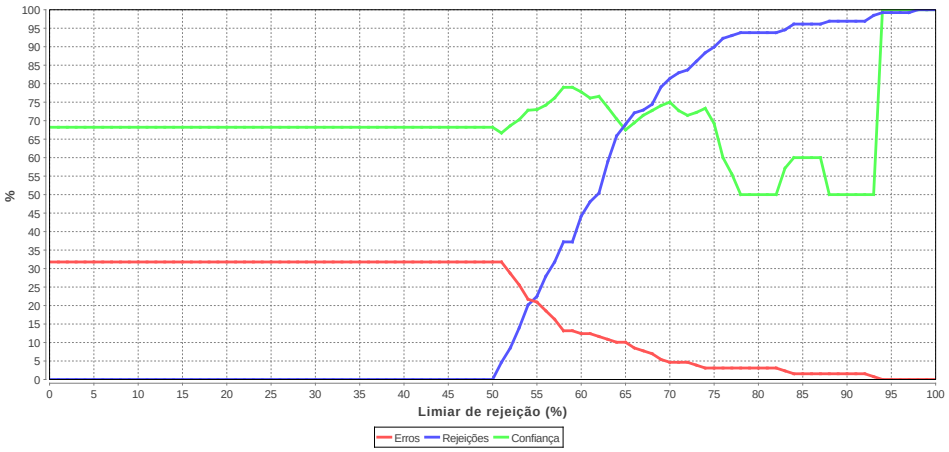

Figura A.13: Gráfico de análise da taxa de rejeição, taxa de erro e cálculo da confiança do SVM para a base LD

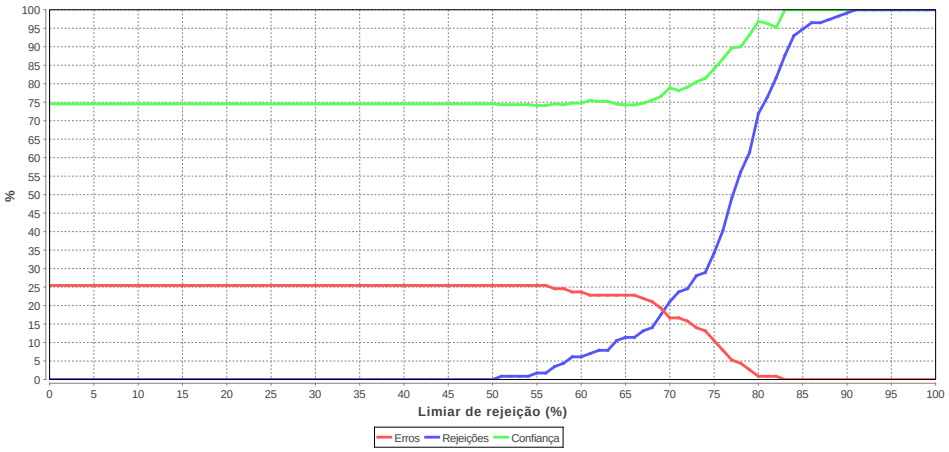

Figura A.14: Gráfico de análise da taxa de rejeição, taxa de erro e cálculo da confiança do SVM para a base HS

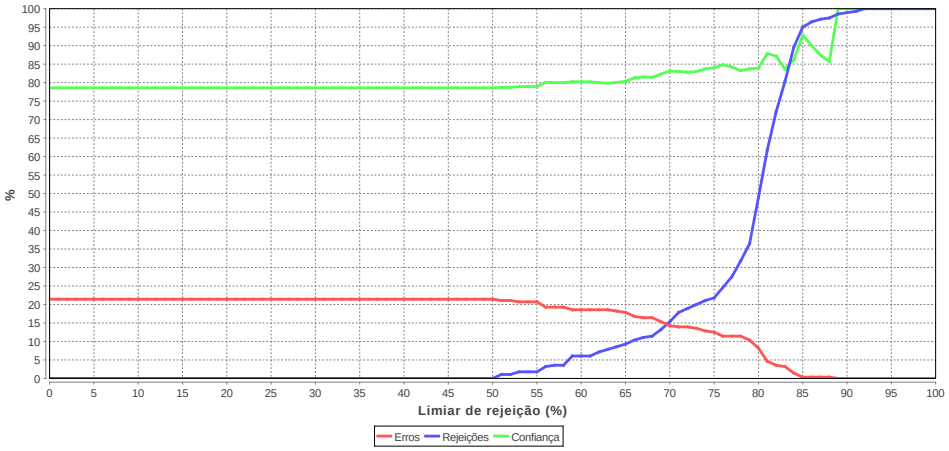

Figura A.15: Gráfico de análise da taxa de rejeição, taxa de erro e cálculo da confiança do SVM para a base BT

Figura A.16: Gráfico de análise da taxa de rejeição, taxa de erro e cálculo da confiança do SVM para a base PD

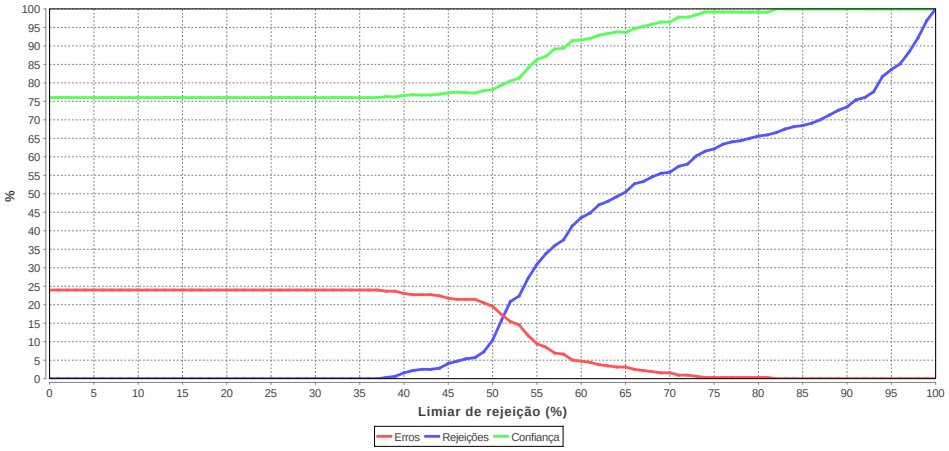

Figura A.17: Gráfico de análise da taxa de rejeição, taxa de erro e cálculo da confiança do SVM para a base VS

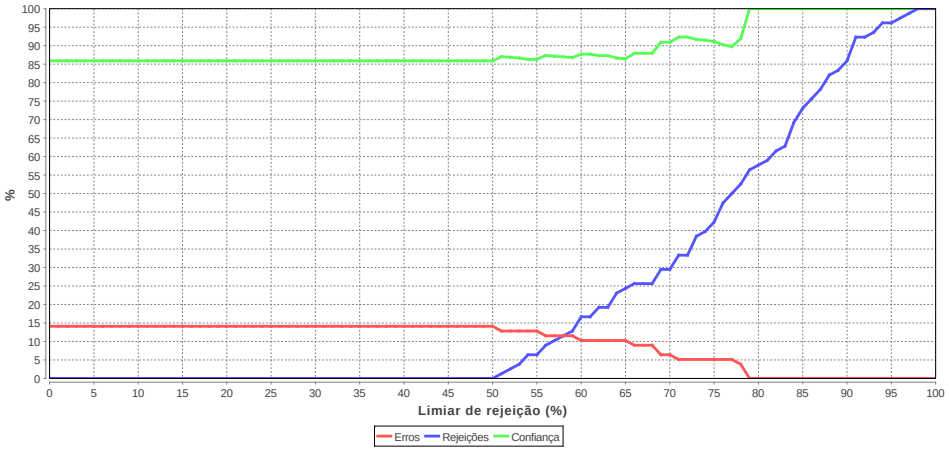

Figura A.18: Gráfico de análise da taxa de rejeição, taxa de erro e cálculo da confiança do SVM para a base SO

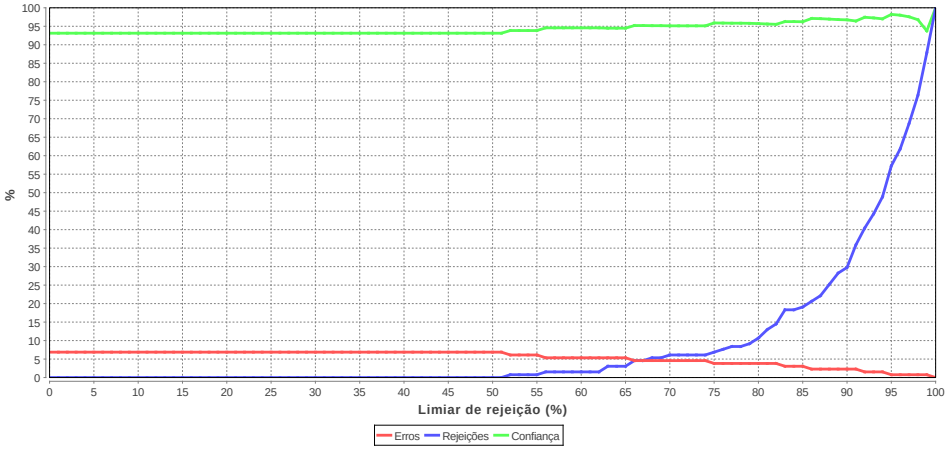

Figura A.19: Gráfico de análise da taxa de rejeição, taxa de erro e cálculo da confiança do SVM para a base IO

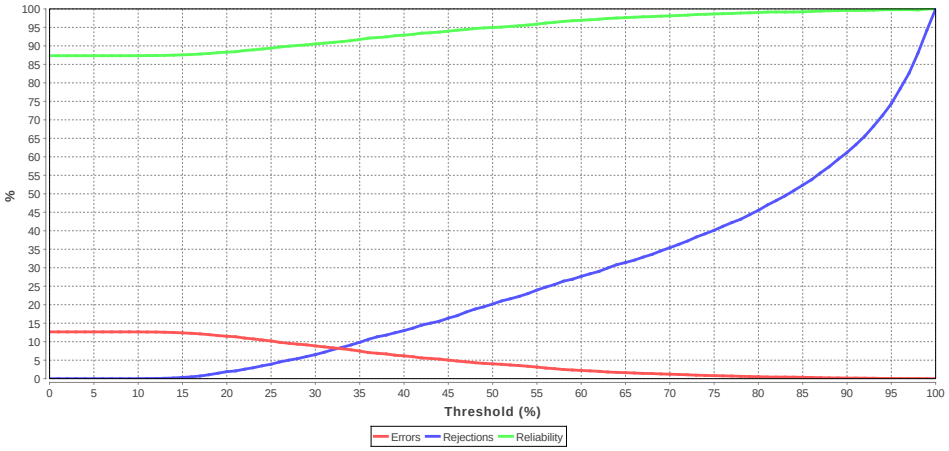

Figura A.20: Gráfico de análise da taxa de rejeição, taxa de erro e cálculo da confiança do SVM para a base FS

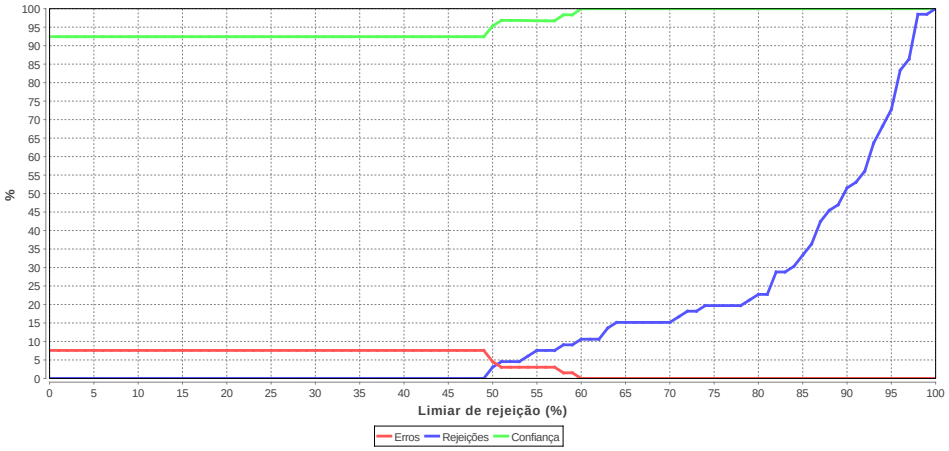

Figura A.21: Gráfico de análise da taxa de rejeição, taxa de erro e cálculo da confiança do SVM para a base WI

Figura A.22: Gráfico de análise da taxa de rejeição, taxa de erro e cálculo da confiança do SVM para a base WC

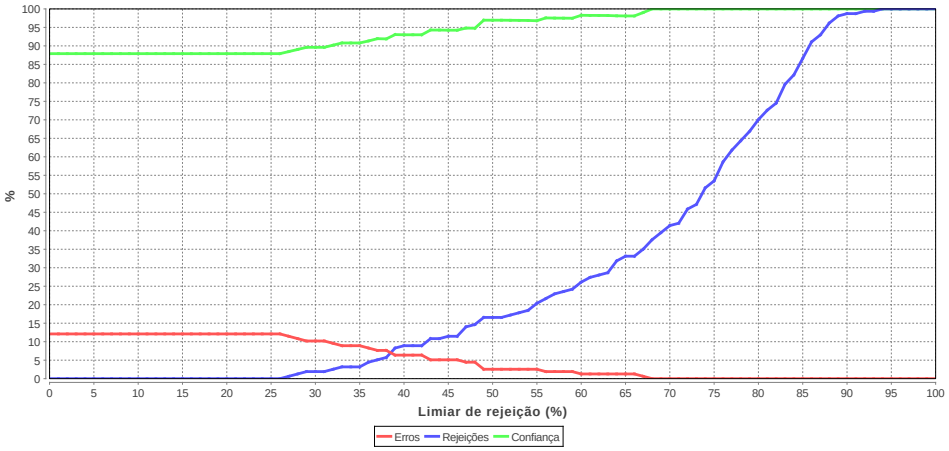

Figura A.23: Gráfico de análise da taxa de rejeição, taxa de erro e cálculo da confiança do SVM para a base IS

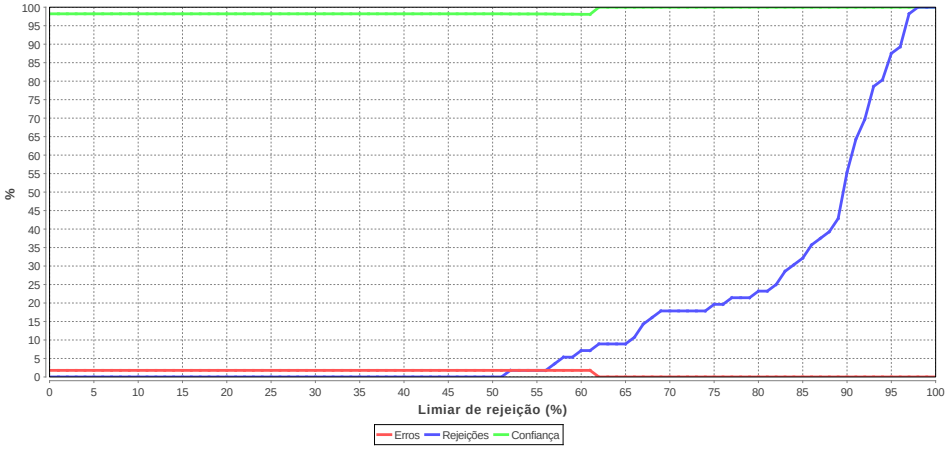

Figura A.24: Gráfico de análise da taxa de rejeição, taxa de erro e cálculo da confiança do SVM para a base IR

A.2 Combinação de conjunto de classificadores

As tabelas a seguir apresentam os resultados experimentais utilizando a combinação de todos os classificadores dos *ensembles*. Os conjuntos de classificadores são gerados com as técnicas *Bagging*, *Boosting* e *Random Subspace Selection*, utilizando dois classificadores base, KNN e SVM, totalizando seis experimentos. Os experimentos foram realizados de duas formas: sem rejeição e com rejeição. Os resultados obtidos como, a taxa de acertos, a taxa de erros e a confiabilidade (REL), são apresentados nas próximas tabelas.

Tabela A.1: Desempenho da combinação do conjunto de classificadores KNN gerados com Bagging

Base	Sem rejeição		Com rejeição			
	acertos	erros	acertos	erros	rejeição	REL
LD	61,27	38,73	3,47	0,58	95,95	85,71
HB	75,16	24,84	10,46	1,96	87,58	84,21
BD	77,81	22,19	11,76	0,80	87,43	93,62
PD	70,31	29,69	16,15	1,30	82,55	92,54
VE	70,69	29,31	29,08	1,65	69,27	94,62
SO	80,77	19,23	46,15	2,88	50,96	94,12
IO	81,25	18,75	0,00	0,00	100,00	–
FS	58,68	41,32	30,80	3,75	65,45	89,15
WI	97,75	2,25	93,26	1,12	5,62	98,81
WC	97,54	2,46	92,98	1,05	5,96	98,88
IS	89,76	10,24	79,52	3,86	16,62	95,37
IR	98,67	1,33	97,33	1,33	1,33	98,65

Tabela A.2: Desempenho da combinação do conjunto de classificadores KNN gerados com Boosting

Base	Sem rejeição		Com rejeição			
	acertos	erros	acertos	erros	rejeição	REL
LD	63,58	36,42	0,00	0,00	100,00	–
HB	75,82	24,18	13,07	3,27	83,66	80,00
BD	79,14	20,86	0,00	0,00	100,00	–
PD	73,44	26,56	0,00	0,00	100,00	–
VE	70,69	29,31	0,00	0,00	100,00	–
SO	84,62	15,38	84,62	15,38	0,00	84,62
IO	85,23	14,77	59,66	7,39	32,95	88,98
FS	57,82	42,18	19,50	10,02	70,48	66,05
WI	97,75	2,25	88,76	1,12	10,11	98,75
WC	97,89	2,11	50,53	1,05	48,42	97,96
IS	91,62	8,38	91,62	8,38	0,00	91,62
IR	96,00	4,00	96,00	4,00	0,00	96,00

Tabela A.3: Desempenho da combinação do conjunto de classificadores KNN gerados com RSS

Base	Sem rejeição		Com rejeição			
	acertos	erros	acertos	erros	rejeição	REL
LD	64,74	35,26	5,78	0,58	93,64	90,91
HB	73,86	26,14	8,50	1,96	89,54	81,25
BD	77,27	22,73	12,30	1,07	86,63	92,00
PD	74,74	25,26	20,05	0,78	79,17	96,25
VE	73,05	26,95	19,15	0,24	80,61	98,78
SO	88,46	11,54	63,46	1,92	34,62	97,06
IO	90,91	9,09	56,82	1,14	42,05	98,04
FS	59,87	40,13	32,04	3,90	64,06	89,14
WI	97,75	2,25	95,51	2,25	2,25	97,70
WC	96,84	3,16	92,63	1,05	6,32	98,88
IS	92,38	7,62	80,57	1,52	17,90	98,14
IR	96,00	4,00	92,00	1,33	6,67	98,57

Tabela A.4: Desempenho da combinação do conjunto de classificadores SVM gerados com Bagging

Base	Sem rejeição		Com rejeição			
	acertos	erros	acertos	erros	rejeição	REL
LD	65,90	34,10	15,61	5,20	79,19	75,00
HB	75,16	24,84	12,42	1,31	86,27	90,48
BD	77,27	22,73	11,50	1,60	86,90	87,76
PD	78,39	21,61	27,86	2,08	70,05	93,04
VE	79,20	20,80	53,90	2,13	43,97	96,20
SO	79,81	20,19	68,27	9,62	22,12	87,65
IO	93,18	6,82	79,55	2,27	18,18	97,22
FS	70,56	29,44	42,51	4,19	53,30	91,02
WI	98,88	1,12	98,88	1,12	0,00	98,88
WC	98,25	1,75	97,19	1,40	1,40	98,58
IS	88,52	11,48	75,05	2,24	22,71	97,10
IR	94,67	5,33	94,67	5,33	0,00	94,67

Tabela A.5: Desempenho da combinação do conjunto de classificadores SVM gerados com Boosting

Base	Sem rejeição		Com rejeição			
	acertos	erros	acertos	erros	rejeição	REL
LD	68,21	31,79	0,00	0,00	100,00	–
HB	72,55	27,45	6,54	3,27	90,20	66,67
BD	79,14	20,86	3,21	0,80	95,99	80,00
PD	77,86	22,14	3,65	0,78	95,57	82,35
VE	79,43	20,57	29,79	5,67	64,54	84,00
SO	76,92	23,08	0,00	0,00	100,00	–
IO	93,75	6,25	0,00	0,00	100,00	–
FS	69,39	30,61	0,00	0,00	100,00	–
WI	94,38	5,62	94,38	5,62	0,00	94,38
WC	97,19	2,81	94,39	1,75	3,86	98,18
IS	85,86	14,14	17,10	0,90	82,00	94,97
IR	96,00	4,00	96,00	4,00	0,00	96,00

Tabela A.6: Desempenho da combinação do conjunto de classificadores SVM gerados com RSS

Base	Sem rejeição		Com rejeição			
	acertos	erros	acertos	erros	rejeição	REL
LD	60,12	39,88	16,18	5,20	78,61	75,68
HB	73,86	26,14	7,84	1,96	90,20	80,00
BD	76,20	23,80	35,56	4,01	60,43	89,86
PD	76,56	23,44	22,66	1,30	76,04	94,57
VE	76,12	23,88	42,79	0,95	56,26	97,84
SO	75,96	24,04	56,73	6,73	36,54	89,39
IO	94,32	5,68	86,36	2,84	10,80	96,82
FS	68,78	31,22	41,58	4,70	53,72	89,85
WI	97,75	2,25	97,75	1,12	1,12	98,86
WC	95,44	4,56	90,18	0,35	9,47	99,61
IS	92,38	7,62	74,71	1,90	23,38	97,51
IR	94,67	5,33	93,33	2,67	4,00	97,22

A.3 Seleção dinâmica de classificadores

As tabelas a seguir apresentam os resultados experimentais utilizando a seleção dinâmica de classificadores disponíveis nos *ensembles*. Os conjuntos de classificadores são gerados com a técnica *Bagging*, utilizando o KNN como classificador base. Foram utilizados quatro métodos de seleção dinâmica, conforme Seção 2.2.2: DS-OLA, DS-LCA, KNORA-Eliminate e KNORA-Union. Os experimentos foram realizados de duas formas: sem rejeição e com rejeição. Os resultados obtidos como, a taxa de acertos, a taxa de erros e a confiabilidade (REL), são apresentados nas próximas tabelas.

Tabela A.7: Desempenho da seleção dinâmica de classificadores DS-OLA em conjunto de classificadores KNN gerados com Bagging

Base	Sem rejeição		Com rejeição			
	acertos	erros	acertos	erros	rejeição	REL
LD	58,96	41,04	13,29	7,51	79,19	63,89
HB	73,20	26,80	8,50	1,96	89,54	81,25
BD	78,34	21,66	7,49	0,53	91,98	93,33
PD	69,27	30,73	0,00	0,00	100,00	–
VE	66,43	33,57	0,00	0,00	100,00	–
SO	78,85	21,15	0,00	0,00	100,00	–
IO	80,68	19,32	0,00	0,00	100,00	–
FS	53,15	46,85	0,00	0,00	100,00	–
WI	97,75	2,25	84,27	1,12	14,61	98,68
WC	96,49	3,51	94,39	1,40	4,21	98,53
IS	89,14	10,86	0,00	0,00	100,00	–
IR	96,00	4,00	96,00	4,00	0,00	96,00

Tabela A.8: Desempenho da seleção dinâmica de classificadores DS-LCA em conjunto de classificadores KNN gerados com Bagging

Base	Sem rejeição		Com rejeição			
	acertos	erros	acertos	erros	rejeição	REL
LD	60,69	39,31	4,05	2,89	93,06	58,33
HB	73,86	26,14	8,50	1,96	89,54	81,25
BD	78,34	21,66	8,29	0,80	90,91	91,18
PD	71,61	28,39	0,00	0,00	100,00	–
VE	67,38	32,62	0,00	0,00	100,00	–
SO	69,23	30,77	14,42	7,69	77,88	65,22
IO	80,11	19,89	0,00	0,00	100,00	–
FS	57,55	42,45	0,00	0,00	100,00	–
WI	97,75	2,25	89,89	1,12	8,99	98,77
WC	96,49	3,51	89,82	1,05	9,12	98,84
IS	86,71	13,29	13,57	4,00	82,43	77,24
IR	97,33	2,67	0,00	0,00	100,00	–

Tabela A.9: Desempenho da seleção dinâmica de classificadores KNORA-Eliminate em conjunto de classificadores KNN gerados com Bagging

Base	Sem rejeição		Com rejeição			
	acertos	erros	acertos	erros	rejeição	REL
LD	61,27	38,73	2,89	1,16	95,95	71,43
HB	73,20	26,80	7,84	1,96	90,20	80,00
BD	75,67	24,33	10,96	1,07	87,97	91,11
PD	72,14	27,86	0,00	0,00	100,00	–
VE	64,54	35,46	0,00	0,00	100,00	–
SO	66,35	33,65	11,54	2,88	85,58	80,00
IO	80,11	19,89	0,00	0,00	100,00	–
FS	53,69	46,31	0,00	0,00	100,00	–
WI	96,63	3,37	94,38	1,12	4,49	98,82
WC	94,74	5,26	93,33	2,46	4,21	97,44
IS	87,10	12,90	2,76	0,14	97,10	95,08
IR	94,67	5,33	94,67	5,33	0,00	94,67

Tabela A.10: Desempenho da seleção dinâmica de classificadores KNORA-Union em conjunto de classificadores KNN gerados com Bagging

Base	Sem rejeição		Com rejeição			
	acertos	erros	acertos	erros	rejeição	REL
LD	67,05	32,95	10,40	3,47	86,13	75,00
HB	73,20	26,80	10,46	1,96	87,58	84,21
BD	79,41	20,59	10,16	1,07	88,77	90,48
PD	70,83	29,17	17,19	1,04	81,77	94,29
VE	66,90	33,10	0,00	0,00	100,00	–
SO	80,77	19,23	0,00	0,00	100,00	–
IO	76,14	23,86	0,00	0,00	100,00	–
FS	57,14	42,03	16,55	0,91	82,54	94,79
WI	96,63	3,37	95,51	1,12	3,37	98,84
WC	96,84	3,16	94,04	1,05	4,91	98,89
IS	87,81	12,19	6,62	0,90	92,48	87,97
IR	97,33	2,67	97,33	2,67	0,00	97,33

A.4 Classificação em cascata

As tabelas a seguir apresentam os resultados experimentais utilizando o método de classificação em cascata proposto neste trabalho. O primeiro nível da cascata é composto por um classificador SVM. Já o segundo nível foi testado com 11 variações de sistemas de múltiplos classificadores, sendo: seis métodos de combinação de classificadores com *pools* gerados com as técnicas *Bagging*, *Boosting* e *Random Subspace Selection*, utilizando dois classificadores base, KNN e SVM; um método de combinação de classificadores com *pools* gerados com a variação da técnica *Boosting* (*BoostingW*, ver Seção 3.2.1), utilizando classificador base SVM; e, quatro métodos de seleção dinâmica, DS-OLA, DS-LCA, KNORA-Eliminate e KNORA-Union, com *pools* gerados com a técnica BAG, utilizando classificador base KNN. Os resultados obtidos como, a taxa de acertos, a taxa de erros e a confiabilidade (REL), são apresentados nas próximas tabelas.

Tabela A.11: Desempenho da cascata de dois níveis com SVM no primeiro nível e conjunto de classificadores KNN gerado com Bagging no segundo nível

Base	1º nível			2º nível			Cascata	
	acertos	erros	rejeições	acertos	erros	rejeições	acertos	REL
LD	5,78	1,16	93,06	3,73	0,62	95,65	9,25	84,21
HB	9,15	1,31	89,54	5,84	0,73	93,43	14,38	88,00
BD	2,14	0,27	97,59	10,14	1,10	88,77	12,03	90,00
PD	8,59	0,52	90,89	12,32	0,86	86,82	19,79	93,83
VE	53,19	1,89	44,92	5,79	3,68	90,53	55,79	94,02
SO	46,15	4,81	49,04	29,41	3,92	66,67	60,58	90,00
IO	55,11	0,57	44,32	0,00	0,00	100,00	55,11	98,98
FS	49,09	5,87	45,05	8,20	4,67	87,13	52,78	89,33
WI	**100,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**100,00**	100,00
WC	93,33	0,70	5,96	47,06	11,76	41,18	96,14	98,56
IS	81,00	1,62	17,38	30,68	13,97	55,34	86,33	95,52
IR	93,33	1,33	5,33	75,00	0,00	25,00	97,33	98,65

Tabela A.12: Desempenho da cascata de dois níveis com SVM no primeiro nível e conjunto de classificadores KNN gerado com Boosting no segundo nível

Base	1º nível			2º nível			Cascata	
	acertos	erros	rejeições	acertos	erros	rejeições	acertos	REL
LD	5,78	1,16	93,06	0,00	0,00	100,00	5,78	83,33
HB	**9,15**	**1,31**	**89,54**	**10,22**	**2,92**	**86,86**	**18,30**	82,35
BD	2,14	0,27	97,59	0,00	0,00	100,00	2,14	88,89
PD	8,59	0,52	90,89	0,00	0,00	100,00	8,59	94,29
VE	53,19	1,89	44,92	0,00	0,00	100,00	53,19	96,57
SO	**46,15**	**4,81**	**49,04**	**82,35**	**17,65**	**0,00**	**86,54**	86,54
IO	55,11	0,57	44,32	24,36	15,38	60,26	65,91	89,92
FS	**49,09**	**5,87**	**45,05**	**8,68**	**15,85**	**75,46**	**53,00**	**80,29**
WI	**100,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**100,00**	100,00
WC	93,33	0,70	5,96	17,65	17,65	64,71	94,39	98,18
IS	**81,00**	**1,62**	**17,38**	**65,75**	**34,25**	**0,00**	**92,43**	92,43
IR	93,33	1,33	5,33	75,00	25,00	0,00	97,33	97,33

Tabela A.13: Desempenho da cascata de dois níveis com SVM no primeiro nível e conjunto de classificadores KNN gerado com RSS no segundo nível

Base	1º nível			2º nível			Cascata	
	acertos	erros	rejeições	acertos	erros	rejeições	acertos	REL
LD	5,78	1,16	93,06	3,73	0,62	95,65	9,25	84,21
HB	9,15	1,31	89,54	4,38	1,46	94,16	13,07	83,33
BD	2,14	0,27	97,59	11,51	1,10	87,40	13,37	90,91
PD	8,59	0,52	90,89	15,19	0,86	83,95	22,40	94,51
VE	53,19	1,89	44,92	1,58	1,05	97,37	53,90	95,80
SO	46,15	4,81	49,04	58,82	1,96	39,22	75,00	92,86
IO	55,11	0,57	44,32	16,67	1,28	82,05	62,50	98,21
FS	49,09	5,87	45,05	8,56	4,70	86,73	52,95	86,90
WI	**100,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**100,00**	100,00
WC	93,33	0,70	5,96	47,06	11,76	41,18	96,14	98,56
IS	81,00	1,62	17,38	44,38	4,93	50,68	88,71	97,28
IR	93,33	1,33	5,33	25,00	25,00	50,00	94,67	97,26

Tabela A.14: Desempenho da cascata de dois níveis com SVM no primeiro nível e conjunto de classificadores SVM gerado com Bagging no segundo nível

Base	1º nível			2º nível			Cascata	
	acertos	erros	rejeições	acertos	erros	rejeições	acertos	REL
LD	5,78	1,16	93,06	10,56	4,35	85,09	15,61	75,00
HB	9,15	1,31	89,54	5,84	0,00	94,16	14,38	91,67
BD	2,14	0,27	97,59	10,68	1,37	87,95	12,57	88,68
PD	**8,59**	**0,52**	**90,89**	**21,20**	**1,72**	**77,08**	**27,86**	93,04
VE	53,19	1,89	44,92	8,95	2,11	88,95	57,21	95,28
SO	46,15	4,81	49,04	50,98	9,80	39,22	71,15	88,10
IO	55,11	0,57	44,32	55,13	3,85	41,03	79,55	97,22
FS	49,09	5,87	45,05	1,38	0,79	97,83	49,71	88,87
WI	**100,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**100,00**	100,00
WC	**93,33**	**0,70**	**5,96**	**64,71**	**11,76**	**23,53**	**97,19**	98,58
IS	81,00	1,62	17,38	5,48	4,38	90,14	81,95	97,18
IR	93,33	1,33	5,33	25,00	75,00	0,00	94,67	94,67

Tabela A.15: Desempenho da cascata de dois níveis com SVM no primeiro nível e conjunto de classificadores SVM gerado com Boosting no segundo nível

Base	1º nível			2º nível			Cascata	
	acertos	erros	rejeições	acertos	erros	rejeições	acertos	REL
LD	5,78	1,16	93,06	0,00	0,00	100,00	5,78	83,33
HB	9,15	1,31	89,54	7,30	3,65	89,05	15,69	77,42
BD	2,14	0,27	97,59	2,74	0,82	96,44	4,81	81,82
PD	8,59	0,52	90,89	4,01	0,57	95,42	12,24	92,16
VE	**53,19**	**1,89**	**44,92**	**14,21**	**10,53**	**75,26**	**59,57**	90,00
SO	46,15	4,81	49,04	0,00	0,00	100,00	46,15	90,57
IO	55,11	0,57	44,32	0,00	0,00	100,00	55,11	98,98
FS	49,09	5,87	45,05	0,00	0,00	100,00	49,09	89,33
WI	**100,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**100,00**	100,00
WC	93,33	0,70	5,96	35,29	17,65	47,06	95,44	98,19
IS	81,00	1,62	17,38	7,12	3,56	89,32	82,24	97,35
IR	93,33	1,33	5,33	75,00	25,00	0,00	97,33	97,33

Tabela A.16: Desempenho da cascata de dois níveis com SVM no primeiro nível e conjunto de classificadores SVM gerado com RSS no segundo nível

Base	1º nível			2º nível			Cascata	
	acertos	erros	rejeições	acertos	erros	rejeições	acertos	REL
LD	5,78	1,16	93,06	12,42	4,35	83,23	17,34	76,92
HB	9,15	1,31	89,54	8,76	2,19	89,05	16,99	83,87
BD	**2,14**	**0,27**	**97,59**	**35,34**	**3,84**	**60,82**	**36,63**	90,13
PD	8,59	0,52	90,89	16,33	0,86	82,81	23,44	94,74
VE	53,19	1,89	44,92	3,16	1,58	95,26	54,61	95,45
SO	46,15	4,81	49,04	37,25	5,88	56,86	64,42	89,33
IO	**55,11**	**0,57**	**44,32**	**70,51**	**5,13**	**24,36**	**86,36**	96,82
FS	49,09	5,87	45,05	4,05	2,28	93,67	50,91	88,07
WI	**100,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**100,00**	100,00
WC	93,33	0,70	5,96	29,41	0,00	70,59	95,09	99,27
IS	81,00	1,62	17,38	16,71	5,21	78,08	83,90	97,08
IR	93,33	1,33	5,33	25,00	25,00	50,00	94,67	97,26

Tabela A.17: Desempenho da cascata de dois níveis com SVM no primeiro nível e conjunto de classificadores SVM gerado com BoostingW* no segundo nível

Base	1º nível			2º nível			Cascata	
	acertos	erros	rejeições	acertos	erros	rejeições	acertos	REL
LD	5,78	1,16	93,06	0,00	0,00	100,00	5,78	83,33
HB	9,15	1,31	89,54	0,00	0,00	100,00	9,15	87,50
BD	2,14	0,27	97,59	1,64	0,00	98,36	3,74	93,33
PD	8,59	0,52	90,89	0,00	0,00	100,00	8,59	94,29
VE	53,19	1,89	44,92	10,00	4,74	85,26	57,68	93,49
SO	46,15	4,81	49,04	0,00	0,00	100,00	46,15	90,57
IO	55,11	0,57	44,32	0,00	0,00	100,00	55,11	98,98
FS	49,09	5,87	45,05	0,00	0,00	100,00	49,09	89,33
WI	**100,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**100,00**	100,00
WC	93,33	0,70	5,96	41,18	11,76	47,06	95,79	98,56
IS	81,00	1,62	17,38	0,00	0,00	100,00	81,00	98,04
IR	**93,33**	**1,33**	**5,33**	**100,00**	**0,00**	**0,00**	**98,67**	98,67

Nota: variação da técnica Boosting adotando peso diferenciado para as amostras rejeitadas na validação cruzada.

Tabela A.18: Desempenho da cascata de dois níveis com SVM no primeiro nível e seleção dinâmica de classificadores DS-OLA gerado com Bagging no segundo nível

Base	1º nível			2º nível			Cascata	
	acertos	erros	rejeições	acertos	erros	rejeições	acertos	REL
LD	**5,78**	**1,16**	**93,06**	**13,66**	**8,70**	**77,64**	**18,50**	66,67
HB	9,15	1,31	89,54	5,11	0,73	94,16	13,73	87,50
BD	2,14	0,27	97,59	6,30	0,55	93,15	8,29	91,18
PD	8,59	0,52	90,89	0,00	0,00	100,00	8,59	94,29
VE	53,19	1,89	44,92	0,00	0,00	100,00	53,19	96,57
SO	46,15	4,81	49,04	0,00	0,00	100,00	46,15	90,57
IO	55,11	0,57	44,32	0,00	0,00	100,00	55,11	98,98
FS	49,09	5,87	45,05	0,00	0,00	100,00	49,09	89,33
WI	**100,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**100,00**	100,00
WC	93,33	0,70	5,96	47,06	11,76	41,18	96,14	98,56
IS	81,00	1,62	17,38	0,00	0,00	100,00	81,00	98,04
IR	93,33	1,33	5,33	25,00	75,00	0,00	94,67	94,67

Tabela A.19: Desempenho da cascata de dois níveis com SVM no primeiro nível e seleção dinâmica de classificadores DS-LCA gerado com Bagging no segundo nível

Base	1º nível			2º nível			Cascata	
	acertos	erros	rejeições	acertos	erros	rejeições	acertos	REL
LD	5,78	1,16	93,06	4,35	2,48	93,17	9,83	73,91
HB	9,15	1,31	89,54	5,11	0,73	94,16	13,73	87,50
BD	2,14	0,27	97,59	6,85	0,82	92,33	8,82	89,19
PD	8,59	0,52	90,89	0,00	0,00	100,00	8,59	94,29
VE	53,19	1,89	44,92	0,00	0,00	100,00	53,19	96,57
SO	46,15	4,81	49,04	13,73	1,96	84,31	52,88	90,16
IO	55,11	0,57	44,32	0,00	0,00	100,00	55,11	98,98
FS	49,09	5,87	45,05	0,00	0,00	100,00	49,09	89,33
WI	**100,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**100,00**	100,00
WC	93,33	0,70	5,96	35,29	11,76	52,94	95,44	98,55
IS	81,00	1,62	17,38	14,25	15,34	70,41	83,48	95,12
IR	93,33	1,33	5,33	0,00	0,00	100,00	93,33	98,59

Tabela A.20: Desempenho da cascata de dois níveis com SVM no primeiro nível e seleção dinâmica de classificadores KNORA-Eliminate gerado com Bagging no segundo nível

Base	1º nível			2º nível			Cascata	
	acertos	erros	rejeições	acertos	erros	rejeições	acertos	REL
LD	5,78	1,16	93,06	1,86	0,62	97,52	7,51	81,25
HB	9,15	1,31	89,54	5,11	0,73	94,16	13,73	87,50
BD	2,14	0,27	97,59	10,68	1,10	88,22	12,57	90,38
PD	8,59	0,52	90,89	0,00	0,00	100,00	8,59	94,29
VE	53,19	1,89	44,92	0,00	0,00	100,00	53,19	96,57
SO	46,15	4,81	49,04	3,92	0,00	96,08	48,08	90,91
IO	55,11	0,57	44,32	0,00	0,00	100,00	55,11	98,98
FS	49,09	5,87	45,05	0,00	0,00	100,00	49,09	89,33
WI	**100,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**100,00**	100,00
WC	93,33	0,70	5,96	58,82	11,76	29,41	96,84	98,57
IS	81,00	1,62	17,38	0,55	1,10	98,36	81,10	97,82
IR	93,33	1,33	5,33	75,00	25,00	0,00	97,33	97,33

Tabela A.21: Desempenho da cascata de dois níveis com SVM no primeiro nível e seleção dinâmica de classificadores KNORA-Union gerado com Bagging no segundo nível

Base	1º nível			2º nível			Cascata	
	acertos	erros	rejeições	acertos	erros	rejeições	acertos	REL
LD	5,78	1,16	93,06	9,94	3,73	86,34	15,03	76,47
HB	9,15	1,31	89,54	8,03	1,46	90,51	16,34	86,21
BD	2,14	0,27	97,59	9,59	1,10	89,32	11,50	89,58
PD	8,59	0,52	90,89	11,75	1,15	87,11	19,27	92,50
VE	53,19	1,89	44,92	0,00	0,00	100,00	53,19	96,57
SO	46,15	4,81	49,04	0,00	0,00	100,00	46,15	90,57
IO	55,11	0,57	44,32	0,00	0,00	100,00	55,11	98,98
FS	49,09	5,87	45,05	0,00	0,00	100,00	49,09	89,33
WI	**100,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**100,00**	100,00
WC	93,33	0,70	5,96	52,94	11,76	35,29	96,49	98,57
IS	81,00	1,62	17,38	6,30	3,29	90,41	82,10	97,40
IR	93,33	1,33	5,33	75,00	25,00	0,00	97,33	97,33