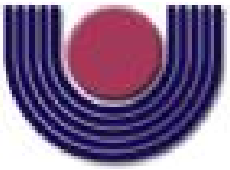

Unioeste - Universidade Estadual do Oeste do Paraná
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
Colegiado de Ciência da Computação
Curso de Bacharelado em Ciência da Computação

Estratégias de combinação em um sistema de múltiplos classificadores heterogêneos

João Victor Canabarro

CASCAVEL
2018

João Victor Canabarro

Estratégias de combinação em um sistema de múltiplos classificadores heterogêneos

Monografia apresentada como requisito parcial para obtendo do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel

Orientador: Prof. André Luiz Brun

CASCAVEL
2018

João Victor Canabarro

Estratégias de combinação em um sistema de múltiplos classificadores heterogêneos

Monografia apresentada como requisito parcial para obtendo do Título de Bacharel em Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Prof. André Luiz Brun
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Adair Santa Catarina
Colegiado de Ciência da Computação,
UNIOESTE

Prof[a]. Cláudia Brandelero Rizzi
Colegiado de Ciência da Computação,
UNIOESTE

Cascavel, 3 de Dezembro de 2018

“We are just an advanced breed of monkeys on a minor planet of a very average star. But we can understand the Universe. That makes us something very special.” — Stephen Hawking

AGRADECIMENTOS

Gostaria de agradecer a Jociara Canabarro, minha mãe, Siro Canabaro, meu pai, Murilo Canabarro, meu irmão, pelo apoio em todas as decisões que tomei e por estarem sempre ao meu lado, pois sem eles nada disso teria sido possível. Agradeço também ao professor André Luiz Brun pela orientação, pela paciência e pelos ensinamentos durante o desenvolvimento dessa monografia.

Agradeço em especial a Eduarda Moraes por ser meu porto seguro e se manter sempre ao meu lado no decorrer de toda a minha gradução e desenvolvimento deste trabalho, por ser meu incentivo por este percurso, servindo como inspiração. Agradeço aos meus avós por todo o carinho e suporte.

Deixo meus agradecimentos também a todos os amigos e colegas que conheci ao longo do período da graduação que de alguma forma contribuíram com meu crescimento.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

DT	Árvore de Decisão (*Decision Tree*)
KNN	K-Vizinhos Mais Próximos (*K-Nearest Neighbors*)
MLP	Perceptron de Múltiplas Camadas (*Multi Layer Perceptron*)
NB	Naive Bayes
OVA	Um contra Todos (*One-versus-All*)
OVO	Um contra Um (*One-versus-One*)
RBF	Função de Base Radial (*Radial Base Funcion*)
SB	*Single Best*
SMC	Sistema de Múltiplos Classificadores
SVM	Máquina de Suporte Vetorial (*Support Vector Machine*)

Conteúdo

Resumo

A classificação é uma sub-área do reconhecimento de padrões que objetiva atribuir classes a determinadas amostras dentro de categorias ou classes adotando para tal, as características do objeto. No entanto, a utilização de apenas um modelo de classificação pode não ser suficiente para cobrir toda a variabilidade do problema. Para isso foram propostos os sistemas de múltiplos classificadores (SMC), que utilizam regras de combinação na tentativa de aumentar a acurácia. Este trabalho teve como objetivo desenvolver e analisar estratégias de combinação em um SMC, considerando a acurácia do sistema. A avaliação é constituída de um SMC contendo 5 classificadores heterogêneos (KNN, SVM, DT, NB, MLP) e 9 técnicas de combinação (Borda Count, Média, Mediana, Ranking, Soma, Produto, Mínimo, Máximo e Voto Majoritário). A validação deu-se sobre um conjunto composto por 30 bases de dados provenientes de repositórios públicos, as quais foram submetidas à divisão estratificada. Os classificadores foram treinados sobre subconjuntos aleatórios gerados pelo *Bagging* com proporções de 20% e 50%. Seus parâmetros foram definidos através da estratégia de combinação por força bruta. Todo o processo é submetido a um protocolo robusto composto de 20 repetições onde analisou-se os valores médios obtidos através de testes estatísticos com 95% de confiança. Os resultados indicaram que a Soma e a Média foram as melhores estratégias de combinação para os dois tamanhos do *Bagging*, e que o Voto Majoritário obteve as piores acurácias médias. Observou-se também que o *Bagging* com 50% obteve os melhores resultados para grande parte das estratégias de combinação, devido ao fato de que o treinamento foi mais robusto e possibilitou maior variabilidade dos dados.

Palavras-chave: Classificação, Parâmetros, Estratégias de Combinação, Aprendizagem de Máquina.

Capítulo 1

Introdução

O reconhecimento de padrões é um ramo da inteligência computacional cujo objetivo é a classificação de objetos dentro de categorias ou classes que podem ser imagens, tipos de flores, sintomas de doenças ou qualquer outro objeto que possa ser caracterizado [Tan et al. 2009].

A classificação é uma das tarefas tratadas na área de reconhecimento de padrões. Seu objetivo é atribuir uma classe, dentre várias possíveis, a uma amostra cuja classe não é previamente conhecida (representada por um vetor de características). Para mapear uma classe, é necessário um algoritmo (modelo de classificação) capaz de gerar conclusões observado valores de entrada. Tendo um conjunto de amostras de dados, um modelo de classificação tentará prever o valor da classe para cada amostra.

Quando existe um problema de classificação muito complexo ou com dados faltantes, utilizar um único classificador (conhecido como monolítico) pode não ser suficiente para obter os resultados desejados. A utilização de um classificador monolítico capaz de cobrir toda a variabilidade do problema é inviável na maioria das vezes, principalmente quando o problema possui um número elevado de características.

A utilização de sistemas de reconhecimento de padrões baseados em sistemas de múltiplos classificadores (SMC) tem sido apresentada como alternativa para construir um sistema que consiga absorver toda, ou pelo menos a maior parte, da variabilidade de características das amostras. Esta ideia é fundamentada no princípio de que classificadores diferentes tendem a gerar erros com baixa correlação entre si em suas decisões. Neste contexto, cada classificador possui um peso em seu voto final, fazendo com que os elementos mais aptos a decidirem a classe do objeto tenham mais peso em sua decisão.

Ao se adotar um SMC espera-se que o desempenho alcançado seja maior em comparação

aos sistemas monolíticos, devido ao fato dele cobrir melhor a variabilidade das características da amostra, acarretando então em maiores taxas de acerto no momento da classificação [Ponti Jr. 2011].

No entanto, quanto mais se busca aumentar a acurácia da classificação, mais complexos os sistemas tendem a ficar. Esse aumento na complexidade em sistemas de múltiplos classificadores é fortemente criticado, principalmente quando o aumento da acurácia não é tão significativo quando comparado com a complexidade [Silva Jr. 2015].

Para o bom funcionamento de um SMC são necessárias bases de dados que sejam representativas, ou seja, que tenham características fiéis à realidade e que representem de forma suficiente as diferenças existentes entre elementos de classes distintas. Todavia, isso é difícil de ser encontrado, pois geralmente as bases de dados ou não contêm um número suficiente de amostras ou não exprimem com precisão o comportamento específico de cada grupo, o que dificulta um aprendizado mais completo e robusto do sistema de classificação.

Tais sistemas podem ser gerados de duas maneiras: a primeira, de forma heterogênea, consiste em utilizar diferentes algoritmos de aprendizado de máquina (indutores). A segunda forma é a estratégia homogênea, em que todos os classificadores utilizam o mesmo indutor, porém, eles sofrem variações nos conjuntos de treino. Uma abordagem alternativa é utilizar ambas as formas gerando um conjunto misto de classificadores [Vriesmann 2013].

Em cenários em que são utilizados sistemas compostos de um conjunto de classificadores, diversos autores propõem que, ao invés de simplesmente combinar a opinião de todos os classificadores do conjunto, pode-se utilizar alguma estratégia para escolher aquele classificador que parecer mais preparado para rotular as instâncias de entrada.

As estratégias de fusão mais comumente utilizadas são: combinação linear, que consiste em combinar linearmente todos os resultados dos classificadores, seja na forma de soma [Kittler et al. 1998], ou média [Kuncheva e Whitaker 2003]; combinação hierárquica: onde o resultado necessita percorrer uma sequência definida de classificadores, ou seja, ao entrar no primeiro classificador os dados são analisados e a saída será a entrada do segundo classificador, até chegar ao último classificador que tem o resultado final [Ranawana e Palade 2006].

O objetivo deste trabalho consistiu em desenvolver e analisar estratégias de combinação (Borda Count, Média, Mediana, Ranking, Soma, Produto, Mínimo, Máximo e Voto Majoritário)

em um sistema de múltiplos classificadores heterogêneo, para determinar sua acurácias.

No Capítulo 2 é apresentada uma revisão da literatura sobre sistemas de classificação, classificadores, topologias de SMCs e estratégias de combinação das predições. A metodologia implementada é detalhada no Capítulo 3, onde é descrito, passo a passo, o funcionamento do sistema de classificação proposto. O Capítulo 4 apresenta uma descrição aprofundada da etapa de experimentos, relacionados a acurácia e escolha dos melhores parâmetros de cada classificador, bem como a análise dos resultados obtidos. O último Capítulo deste trabalho apresenta as conclusões alcançadas e sugestões de novos trabalhos.

Capítulo 2

Revisão

Este capítulo apresenta uma revisão da literatura sobre sistema de classificação e reconhecimento de padrões de forma a embasar os métodos empregados na pesquisa. Os tópicos seguintes irão abordar o que é o reconhecimento de padrões e o que são classificadores monolíticos. Em seguida, aborda-se os SMC's, tratando da diversidade, seleção e combinação de múltiplos classificadores. Além disso, será apresentada a fundamentação teórica de cada classificador.

2.1 Reconhecimento de padrões

O reconhecimento de padrões é uma área da inteligência computacional cujo objetivo é a classificação de objetos em categorias ou classes que podem ser imagens, tipos de flores, diagnósticos de doenças ou qualquer outro objeto que possa ser categorizado.

Alguns conceitos básicos dentro do reconhecimento de padrões são pertinentes e merecem destaque:

- **Amostra**: são os exemplos ou instâncias de uma base de dados.
- **Atributos**: também conhecidos por vetor de características, os atributos representam os atributos que compõem uma amostra.
- **Classe**: conjunto de amostras que possuem características em comum.
- **Classificação**: é a ação de atribuir uma classe para uma amostra com base na semelhança entre os seus atributos.
- ***Pool*:** subconjunto de classificadores homogêneos ou heterogêneos.

Um sistema de reconhecimento de padrões pode ser dividido em seis etapas [Silva Jr. 2015]:

1. **Aquisição:** Essa etapa tem como função a conversão de um conjunto de características de uma determinada amostra que se apresenta no conjunto contínuo para o conjunto discreto. Ou seja, estimar valor para cada característica de modo que o sistema de reconhecimento de padrões a possa interpretar.

2. **Pré-Processamento:** A amostra resultante do passo anterior pode apresentar diversas imperfeições, como ruídos, dados faltantes acarretados por uma série de problemas, como falhas no momento da aferição dos dados ou erros de digitação. A etapa de pré-processamento é utilizada para aprimorar a qualidade da amostra, de modo que ela não tenha qualquer um dos problemas relatados acima e que torne os passos seguintes mais efetivos.

3. **Segmentação:** O objetivo da etapa de segmentação de amostras é dividi-las de modo a agrupar todas as informações que podem ser relevantes para o sistema, e quais podem ser descartas por não contribuírem para o reconhecimento ou mesmo atrapalharem o processo.

4. **Extração de Características:** Este passo faz com que seja possível separar e levantar as características da amostra de entrada, de modo que possamos ter os valores definidos para cada uma. Nessa etapa cada amostra já deve ser transformada para se adequar com a estrutura do sistema de reconhecimento, fazendo com que ela já esteja de acordo com as demais amostras, ou seja, seguindo uma padronização definida.

5. **Classificação:** Logo após a extração das características da amostra, todas elas servem de entrada a um classificador para que o mesmo consiga estimar, segundo algum critério, qual é a classe que mais se aproxima da amostra.

6. **Decisão Final:** A decisão final analisa o que foi avaliado pelo classificador e apresenta o resultado da classificação da amostra. Nos casos para múltiplos classificadores podemos combinar os resultados de suas decisões utilizando estratégias de combinações para tentar atingir resultados mais acurados.

O aprendizado de máquina pode ser separado em duas abordagens de aprendizagem, o supervisionado e não supervisionado [Jain, Duin e Mao 2000]. A distinção entre um tipo e o outro é que no caso supervisionado para toda amostra de entrada já se sabe previamente qual é classe a qual ela pertence, então dessa forma é possível comparar se o resultado gerado pelo sistema de classificação é preciso. Já no não supervisionado, não se conhece (ou se omite) a classe à qual as instâncias pertencem [Vriesmann 2013].

Neste trabalho aplicou-se a abordagem supervisionada, utilizando um conjunto de bases de dados bastante empregadas na literatura, para treinar os classificadores de forma que eles sejam capazes de posteriormente rotular as amostras desconhecidas.

2.1.1 Identificação de classes

A classificação é uma das tarefas tratadas na área de reconhecimento de padrões. Seu objetivo é atribuir uma classe, dentre várias possíveis, a um objeto de teste (representada por um vetor de características). Para mapear uma classe, é necessário ter um modelo capaz de interpretar tais características e determinar a classe da amostra.

2.1.2 Treinamento

Um conjunto de dados costuma ser separado em dois grupos principais, treinamento e testes. O conjunto de treino é utilizado para a calibração do classificador. Ou seja, é responsável por definir os melhores conjunto de valores para o classificador na tentativa de melhorar a acurácia. Na etapa de treino pode ser utilizado o *Bagging*, conhecido como um meta-classificador de conjunto, empregado na geração de subconjuntos retirando amostras aleatórias da base de dados com reposição.

O *Bagging* é utilizado para reduzir a variância de um classificador, introduzindo a randomização do procedimento de escolha do subconjunto de treino. Em muitos casos, o método apresenta-se como uma solução simples de melhorar a divisão do conjunto de dados, sem que seja necessário adaptar algoritmos específicos para cada base de entrada.

A Figura 2.1 demonstra como o *Bagging* faz a divisão da base em subconjuntos e envia cada um deles para o treinamento dos classificadores do *pool*. Na representação, verifica-se que o conjunto de treino serve de fonte para a escolha das instâncias que formarão cada um dos N

novos subconjuntos.

Podemos observar também que o algoritmo sempre tenta manter ao máximo as proporções de classes das amostras na base de dados em relação às amostras presentes no subconjunto, para que não incorra em tendências para qualquer uma das classes.

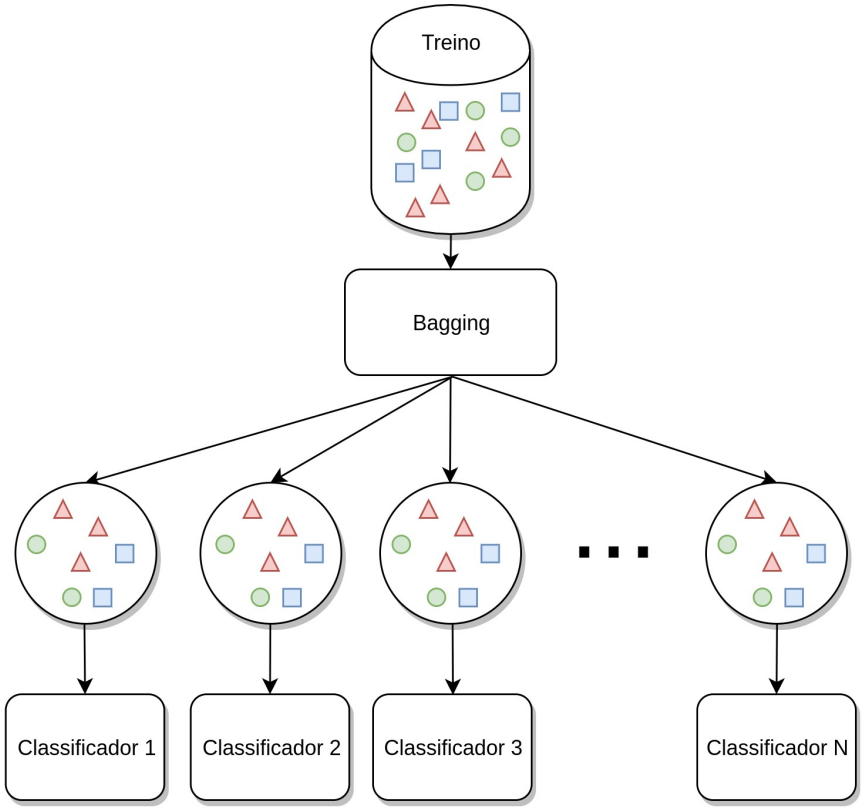

Figura 2.1: Representação gráfica do Bagging

É importante destacar que os conjuntos de teste e treino devem ser mutuamente exclusivos, de modo que a intersecção entre os mesmos seja sempre vazia. A presença de amostras semelhantes em ambos os conjuntos, pode fazer com que o resultado se torne tendencioso.

2.2 Métodos de Classificação

Os métodos de classificação são os algoritmos que, dadas as características da amostra a ser analisada, estimarão uma porcentagem de confiança da classe a que ela pertence, apresentando aquela com maior percentual de certeza.

As próximas subseções apresentam os métodos de classificação mais utilizados em pesquisas na área de reconhecimento de padrões, os quais serão avaliados neste trabalho. Sendo eles:

Máquina de Vetores de Suporte (SVM, *Support Vector Machine*), K-Vizinhos Mais Próximos (KNN, *K-Nearest Neighbors*), Naive Bayes (NB), Árvore de Decisão (DT, *Decision Tree*) e Perceptron de Múltiplas Camadas (MLP, *Multilayer Perceptron*).

A escolha por estes métodos de aprendizado orientou-se no princípio de construir um conjunto de classificadores robusto em que cada técnica é baseada em uma estratégia diferente. O KNN é baseado em instâncias, o SVM em um separador ótimo, a MLP baseado em redes neurais, a DT em regras do tipo se-então-senão e o NB em um aprendizado probabilístico.

2.2.1 Máquinas de Vetores de Suporte

O método de Máquinas de Vetores de Suporte (SVM, *Support Vector Machine*) é uma técnica de classificação supervisionada, que não possui armazenamento da classe de treino para classificar novas amostras, ou seja, é um classificador não paramétrico. Além disso, o SVM é um classificador binário (não consegue identificar um conjunto composto por mais de duas classes).

O SVM baseia-se na construção de hiperplanos para determinar sua decisão. Para tanto, ele maximiza a margem de separação entre as amostras de testes positivas (que pertencem a determinada classe) e negativas (não pertencem à classe).

Caso o problema não seja linearmente separável, então é necessário mapear o conjunto de treinamento original para um espaço de maior dimensão, designando um novo espaço de características onde o problema é linear. Para isso é preciso encontrar uma transformação não linear, chamada *Kernel Trick*.

Para determinar a transformação de espaço são utilizadas funções de *Kernel* [Hsu, Chang e Lin 2010], sendo elas: Linear, Polinomial, Função de Base Radial e Sigmóide.

Segundo Hsu, Chang e Lin (2010) o *Kernel* linear é inferior quando comparado com a Função de Base Radial (RBF, *Radial Basis Function*), pois a última consegue tratar casos em que a relação entre as características das amostras não são linearmente separáveis. Além disso, a RBF possui uma demanda computacional similar aos *kernels* linear e sigmóide, e gera um número menor de hiperplanos quando comparado com polinomial.

A Figura 2.2[1] mostra como o SVM é utilizado para fazer a diferenciação de duas classes, é

[1] https://www.jeremyjordan.me/support-vector-machines/

possível observar que quando os valores são dispostos em um gráfico de duas dimensões, não existe uma função linear capaz de separar todos os elementos das duas classes com precisão. No entanto, se utilizarmos uma transformação de espaço (neste caso de 2D para três dimensões) através de uma função de *kernel*, conseguimos determinar um plano suficientemente capaz de separar as duas classes.

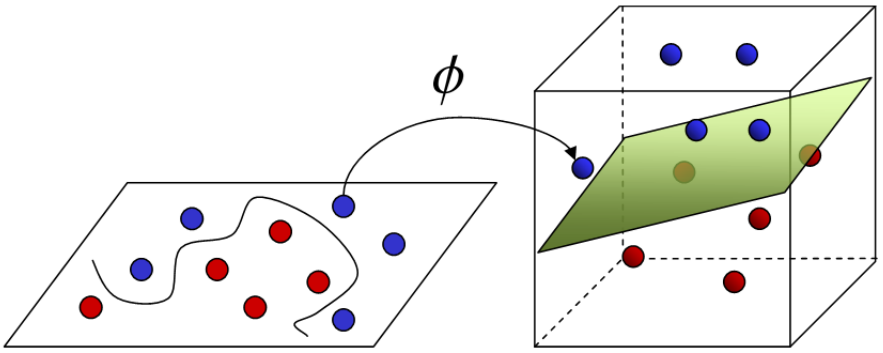

Figura 2.2: Representação gráfica de SVM

Como citado anteriormente, o SVM é um classificador binário, ou seja, faz a diferenciação apenas entre duas classes. Uma abordagem para resolução de problemas multi classes é considerar o problema como um conjunto de problemas de classificação binária [Weston e Watkins 1999].

Considere o exemplo da Figura 2.3, onde o problema possui três classes (representados por quadrados, triângulos e círculos), portanto, é necessário dividí-lo em subproblemas de modo que a separação das classes desses subproblemas seja binária.

Existem duas abordagens principais para o tratamento deste desafio. A primeira é conhecida por um-contra-todos (OVA, *One-versus-all*). A estratégia desta abordagem é ilustrada na Figura 2.4, onde é demonstrado como a separação das classes deve ser interpretada pelo SVM de modo que todos os subproblemas sejam binários. Para obter tal cenário, o método separa as instâncias de uma classe em um conjunto alvo e, todas as demais instâncias, pertencentes às outras classes são agrupadas em um único conjunto. Esse processo é realizado para cada uma das M classes presentes no problema original.

A Figura 2.3 ilustra um problema composto por três classes. Para transformá-lo em um problema binário, deve-se separar uma das classes de todas as outras, formando assim apenas

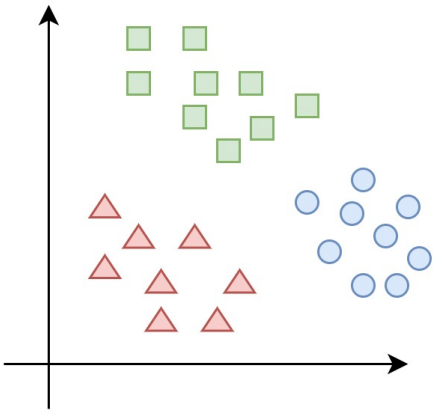

Figura 2.3: Problema hipotético multi classe

dois conjuntos. Na Figura 2.4 (a) a classe representada pelos triângulos é separada das demais (quadrados e círculos). O mesmo ocorre com os conjuntos representados pelas figuras 2.4 (b) e 2.4 (c) onde são separadas as classes quadrado e círculo, respectivamente.

Esta abordagem então treina um classificador sobre um subproblema binário, por exemplo, classe triângulo versus classe diferente de triângulo. Esse processo é executado para todas as M classes no problema original.

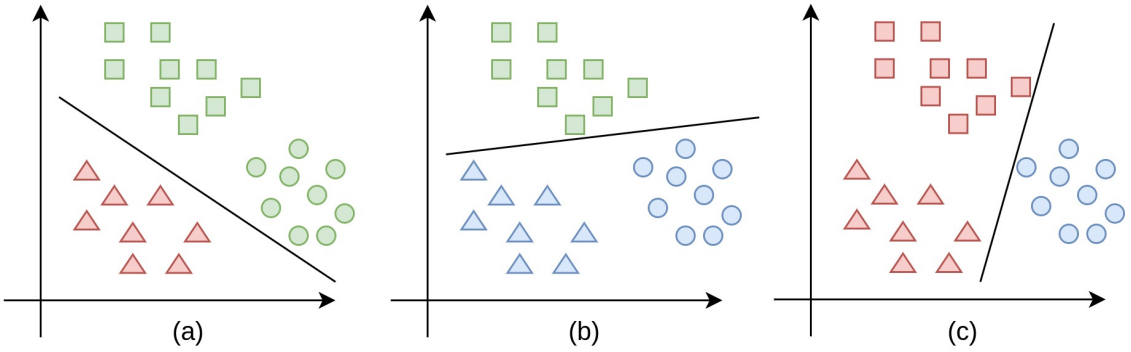

Figura 2.4: Separação utilizando um-contra-todos

A segunda abordagem para reconhecer problemas multi classes é o um-contra-um (OVO, *One-versus-One*) a qual, para treinar um problema com M classes, necessita um total de $\frac{M(M-1)}{2}$ classificadores.

A Figura 2.5 ilustra a separação das classes representadas na Figura 2.3 de modo que no conjunto existam apenas duas. As demais classes serão desconsideradas, gerando assim todas as combinações possíveis. Na Figura 2.5 (a) a classe círculo é omitida, restando ao problema,

apenas a separação das classes triângulo e quadrado. Esse mesmo processo é repetido até que todas as combinações de pares de classes sejam geradas (Figura 2.5 (b) e (c)).

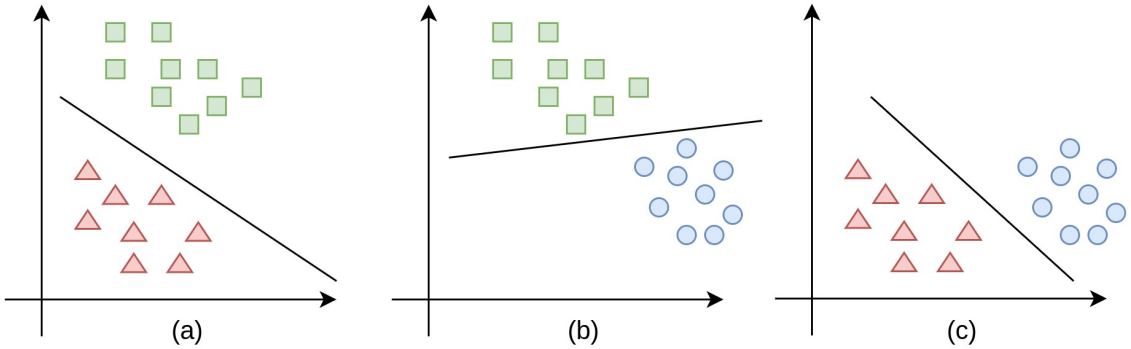

Figura 2.5: Separação utilizando um-contra-um

A abordagem OVO consegue ser menos sensível a problemas dos conjuntos de dados desbalanceados, que ocorre quando as classes das amostras do conjunto de dados não mantém proporcionalidades aproximadas. Quando comparados com a forma OVA, a estratégia OVO realiza um número maior de operações, tornando a abordagem computacionalmente mais cara.

2.2.2 K Vizinhos Mais Próximos

O aprendizado do método de K-Vizinhos Mais Próximos (KNN, *K-Nearest Neighbors*) consiste em armazenar todas as amostras do conjunto de treinamento. Quando uma nova amostra precisa ser classificada, o algoritmo recupera os K casos no conjunto de treinamento mais similares à nova amostra e, com base nos rótulos dos exemplos recuperados, realiza a classificação. Essa correlação é obtida através de uma medida de distância [Mitchell 1997].

A estratégia do KNN tem como vantagem a simplicidade na etapa de treinamento, consistindo apenas em armazenar todos casos do conjunto de teste. A desvantagem apresenta-se na hora de classificação, por ter que examinar todos os dados de treinamento a cada nova classificação para identificar quais são os mais parecidos ao novo padrão.

A abordagem mais utilizada para o cálculo da distância entre a nova amostra e os casos armazenados é a distância Euclidiana, que é uma das métricas mais simples [Mitchell 1997].

No entanto, não é tão simples encontrar o valor ótimo de k, pois cada base de dados pode apresentar casos particulares, sendo muito complexo definir um valor único de k para qualquer

base de entrada com resultados satisfatórios. Existem diversas abordagens para encontrar o valor mais adequado de k, de modo a alcançar o melhor resultado possível [Hassanat et al. 2014].

Como a etapa de treino do classificador consiste apenas em armazenar todas as instâncias, na etapa de teste é necessário então calcular a distância para todos as amostras em que o classificador já conhece a classe, e então selecionar os k casos mais próximos de modo a definir qual é a classe da nova amostra. Podemos observar que na Figura 2.6 existem duas classes (círculos e triângulos) e uma nova instância (quadrado) deve ser classificada.

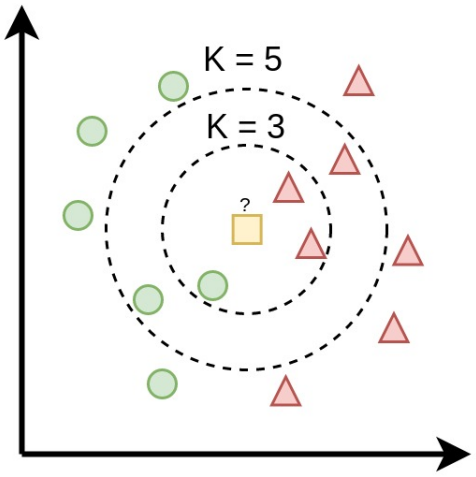

Figura 2.6: Representação gráfica de um KNN

Em sua grande maioria o valor pode ser obtido de forma empírica para cada base de dados. Um dos pontos pertinentes a ser destacado é que pelo fato do classificador analisar os casos mais semelhantes, com o caso atual, o número de k não deve ser par, para que se diminua a chance de empate entre os resultados.

2.2.3 Naive Bayes

O algoritmo de Naive Bayes (NB) classifica uma amostra baseado na máxima probabilidade a *posteriori* considerando o conjunto de características da própria instância. A probabilidade $P(A|\vec{B})$ da classe A, dado um vetor de atributos $\vec{B}$ de uma amostra qualquer, é determinada utilizando o teorema de Bayes.

A Equação 2.1 mostra como é expresso matematicamente este teorema:

$$P(A|B) = P(A)\frac{P(B|A)}{P(B)} \tag{2.1}$$

1. $P(A)$ e $P(B)$ são probabilidade a *priori* dos eventos A e B, e $P(B) \neq 0$;

2. $P(A|B)$ é a probabilidade a *posteriori* de A condicional a B;

3. $P(B|A)$ é a probabilidade a *posteriori* de B condicional a A.

Segundo Michie, Spiegelhalter e Taylor (1994), o classificador é recomendado para conjuntos de treinamento grandes ou moderados, pois o mesmo considera que cada uma das características contribui independentemente da probabilidade conjunta e de possíveis correlações entre as características.

Considere um problema de diagnóstico médico onde existem duas classes possíveis, o paciente possui H1N1 e o paciente não possui H1N1. Portanto, o exame laboratorial capaz de inferir o resultado deve retornar positivo ou negativo.

Existem algumas probabilidades a *priori* já definidas com base nas pessoas que já realizaram o exame, sendo eles:

1. De toda a população, apenas 0.008% tem a doença;

2. O exame retorna resultado verdadeiro positivo em 98% dos casos em que a doença esta presente;

3. O exame retorna um resultado verdadeiro negativo em 97% dos casos nos quais a doença não está presente;

4. Todos os demais casos, o teste retorna o resultado oposto (falsos positivos e falsos negativos).

Supondo que um paciente fez um exame de laboratório e o resultado foi positivo, a chance do mesmo realmente possuir H1N1 é baseado em suas probabilidades *a priori*, portanto temos:

1. $P(Positivo|H1N1) \times P(H1N1) = 0.98 \times 0.008 = 0.0078$

2. $P(Positivo|!H1N1) \times P(!H1N1) = 0.03 \times 0.992 = 0.0298$

Dado que a maior probabilidade a *posteriori* foi obtida para o paciente não ter a doença, infere-se que o resultado do exame não está correto.

2.2.4 Árvore de Decisão

A Árvore de Decisão (DT, *Decision Tree*) C5.0 é um algoritmo que divide o espaço ou conjunto em características, de forma a encontrar aquele que tem maior capacidade de separação dos dados com base no valor da característica (o qual gera dois conjuntos com a menor entropia). Esse método apresenta sucesso quando a amostra possui características ruidosas, já que apenas os atributos mais relevantes são utilizados. No entanto, a Árvore de Decisão é sensível à variabilidade dos dados [Utgoff 1989].

A Figura 2.7 representa um exemplo da construção de uma árvore de decisão partindo do princípio de que uma pessoa irá jogar tênis, dependendo das condições climáticas.

Neste caso, podemos observar que cada nó da árvore representa uma característica das instâncias da base de dados, e que para avançar para os próximos nós da árvore, a amostra passa por um teste lógico que define se deve seguir para o nó mais a direita ou mais a esquerda, até chegar em uma folha, que na estrutura de uma DT, representa as possíveis classes que a amostra pode ser atribuída.

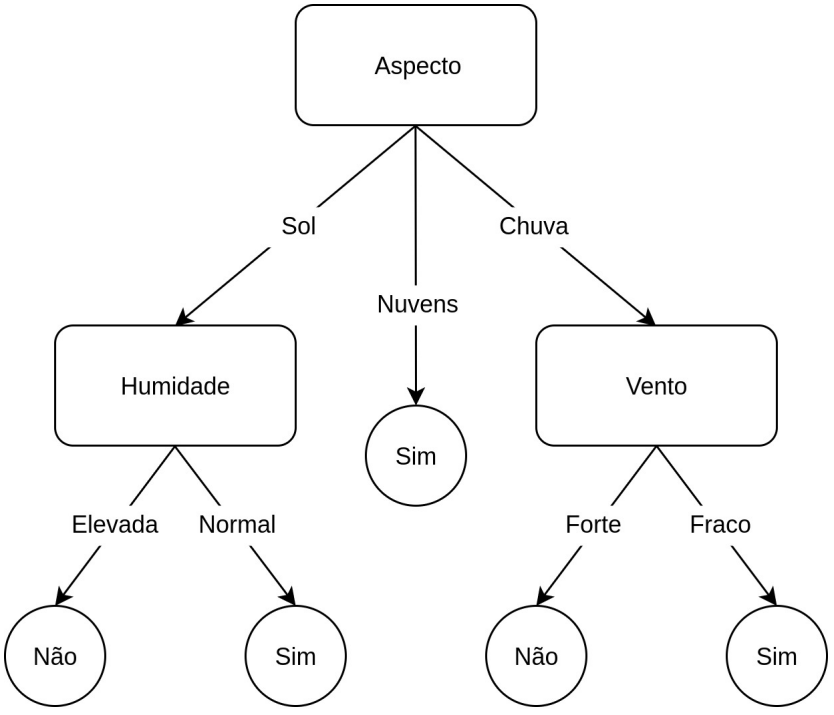

Figura 2.7: Representação gráfica de uma Árvore de Decisão

A estratégia de DT possui várias implementações distintas na literatura. A mais conhecida destas é a C4.5. No entanto, recentemente o autor desta estratégia propôs uma nova abordagem,

considerada evolução da antiga. Esta nova proposta é chamada C5.0.

Em relação a diferença entre a Árvore de Decisão C5.0 e C4.5 foram avaliados os seguintes parâmetros[2]:

- **Acurácia:** Quando comparada a taxa de erro entre os dois algoritmos, o C5.0 tende a ser sempre muito próximo ao C4.5.
- **Velocidade:** O algoritmo C5.0 é extremamente mais rápido na grande maioria dos casos.
- **Uso de Memória:** Neste item em questão o C5.0 também é mais eficiente pois utiliza muito menos memória em comparação ao C4.5. Em alguns experimentos realizados pelo autor, por exemplo, o C4.5 utilizou cerca de 3GB, enquanto o C5.0 demandou menos de 200MB.

2.2.5 Perceptron de Múltiplas Camadas

O algoritmo do Perceptron de Múltiplas Camadas (MLP, *Multilayer Perceptron*) é inspirado no sistema nervoso humano, onde as informações são processadas através de uma rede de neurônios [Bishop 1995]. O MLP é uma interconexão de neurônios que tem a sua propagação no sentido da entrada para a saída, passando por camadas intermediárias, forçando a geração de um modelo simples que seja suficientemente generalizável para os padrões desconhecidos [Michie, Spiegelhalter e Taylor 1994].

Para esse classificador como entrada são informados os valores do vetor de características e a saída é a classe da amostra. O resultado é a combinação ponderada das respostas de todas as camadas que as características passaram até chegar a última camada com o resultado da classe.

A Figura 2.8 representa a arquitetura de um MLP, onde os valores são inseridos na camada inicial passando pelas camadas escondidas até chegaram a camada final, e o valor da classe da amostra é apresentado, é importante ressaltar que quanto maior o número de camadas escondidas, maiores são as chances do aumento da precisão.

[2]Para observação mais completa dos resultados e dados utilizados, o link para acesso é: http://rulequest.com/see5-comparison.html

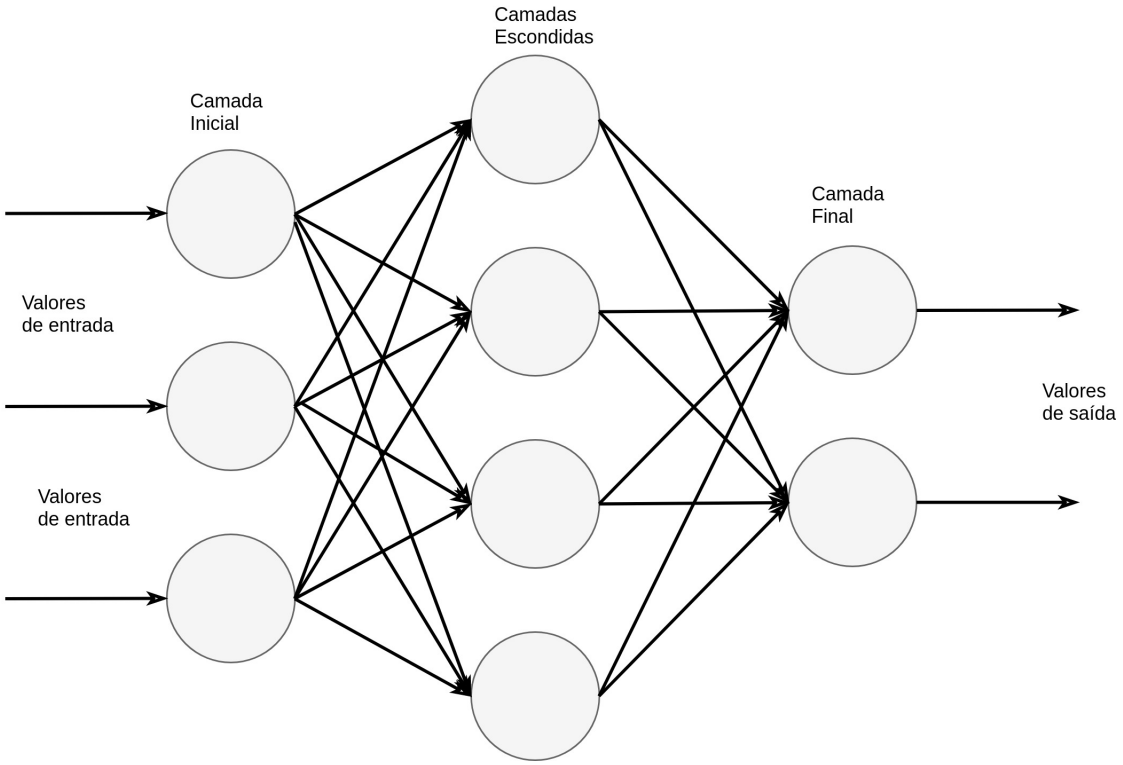

Figura 2.8: Representação de um MLP genérico

2.3 Sistemas de Classificação

A classificação é uma das tarefas tratadas na área de reconhecimento de padrões. No entanto, ela é apenas uma etapa dentre várias de um sistema completo. Neste sistema se objetiva atribuir uma classe, dentre as várias possíveis a um objeto. Porém, são necessários passos anteriores, como a coleta dos valores do atributos, a segmentação dos dados, a extração de características de cada amostra que formarão o vetor de características.

Seu objetivo é atribuir uma classe, dentre várias possíveis, a um objeto de teste (representada por um vetor de características). Para mapear uma classe, é necessário um modelo para a classificação, que pode ser individual ou baseado em múltiplos classificadores.

2.3.1 Sistema Monolítico

Para um sistema de classificação cada amostra é um elemento em um espaço que pode ter duas ou mais dimensões, as quais são delimitadas pelo número de atributos que descrevem a amostra. Para se realizar a classificação, divide-se o espaço atribuindo limites de decisão para

conseguir separar todos os dados. No entanto, essa separação geralmente não é linear, ou seja, é preciso definir uma função que consiga melhor representar a separação de todos os atributos da amostra.

A etapa seguinte à criação do classificador é o treinamento, que consiste em calibrar o modelo, fazendo com que seus parâmetros sejam variados de modo a encontrar uma combinação mais adequada ao problema proposto, uma vez que todas as amostras no conjunto de treino já possuem a sua classe mapeada *a priori*. Nesta etapa é sempre desejado que o sistema consiga ser o mais generalista possível para que, ao receber novas amostras, o classificador tenha facilidade para poder determinar a sua classe com maior precisão.

A adoção de um classificador individual, no entanto, pode não ser suficiente para absorver toda a variabilidade que um problema apresenta. Uma estratégia para tentar atender a tal demanda é empregar diversos classificadores em conjunto.

2.3.2 Sistema com Múltiplos Classificadores

Um Sistema de Múltiplos Classificadores (SMC), consiste em um conjunto de vários classificadores onde a amostra de entrada será analisada por todos os elementos da rede até que se consiga determinar a sua classe. Um SMC pode ser construído seguindo duas abordagens [Gilpin e Dunlavy 2009]. Na primeira, chamada homogênea, todos os classificadores possuem o mesmo tipo/ de aprendizado.

Esta proposta é representada pela Figura 2.9, em que são adotados vários classificadores similares, onde cada nova amostra a ser classificada é fornecida para um *pool* de n classificadores com o mesmo tipo de aprendizado gerando-se n resultados. Portanto, faz-se necessária a adoção de regras de decisão de modo a combinar todos esses resultados para que seja obtido um voto único ao final do processo, que irá definir a classe da amostra.

Visando obter certa robustez no sistema, cada classificador possui variações em seus parâmetros ou no conjunto sobre o qual é treinado com o objetivo de conseguir uma análise mais robusta.

A segunda abordagem de um SMC é a heterogênea, onde o sistema é composto por classificadores que possuem diferentes tipos de aprendizagem, divergindo da Figura 2.9 onde todos os classificadores possuem a mesma estratégia de aprendizado, a Figura 2.10 mostra o esquema de

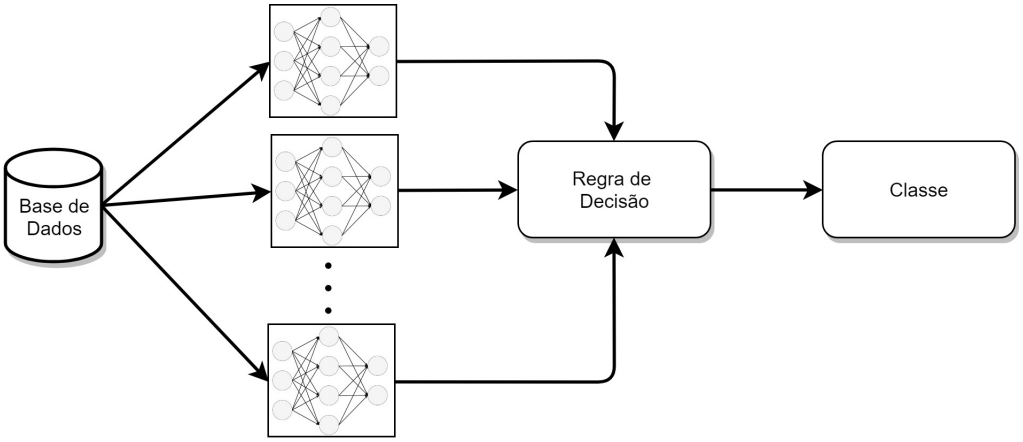

Figura 2.9: Estrutura genérica de um SMC Homogêneo

um SMC heterogêneo.

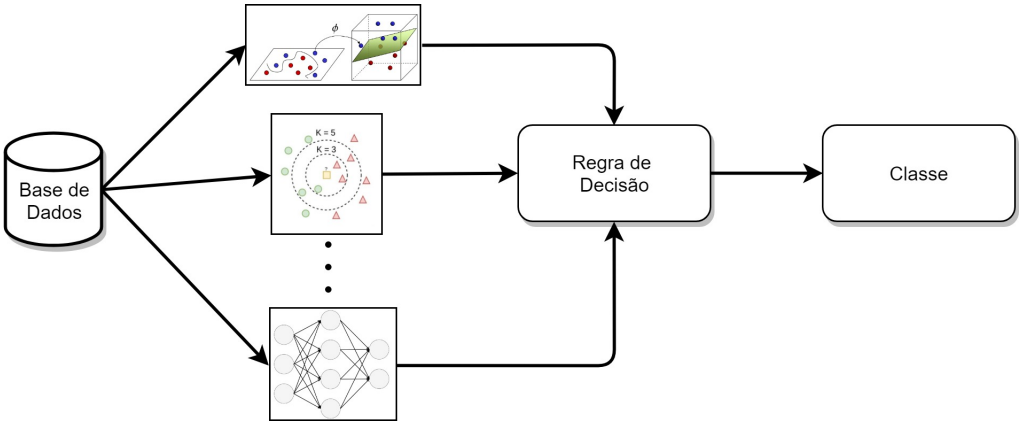

Figura 2.10: Estrutura genérica de um SMC Heterogêneo

A utilização de SMC tem sido defendida como uma alternativa quando comparada à difícil tarefa de construir um classificador monolıtico capaz de absorver toda a variabilidade de um problema de classificação. A justificativa 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 um *pool* de classificadores superar o desempenho em termos de acurácia de um sistema monolítico, uma vez que o primeiro pode cobrir melhor a variabilidade do problema.

2.3.3 Diversidade de Conjunto

Quando existe um problema de classificação muito complexo ou com falta de dados, utilizar um classificador monolítico pode não ser suficiente para obter os resultados desejados. A criação de um classificador monolítico capaz de cobrir toda a variabilidade do problema de entrada é uma tarefa inviável na maioria das vezes, principalmente quando o problema possui um número elevado de características [Silva Jr. 2015].

A utilização da diversidade em um conjunto de classificadores realiza um papel fundamental em um SMC, de maneira que o desempenho de um conjunto de classificadores diversos possa ser superior quando comparado com uma abordagem individual [Kuncheva e Whitaker 2003]. No entanto, dada uma determinada complexidade de interpretação da diversidade, ainda não existe um consenso sobre o seu grau de influência na acurácia dos métodos.

Alguns autores propõem métricas para calcular a relação entre acurácia do SMC e a sua diversidade de conjunto. Deve-se observar que as métricas permitem apenas a avaliação dos classificadores em pares, ou seja, devem ser geradas todas as possibilidades para conseguir estimar as métricas do sistema [Kuncheva e Whitaker 2003]. Podemos citar duas dessas métricas:

1. *Double Fault*: Define o percentual de instâncias em que os dois classificadores não conseguem acertar a classe da amostra. Representado pela primeira linha da Tabela 2.1.

2. *Disagreement*: Estima o percentual de instâncias em que os dois classificadores divergem suas opiniões, ou seja, cada classificador irá atribuir classes diferentes entre si. Um exemplo disso pode ser visto na segunda e terceira linhas da Tabela 2.1

Tabela 2.1: Base de Dados Hipotética

Amostras	Opinião do Classificador 1	Opinião do Classificador 2	Classe Real
Amostra 1	1	1	0
Amostra 2	1	0	1
Amostra 3	0	1	1
Amostra 4	0	0	0

A utilização de sistemas de reconhecimento de padrões baseados em SMCs tem sido apresentada como alternativa para construir um sistema que consiga absorver toda, ou pelo menos a

maior parte, da variabilidade presente em um problema de classificação. Esta ideia é fundamentada na premissa de que a presença de diversidade no conjunto consegue gerar erros com baixa correlação entre si, pois quando um conjunto de classificadores apresentam erros não correlatos, a combinação entre eles tem maiores chances de aumentar a performance do sistema quando comparada com classificadores que sempre geram erros com alta correlação [Ponti Jr. 2011].

Em seu trabalho, Santana *et al.*(2006) propuseram duas abordagens de seleção dinâmica de classificadores com base na acurácia e diversidade dos mesmos. Resultados mostraram que a adoção da diversidade tem peso significativo dentro da escolha do subconjunto de classificadores na formação dos *pools*.

No entanto, quanto mais se busca aumentar a acurácia da classificação, mais complexos os sistemas tendem a ficar. Esse aumento na complexidade em sistemas de múltiplos classificadores é fortemente criticado, principalmente quando o aumento da acurácia não é tão significativo quando comparado com a complexidade [Silva Jr. 2015].

2.3.4 Seleção de Classificadores

A seleção de classificadores pode ocorrer de duas formas: a estática e a dinâmica [Gunes et al. 2003]. Na primeira, todas as amostras de teste sempre serão rotuladas pelo mesmo subconjunto de classificadores, os quais são definidos antes da etapa de treinamento. Na abordagem dinâmica, o subconjunto de classificadores é selecionado com base nas características da amostra de teste. De forma que com a seleção dinâmica os classificadores selecionados são apenas aqueles que estão mais aptos para cada amostra classificada.

Nas duas abordagens, quando uma nova amostra entrar no sistema, os respectivos subconjuntos irão processar os dados para definição da classe da amostra. No entanto, como cada classificador irá gerar um resultado é necessário a adoção de uma combinação de resultados para definir como serão ponderadas as saídas dos classificadores.

A sessão a seguir descreve algumas abordagens de implementações de combinação de classificadores.

2.4 Combinação de Classificadores

A última etapa dentro de um processo de classificação é determinar a classe que a amostra pertence. Em um sistema com um classificador monolítico é necessário apenas utilizar o resultado retornado pelo classificador. No entanto, quando se trabalha com SMCs, cada classificador irá retornar a sua opinião para a classificação da amostra. Os resultados obtidos são então combinados para se encontrar a classificação final.

A combinação, segundo Lu (1996), Ranawana e Palde (2006) e Ponti Jr. (2011), pode ser executadas por três topologias diferentes, sendo elas: Paralela, Serial e Mista (Híbrida).

2.4.1 Topologia Paralela

Na topologia paralela, todos os classificadores presentes no SMC executam a mesma tarefa, trabalhando sempre no mesmo conjunto de dados, e ao final da execução retornando a classificação da nova amostra.

Todas as opiniões dos classificadores então devem ser combinadas utilizando alguma métrica de combinação, tendo assim a resposta final da classe da amostra. Um bom exemplo de SMC com topologia paralela pode ser representado pela Figura 2.10.

2.4.2 Topologia Serial

Na abordagem de topologia serial, ilustrada na Figura 2.11, os classificadores são "enfileirados" em ordem de complexidade, sendo os primeiros menos e os últimos mais complexos.

Nesta topologia cada nova amostra é submetida ao classificador 1, caso o mesmo não consiga demonstrar um determinado grau de certeza na classificação, a amostra segue para o classificador 2. Esse processo é repetido até que o grau desejado de certeza da amostra seja alcançado, ou até chegar ao último classificador.

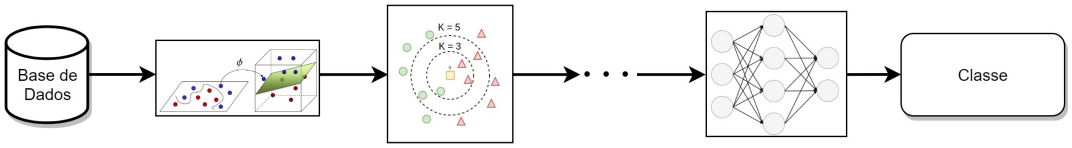

Figura 2.11: Topologia Serial

A principal desvantagem dessa estrutura é que classificadores subsequentes não conseguem

corrigir erros gerados por classificadores anteriores, fazendo com que haja uma propagação desse erro até o classificador final.

2.4.3 Topologia Mista

A abordagem da topologia mista visa combinar características das técnicas paralela e serial, buscando melhorias na performance. Além de oferecer a possibilidade de paralelização do processamento e checagem da não propagação dos erros.

A Figura 2.12 representa a ideia da topologia mista, onde podemos ter uma mistura de classificadores paralelos e seriais, portanto, a amostra é submetida aos classificadores 1, 2 e N. Caso o classificador 2 não consiga atingir o grau de certeza sobre a classe, a amostra então é enviada ao classificador 3, e ao final do processo, todas as opiniões geradas pelos classificadores, são combinadas de modo a obter apenas um resultado.

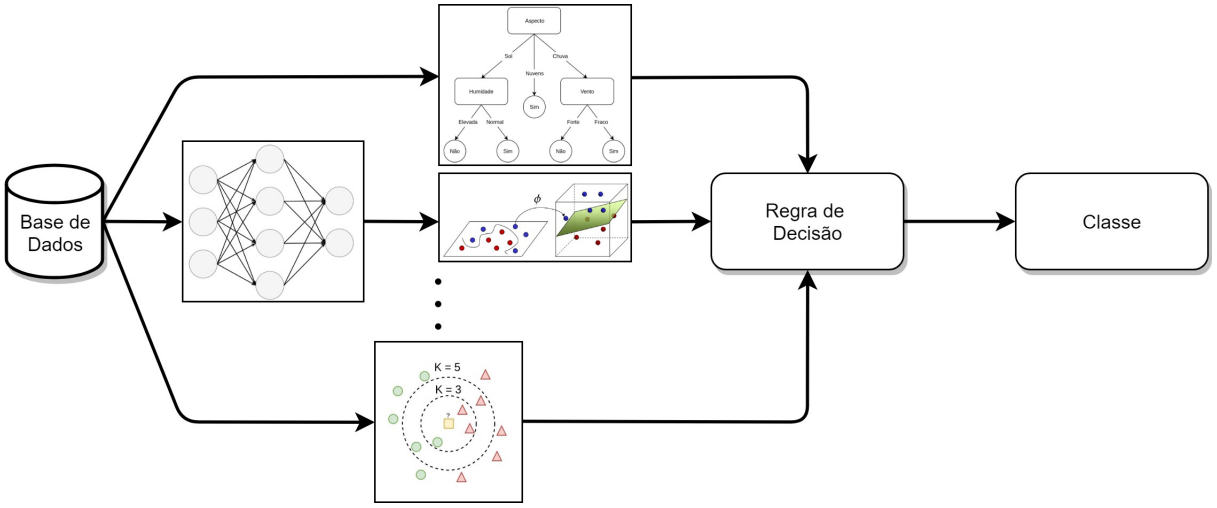

Figura 2.12: Topologia Mista

2.4.4 Regras de Combinação de Opiniões

Devemos considerar que o resultado final da combinação dos classificadores do *pool* deve representar a distribuição de probabilidade conjunta dos resultados de atribuição de classes de cada classificador.

Existem diversas metodologias de combinação de classificadores [Kittler et al. 1998] [Gunes et al. 2003]. Neste trabalho serão utilizados nove dos métodos mais empregados na

literatura, são eles: Soma, Produto, Média, Medina, Max, Min, Voto Majoritário, Borda Count e Ranking. Cada uma dessas abordagens são explicadas a seguir e exemplificadas na Figura 2.13.

Existem alguns pontos a serem destacados para o melhor entendimento das estratégias de combinação, sendo eles:

- m representa o número de classificadores que estão sendo combinados;
- $p(x_1, ..., x_m)$ representa as probabilidades a *posteriori* de uma amostra;
- $p(x_i)$ representa a probabilidade individual de cada classificador para um amostra;
- ω representa as classes às quais a amostra pode pertencer;
- $p(\omega_k|x_i)$ corresponde ao vetor de probabilidades x_i da amostra pertencer à classe ω segundo o classificador k;
- R corresponde ao número de classes que o problema apresenta.

1. **Soma:** É o somatório de todas as probabilidades geradas individualmente por cada classificador quantificando a classe que será atribuída a amostra [Kittler et al. 1998]. A classe que acumular o maior somatório de probabilidades será atribuída à instância de teste. A Equação 2.2 demonstra como a estratégia da soma combina os resultados.

$$p(x_1, ..., x_m) = \sum_{i=1}^{m} p(x_i) \tag{2.2}$$

2. **Produto:** A combinação por produto quantifica a probabilidade de uma hipótese combinando todas as opiniões a *posteriori* geradas pelos classificadores e feito o produto para se estabelecer um valor final de forma a definir a classe da amostra [Kittler et al. 1998]. Será definida como resultado a classe cujo o produtório possuir o maior valor. Análogo, ao processo definido pela regra de combinação da soma, a regra do produto irá multiplicar ao invés de somar. A Equação 2.3 demonstra como a estratégia do produto combina os resultados.

$$p(x_1, ..., x_m) = \prod_{i=1}^{m} p(x_i) \tag{2.3}$$

3. **Média:** Essa estratégia de combinação consiste em aplicar a média de todas as probabilidades geradas individualmente por cada classificador quantificando a classe que será atribuída à amostra. A Equação 2.4 demonstra como a estratégia da média combina os resultados.

$$p(x_1, ..., x_m) = \frac{1}{m} \sum_{i=1}^{m} p(x_i) \tag{2.4}$$

4. **Mediana:** Diferentemente da combinação por média, a combinação por mediana consistem em aplicar a mediana de todas as probabilidades geradas individualmente por cada classificador, a classe que possuir o maior valor será a escolhida como classe da amostra. A Equação 2.5 demonstra como a estratégia da mediana combina os resultados.

$$\underset{i=1}{\overset{R}{median}}\ p(\omega_j|x_i) = \max_{k=1}^{m} \underset{i=1}{\overset{R}{median}}\ p(\omega_k|x_i) \tag{2.5}$$

5. **Max:** A combinação pela regra do máximo consiste em escolher a classe cujo probabilidade de acerto dada *a priori* por cada classificador seja a maior possível [Kittler et al. 1998]. A Equação 2.6 demonstra como a estratégia do max combina os resultados.

$$\max_{i=1}^{R} p(\omega_j|x_i) = \max_{k=1}^{m} \max_{i=1}^{R} p(\omega_k|x_i) \tag{2.6}$$

6. **Min:** Contrário a regra do máximo, o mínimo irá limitar o produto das probabilidades segundo um limiar mínimo das probabilidades geradas das opiniões dos classificadores [Kittler et al. 1998]. A Equação 2.6 demonstra como a estratégia do min combina os resultados.

$$\min_{i=1}^{R} p(\omega_j|x_i) = \max_{k=1}^{m} \min_{i=1}^{R} p(\omega_k|x_i) \tag{2.7}$$

7. **Voto Majoritário:** Muito parecido com uma escolha eleitoral, na estratégia do voto majoritário a opinião de cada classificador sobre as instância de teste é contabilizada na forma de voto, e aquela classe que apresentar o maior número de votos será atribuída para a amostra. A Equação 2.8 demonstra como a estratégia do voto majoritário combina os resultados, onde Δ representa um contador de votos que cada classe recebe.

$$\sum_{i=1}^{R} \Delta_{ji} = \max_{k=1}^{m} \sum_{i=1}^{R} \Delta_{ki} \tag{2.8}$$

8. **Contagem de Borda (*Borda Count*):** A regra de contagem de borda consiste em determinar uma classe à instância de teste com base na soma de posições atribuídas a cada classe pelos classificadores do conjunto. A estratégia sugere atribuir valores de acordo com o índice de certeza de cada classe sugerida pelo classificador. Como exemplo, podemos nos basear na Tabela 2.2, onde um classificador C_i terá 5 opiniões diferentes (5 classes), cada uma com um grau de certeza que totalizem 100%. Para aquela com o menor grau de certeza, ele atribui o valor 1, à segunda classe com menor, o valor 2 e assim sucessivamente. Esse processo é feito para todos os classificadores do *pool*. Ao final, soma-se os valores de todas as classes. Aquela que possuir o maior somatório, será a escolhida. No caso ilustrado na tabela, o voto final da combinação seria a classe 2.

Tabela 2.2: Funcionamento do Método BordaCount

C_1		C_2		C_3		Resultado	
Classe	Rank	Classe	Rank	Classe	Rank	Classe	Soma
1	5	4	5	2	5	1	5+2+2=9
2	4	2	4	4	4	2	4+4+5=13
3	3	5	3	3	3	3	3+1+3=7
4	2	1	2	1	2	4	2+5+4=11
5	1	3	1	5	1	5	1+3+1=5

9. **Ranking:** Na abordagem de *ranking* [Ruta e Gabrys 2000] atribui-se um ranque crescente para cada classe de acordo com a sua probabilidade, para cada classificador. A classe com maior certeza recebe ranque 1, a segunda classe, ranque 2, e assim, sucessivamente. A classe escolhida será aquela que apresentar o melhor ranque. No caso de empate, por exemplo, de duas classes receberem o mesmo ranque, pode-se escolher a classe com o melhor ranque médio.

A Figura 2.13 representa um exemplo hipotético onde uma amostra é analisada por três classificadores diferentes que podem atribuir uma entre cinco possíveis classes.

Após a amostra ser analisada pelo *pool* de classificadores, temos a porcentagem de certeza de cada classificador para cada uma das cinco classes possíveis a que a amostra pode pertencer. A Tabela 2.3 ilustra algumas das estratégias de combinação das porcentagens levantadas, com base nos resultados obtidos na Figura 2.13.

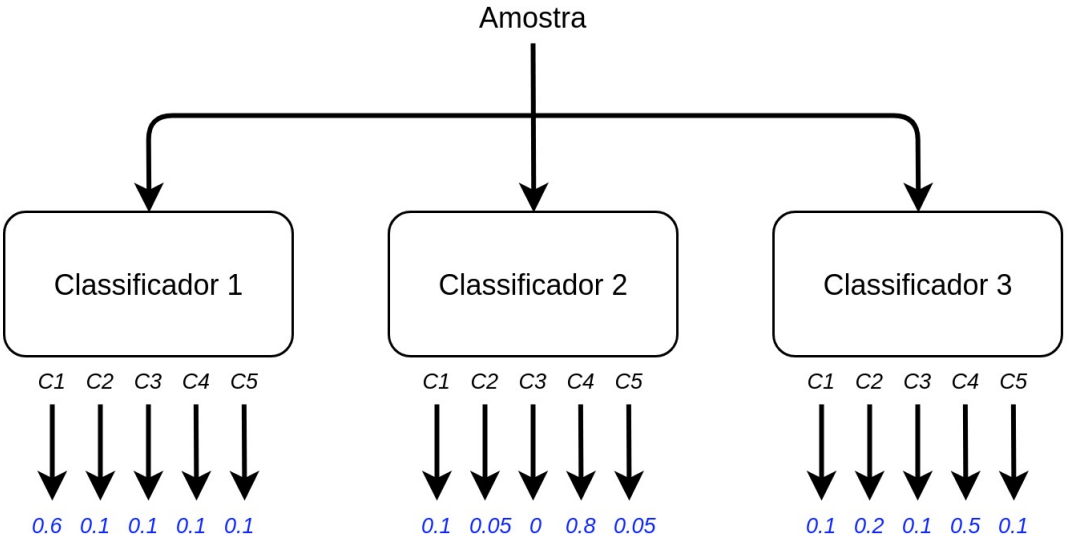

Figura 2.13: Porcentagem de certeza de uma amostra por classificador

Tabela 2.3: Tabela de combinação de porcentagem

Estratégias de Combinação					
Classes	C1	C2	C3	C4	C5
Soma	0.8	0.35	0.2	**1.4**	0.25
Produto	0.006	0.001	0.0	**0.04**	0.0005
Max	0.6	0.2	0.4	**0.8**	0.1
Min	**0.1**	0.05	0	**0.1**	0.05
Média	0.18	0.07	0.04	**0.28**	0.05
Mediana	0.1	0.1	0.1	**0.5**	0.1
Voto majoritário	1	0	0	**2**	0

Capítulo 3

Metodologia

Neste trabalho fez-se a implementação de diversas estratégias de combinação de classificadores heterogêneos com o intuito de estimar suas acurácias e também levantar o custo computacional de cada uma. A Figura 3.1 mostra a visão geral da arquitetura do sistema.

Nesta arquitetura, inicialmente define-se a base de dados que será utilizada. Em seguida, é determinado como serão separados o conjunto de dados (em treino, teste e validação). A terceira etapa consiste na geração dos subconjuntos de treinamento dos classificadores que serão utilizados para determinar a classe de entrada, sendo levado em consideração todo o conjunto de treino e os rótulos já conhecidos das classes. Nesta etapa é importante ressaltar a utilização da estratégia de *Bagging* para randomização do subconjunto de treino. Por fim, na quarta etapa são aplicadas as regras de decisão para combinar as opiniões dos classificadores e atribuir a classe à cada uma das amostras de entradas. As etapas descritas e apresentadas na Figura 3.1 serão detalhadas nas seções seguintes.

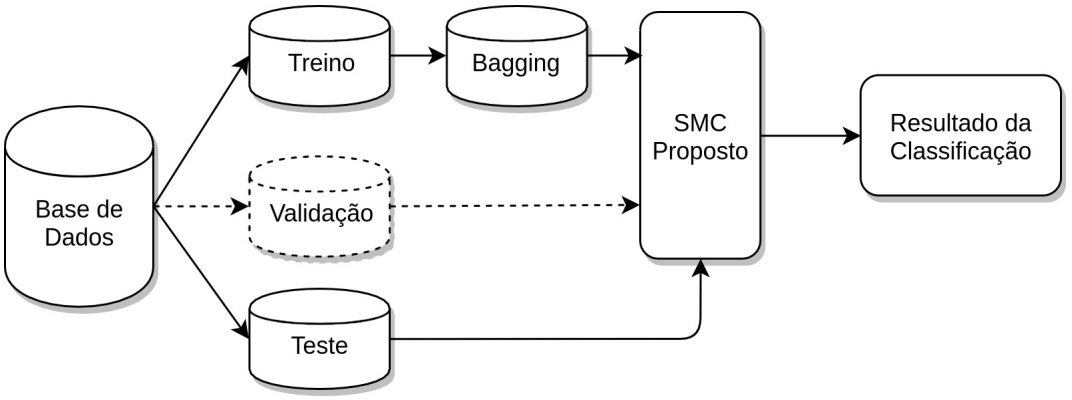

Figura 3.1: Estrutura global do SMC proposto

O desenvolvimento de todo sistema foi feito em Python Versão 3.6 com a utilização de

duas bibliotecas. A primeira é o Pandas[1] empregada na manipulação dos dados tanto para leitura quanto escrita. A segunda biblioteca é o *Scikit-learn*[2] que oferece as implementações dos métodos de classificação mais conhecidos na literatura, bem como estratégias para as separações da base de dados em subconjuntos.

3.1 Divisão das Bases em Subconjuntos

A etapa da entrada dos dados basicamente envolve a leitura das bases que estão em arquivos com formato CSV (*Comma Separated Values*). Cada base representará uma estrutura de dados dentro do sistema, que por sua vez engloba os métodos de manipulação.

Depois da entrada dos dados e instanciação de cada base como uma estrutura em memória é feita a separação do conjunto de dados em três partes disjuntas: treinamento, validação e teste. A Figura 3.2 mostra como serão divididos esses subconjuntos.

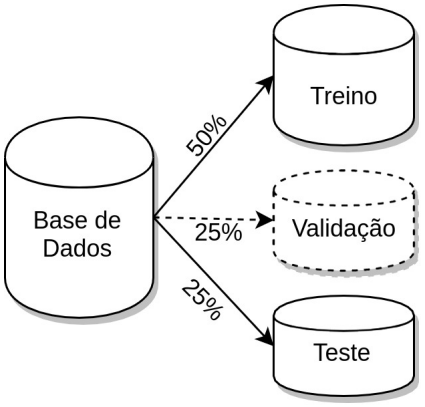

Figura 3.2: Separação da Base de Dados

O subconjunto de treino é o primeiro a ser utilizado. É a partir dele que os classificadores são treinados para tornarem-se aptos a determinar a classe pertencente de toda nova amostra de entrada que será fornecida ao sistema futuramente.

Um ponto relevante durante a separação das bases de dados é a estratificação das classes. O processo é feito mantendo as proporcionalidades das classes das amostras. Por exemplo, dada uma base de dados hipotética que possua 100 amostras distribuídas em duas classes. A primeira contendo 60 amostras e a segunda as 40 restantes. A Tabela 3.1 representa como devem ser

[1]https://pandas.pydata.org/
[2]http://scikit-learn.org/stable/

feitas as divisões para que seja mantida a proporcionalidade de classes considerando 50% para treino, 25% para validação e 25% para teste.

Tabela 3.1: Estrutura de divisão estratificada de uma base hipotética

Classe	Treino	Teste	Validação	Total
C1	30	15	15	60
C2	20	10	10	40
Total	50	25	25	100

3.2 Geração de subconjuntos de Treino

Todas as amostras presentes no conjunto de treino obtido na seção anterior são submetidas ao algoritmo de *Bagging* [Breiman 1996]. No método geralmente se determina o tamanho inicial que o subconjunto gerado deverá possuir. Usualmente este tamanho é definido com base em proporções relacionadas com o conjunto de treino como um todo. Por exemplo, pode-se determinar que o *Bagging* deverá gerar subconjuntos com 50% do tamanho do conjunto de treino.

3.3 Ajuste da Precisão dos Classificadores

O ajuste de precisão ocorre logo após o treinamento dos classificadores pertencentes ao conjunto. O processo busca determinar os melhores parâmetros para cada classificador utilizado. A Figura 3.3 demonstra como ocorre o processo.

Durante a etapa de treinamento, as instâncias dos subconjuntos gerados pelo *Bagging* são fornecidas ao classificador para que este construa um modelo que represente da forma mais precisa possível as informações contidas nas instâncias de treino. Na etapa de validação, no entanto, este modelo não sofre alterações. As únicas variações aplicadas neste momento dizem respeito aos parâmetros adotado na classificação.

A utilização do conjunto de validação tem puramente o intuito de refinar os classificadores, ou seja, melhor adaptá-los para uma determinada base de dados aumentando as chances de obter taxas de reconhecimento melhores na classificação de novas amostras de entrada.

É importante destacar que não foi utilizado o conjunto de validação para medir o desempenho do SMC, pois isso ajustaria deliberadamente (*overfitting*) os resultados para obter o melhor

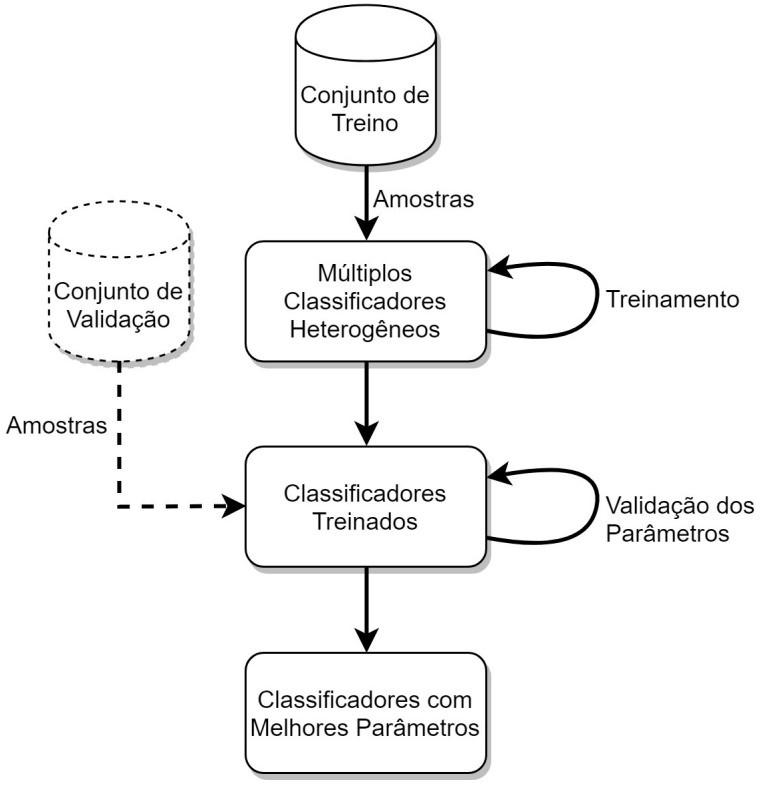

Figura 3.3: Aplicação do Subconjunto de Validação

desempenho possível.

Quando a base de dados é divida em apenas dois subconjuntos (treino e teste), e utiliza o conjunto de treino para selecionar os melhores parâmetros de cada classificador, o desempenho na maioria das vezes sofreria enfraquecimento, isso ocorre pois o conjunto de treino tem a simples funcionalidade de ajustar os parâmetros do modelo de classificação, devendo ser utilizado uma única vez e da forma mais neutra possível.

3.4 Levantamento e Combinação das opiniões

Depois de treinados e ajustados os classificadores, cada instância do conjunto de teste t_i é então submetida ao processo de reconhecimento por cada classificador (C_1, C_2, C_3,..., C_n). A Figura 3.4 expressa a etapa descrita. Na ilustração, o vetor de características de cada instância de teste t_i é fornecido como entrada aos n classificadores adotados. Cada um destes tem a incumbência de determinar, com certo grau de certeza, a classe (OP_1, OP_2, OP_3,..., OP_n) a qual ele acha que a instância t_i pertence.

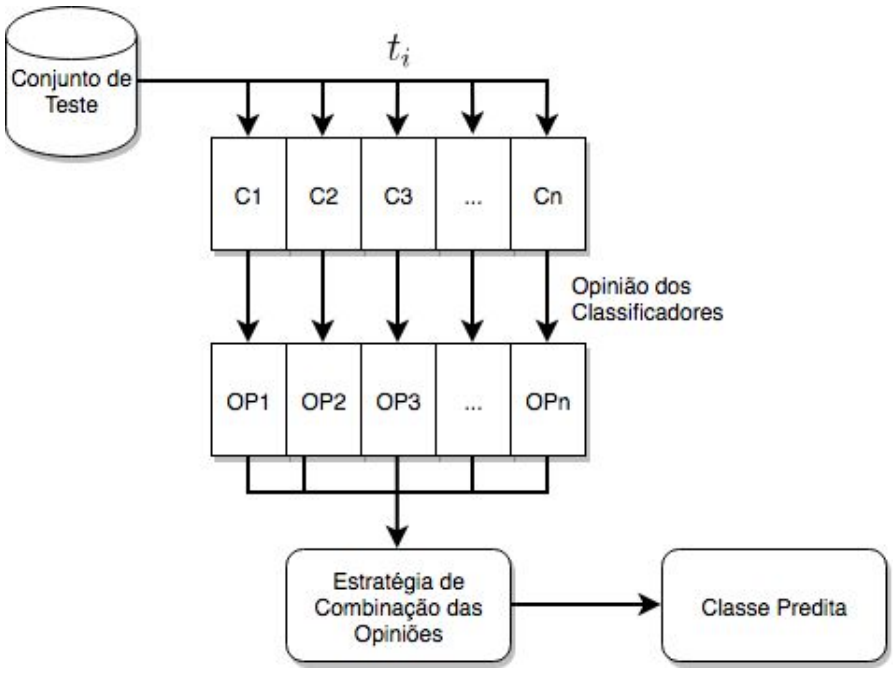

Figura 3.4: Critérios de Combinação

Colhidas as opiniões de todos os classificadores do conjunto sobre a instância de teste, aplica-se então as estratégias de combinação desenvolvidas: soma, produto, média, mediana, mínimo, máximo, borda count e ranking.

Cada estratégia, ao combinar as opiniões dos classificadores, definirá uma opinião comum para todo o conjunto. Essa opinião será então comparada à da instância de teste para verificar se o processo obteve sucesso na classificação. Neste contexto, a acurácia de cada abordagem de combinação será medida através da relação do número de instâncias rotuladas corretamente perante o número total de instâncias presentes no conjunto de teste.

Capítulo 4

Experimentos

As seções a seguir descrevem detalhadamente o processo da execução dos experimentos do sistema descrito no Capítulo 3, apresentando detalhes da separação das bases de dados, das etapas de treinamento, quais foram os melhores parâmetros durante a etapa da validação, bem como a aplicação dos testes e das estratégias de combinação.

4.1 Conjunto de dados

Durante os experimentos foram utilizadas 30 bases de dados de diferentes contextos provenientes dos seguintes repositórios: UCI [Dheeru e Taniskidou 2017], KEEL (*Knowledge Extraction based on Evolutionary* Learning) [Alcalá-Fdez et al. 2008], LKC (*Ludmila Kuncheva Collection of Real Medical Data*) [Kuncheva 2002] e STATLOG [King, Feng e Sutherland 1995]. O critério de escolha para estas bases foi o fato de que elas são amplamente adotadas em trabalhos da área, o que permite um certo nível de comparação de comportamentos.

Além disso, as bases Banana e Lithuanian foram geradas artificialmente através da ferramenta PRtools[1] do Matlab. A escolha por tais bases dá-se pelo fato de ambas apresentarem um comportamento esperado.

A Tabela 4.1 apresenta detalhes como o número de instâncias, de classes e quantidade de atributos de cada uma das bases empregadas no estudo. Além de apontar a fonte de cada conjunto de dados.

O número de instâncias que compõe cada um dos subconjuntos é apresentado na Tabela 4.1, nas colunas "Treino", "Teste"e "Validação". Para a obtenção destes conjuntos, adotou-se as

[1] http://prtools.org/

Tabela 4.1: Principais características das bases usadas nos experimentos

Base	Instâncias	Treino	Teste	Validação	Atributos	Classes	Fonte
Adult	690	345	172	173	14	2	UCI
Banana	2000	1000	500	500	2	2	PRTools
Blood	748	374	187	187	4	2	UCI
CTG	2126	1063	531	532	21	3	UCI
Diabetes	766	383	192	191	8	2	UCI
Ecoli	336	168	84	84	7	8	UCI
Faults	1941	971	485	485	27	7	UCI
German	1000	500	250	250	24	2	STATLOG
Glass	214	107	53	54	9	6	UCI
Haberman	306	153	76	77	3	2	UCI
Heart	270	135	67	68	13	2	STATLOG
ILPD	583	292	145	146	10	2	UCI
Segmentation	2310	1155	577	578	19	7	UCI
Ionosphere	350	176	87	87	34	2	UCI
Laryngeal1	213	107	53	53	16	2	LKC
Laryngeal3	353	177	88	88	16	3	LKC
Lithuanian	2000	1000	500	500	2	2	PRTools
Liver	345	173	86	86	6	2	UCI
Magic	19020	9510	4755	4755	10	2	KEEL
Mammo	830	415	207	208	5	2	KEEL
Monk	432	216	108	108	6	2	KEEL
Phoneme	5404	2702	1351	1351	5	2	ELENA
Sonar	208	104	52	52	60	2	UCI
Thyroid	692	346	173	173	16	2	LKC
Vehicle	847	423	212	212	18	4	STATLOG
Vertebral	300	150	75	75	6	2	UCI
WBC	569	285	142	142	30	2	UCI
WDVG	5000	2500	1250	1250	21	3	UCI
Weaning	302	151	75	76	17	2	LKC
Wine	178	89	44	45	13	3	UCI

proporções 50%, 25% e 25% respectivamente.

Todas as bases mencionadas já passaram pelas etapas tanto de aquisição quanto de pré-processamento, portanto, não houve necessidade de verificação ou implementação de métodos para preenchimento de dados faltantes ou correção de distorções nas bases.

4.2 Treinamento

Identificados os conjuntos de treino a partir da base original, estes foram submetidos ao algoritmo de *Bagging* com 50% e 20% do tamanho do conjunto de treino. Formou-se os subconjuntos empregados na construção dos modelos. Durante a etapa de construção dos modelos de classificação, os subconjuntos de treino (que correspondem a 20% e 50% do conjunto original) foram utilizados para o treinamento de cada classificador (KNN, SVM, MLP, DT, NB).

4.3 Estimação dos Parâmetros

Depois de treinados os classificadores, foi necessário encontrar as melhores configurações de parâmetros com o objetivo de aumentar a acurácia dos classificadores para cada base analisada individualmente. Este processo foi realizado com base no conjunto de validação, evitando assim possível *overfitting*. Cada classificador, por sua vez, possui diversos parâmetros. No entanto, este trabalho focou na calibragem apenas dos mais significativos. O processo de estimação para cada um destes parâmetros é detalhado a seguir.

1. K-Vizinhos mais próximos (KNN)

    - **Número de vizinhos**: foram utilizados todos os valores ímpares e inteiros do intervalo [1,20];

2. Máquina de vetor de suporte (SVM)

    - **Funções de Kernel**: Polinomial, RBF e Linear;
    - **Gamma**: é o coeficiente das funções de hiperplano. Foram utilizados os valores 0.1, 0.01, 0.001, 0.0001 e $\frac{1}{NumeroAtributos}$;
    - **C**: corresponde ao parâmetro de penalidade de erro. Utilizou-se valores entre 0 e 1, com incrementos de 0.1.

3. Árvore de decisão (DT)

    - **Função da qualidade de divisão**: foram utilizados impurezas de Gini e Entropia;
    - **Estratégia de divisão de nós**: utilizou-se a divisão Randômica e a Ótima;
    - **Número mínimo de amostras para divisão interna de um nó**: os valores testados foram 2, 10 e 20;
    - **Número mínimo de amostras contidas em uma folha**: os valores utilizados foram 1, 5 e 10;
    - **Máxima profundidade**: adotou-se as profundidades 5, 10, 20 e ilimitada;
    - **Número máximo de folhas**: testou-se os limites 5, 10, 20 e ilimitado.

4. Perceptron de múltiplas camadas (MLP)

    - **Taxa de aprendizagem**: avaliou-se as abordagens constante, *invscaling* e adaptativa;
    - **Número de neurônios nas camadas escondidas**: O intervalo testado foi de 10 a 100 com passos de 20;
    - **Número de camadas escondidas**: testou-se o modelo com 1, 2 e 3 camadas;
    - **Função de ativação**: utilizou-se a função sigmoide logística e a unidade linear retificada (relu);
    - **Máximo de épocas**: foram utilizadas valores no intervalo de 100 a 500, com passos de 100.

5. Naive Bayes (NB) por ser um classificador probabilístico, não existem parâmetros a serem variados.

4.3.1 Melhores Parâmetros

As Tabelas A.1, A.2, A.3 e A.4 apresentadas no Apêndice A, detalham os melhores parâmetros para cada classificador em cada uma das bases. Visando análise mais sucinta dos parâmetros selecionados, a seguir são feitos alguns apontamentos e representações de frequência dos mesmos.

Dado o grande número de parâmetros, foram escolhidos apenas alguns deles para serem representados graficamente. A justificativa pela escolha dos parâmetros em específico deu-se do fato de que várias bases compartilham os mesmos valores, diferentemente de outros parâmetros, que não apresentaram repetição em diferentes bases.

SVM

Para as opções de *kernel*, a mais frequente foi o RBF com 16 ocorrências. A segunda opção mais frequente foi o kernel linear, tendo sido a melhor opção em 11 casos e, por último, o kernel polinomial foi a melhor em apenas 3 bases.

O valor mais frequente para o parâmetro *gamma* foi adotar $1/NumAtributos$ com 14 ocorrências. O segundo valor mais adotado foi 0.1, com 8 ocorrências, seguido do valor 0.01 com

7. Por fim, em apenas uma das bases o melhor valor foi 0.001.

O número de ocorrências de cada penalidade do erro do SVM é apresentado no gráfico da Figura 4.1. As barras indicam o número de bases em que aquele valor foi o mais adequado. Analisando-se o comportamento de tal parâmetro percebe-se que não há um valor ótimo para todas as bases.

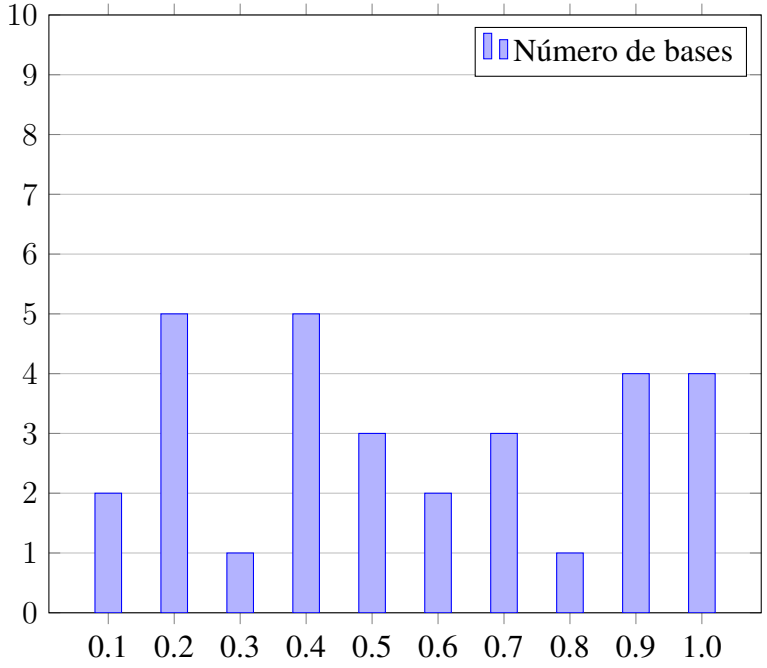

Figura 4.1: Número de bases em que cada Penalidade de Erro SVM foi aplicada

KNN

Os valores para o número de vizinhos observados durante as execuções são apresentados no gráfico da Figura 4.2 onde as barras correspondem ao número de ocorrências da quantidade de vizinhos. Foi possível observar que, para nenhuma das bases, o melhor número de vizinhos foi 15 ou 19.

DT

A função para determinar a qualidade da divisão dos nodos mais adotada foi a Gini, com 24 ocorrências. Para as bases restantes, foi escolhida a entropia como critério.

O critério de divisão mais frequente foi o randômico, com 21 ocorrências. Já a divisão do

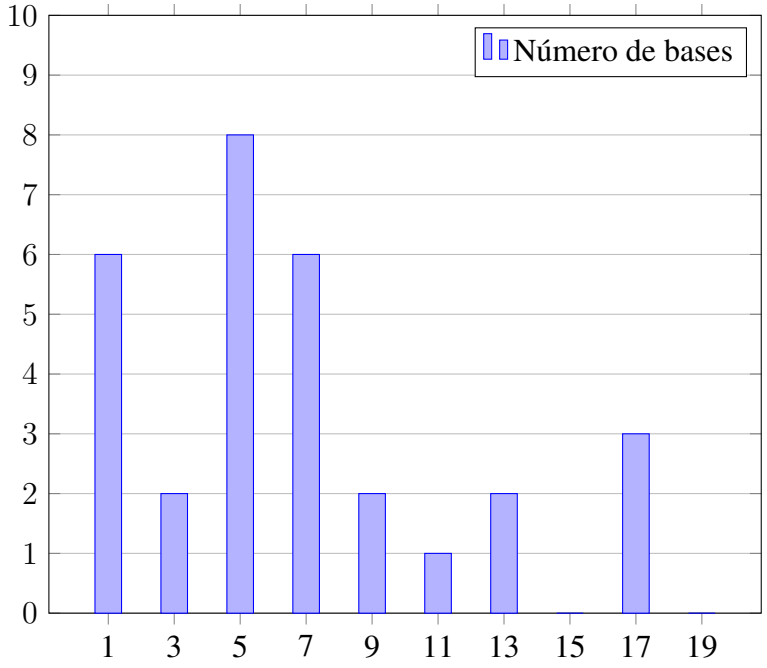

Figura 4.2: Contagem do número de bases em que cada Número de Vizinhos foi adotado no KNN

tipo *best* mostrou-se a melhor opção em 11 bases. Importante ressaltar que, para duas bases, não houve diferença entre o critério escolhido (por isso a soma totaliza 32 casos).

O número mínimo de amostras para realizar a divisão mais comum foi de 2 unidades, o qual mostrou-se a melhor solução para 21 das 30 bases. Além disso, foram observados os valores 10 (6 casos), 20 instâncias mínimas (5 casos) e, em um caso, o valor 1. Para algumas bases, mais de um valor para o parâmetro foi apresentado como melhor configuração.

Para o parâmetro número mínimo de instâncias por folha, o valor mais comumente adotado foi uma unidade (em 19 casos). Na sequência, os valores mais frequentes foram o 10 (6 casos), 5 (5 casos) e, para uma base, o valor escolhido foi de 2 unidades.

Para o parâmetro máxima profundidade da árvore houve triplo empate entre os valores 5 níveis, 10 níveis e sem poda (sem limitação de altura), com dez casos para cada um.

Analisando-se o número máximo de folhas da árvore, o valor mais frequentemente empregado foi não delimitar a quantidade máxima (observado em 15 bases). Na sequência, os valores mais frequentes para o máximo de folhas foram de 10 e 20, com 8 e 7 ocorrências, respectivamente. Em apenas 3 bases a melhor opção foi estabelecer um máximo de 5 folhas.

MLP

Para todas as bases, a taxa de aprendizagem que mais se mostrou adequada foi a constante, com valor inicial de 0.001.

O número máximo de épocas mais frequente foi de 100 iterações, sendo adotado em 23 bases. Em cinco bases, o melhor número de épocas foi definido em 200 iterações e, nos dois casos restantes, os melhores valores foram de 300 e 400 épocas.

Definidos os melhores parâmetros para cada método de aprendizagem submeteu-se o conjunto de teste para avaliação. Nesta etapa, foram colhidas as opiniões de cada classificador, as quais contêm o grau de certeza (probabilidades), da amostra, para cada uma das possíveis classes.

4.4 Resultados das Combinações

Nesta etapa, com o objetivo de alcançar a maior taxa de acertos possível, todas as opiniões e probabilidades foram submetidas a cada uma das estratégias de combinação descritas na Seção 2.4.4. Em seguida, os valores de acurácia alcançados pelas abordagens foram comparados.

Com o objetivo de obter o desempenho médio de cada uma das estratégias de combinação, foram executadas 20 repetições do processo descrito acima. Os valores médios de acurácia e seus respectivos desvios padrão são apresentados na Tabela 4.2. Os melhores desempenhos para cada conjunto de dados são apresentados em negrito.

Para critério de comparação dos resultados obtidos na Tabela 4.2 foram estimadas duas métricas de *groundtruth*:

- Oráculo: Para cada nova entrada que é analisada pelo *pool* de classificadores, essa estratégia supõem que, se pelo menos um classificador do conjunto consegue reconhecer com sucesso a classe da amostra, então o *pool* também é capaz de reconhecê-lo.

- *Single Best* (SB): Essa estratégia de combinação de resultados leva em consideração que o classificador que irá avaliar as amostras de teste é aquele que atingiu a maior acurácia durante a etapa de validação dos parâmetros.

Tabela 4.2: Acurácias médias das estratégias de combinação com *Bagging* em 50%

Bases	Borda	Majoritário	Máximo	Média	Mediana	Mínimo	Produto	Ranking	Soma	Oráculo	*Single Best*
Adult	0.8702±0.00	0.8527±0.01	0.8663±0.00	0.8663±0.01	0.8682±0.00	0.8682±0.00	0.8663±0.00	**0.8721±0.00**	0.8663±0.01	0.9457 ±0.00	0.8760 ±0.00
Banana	**0.9593±0.00**	0.8553±0.00	0.9573±0.00	0.9580±0.00	0.9580±0.00	0.9573±0.00	0.9573±0.00	0.9569±0.00	0.9580±0.00	0.9853 ±0.00	0.8300 ±0.00
Blood	0.7308±0.00	0.7166±0.0	**0.7540±0.00**	0.7433±0.00	0.7362±0.00	**0.7540±0.00**	0.7451±0.00	0.7408±0.00	0.7433±0.00	0.8645 ±0.00	0.7125 ±0.00
Ctg	0.8933±0.00	0.8180±0.00	0.8638±0.00	0.8858±0.00	0.8927±0.00	0.8675±0.00	0.8782±0.00	**0.8965±0.00**	0.8858±0.00	0.9787 ±0.00	0.9259 ±0.01
Diabetes	0.7469±0.00	0.7208±0.01	0.7417±0.01	0.7365±0.01	0.7347±0.01	0.7417±0.01	0.7365±0.01	**0.7487±0.00**	0.7365±0.01	0.9389 ±0.01	0.7138 ±0.00
Ecoli	0.4423±0.04	0.7446±0.04	0.4167±2.27	0.4167±2.27	0.4167±2.27	0.4167±2.27	0.4167±2.27	**0.7464±0.03**	0.4167±2.27	0.9071 ±0.01	0.4780 ±0.00
Faults	**0.7573±0.01**	0.6344±0.00	0.7024±0.00	0.7430±0.00	0.7436±0.01	0.7141±0.00	0.7354±0.00	0.7567±0.01	0.7430±0.00	0.9203 ±0.00	0.6955 ±0.00
German	0.7707±0.00	0.3573±0.03	0.7333±0.03	**0.7733±0.02**	0.7613±0.01	0.7333±0.03	0.7707±0.03	0.7701±0.00	**0.7733±0.02**	0.9773 ±0.01	0.7413 ±0.02
Glass	**0.6774±0.03**	0.6660±0.05	0.3019±5.69	0.3019±5.69	0.3019±5.69	0.3019±5.69	0.3019±5.69	0.6434±0.03	0.3019±5.69	0.8868 ±0.03	0.3321 ±0.02
Haberman	0.7807±0.01	0.7061±0.01	0.7675±0.00	0.7763±0.01	0.7632±0.0	0.7675±0.00	0.7763±0.01	**0.7822±0.01**	0.7763±0.01	0.8289 ±0.01	0.7632 ±0.00
Heart	0.8900±0.01	0.8010±0.00	0.8756±0.02	0.8856±0.00	**0.9005±0.00**	0.8756±0.02	0.8806±0.01	0.8955±0.01	0.8856±0.00	0.9303 ±0.00	0.8756 ±0.01
Ilpd	0.7312±0.00	0.5365±0.01	0.5868±0.02	0.6735±0.01	**0.7717±0.00**	0.5868±0.02	0.5982±0.02	0.7443±0.00	0.6735±0.01	0.9772 ±0.00	0.7078 ±0.01
Ionosphere	0.8826±0.02	**0.9091±0.01**	0.9053±0.00	0.8977±0.01	0.8826±0.01	0.9053±0.00	0.9015±0.01	0.8721±0.01	0.8977±0.01	0.9621 ±0.01	0.8523 ±0.0
Laryngeal1	0.7712±0.00	**0.8050±0.01**	0.7610±0.01	0.7610±0.01	0.7547±0.0	0.7610±0.01	0.7547±0.0	0.7610±0.01	0.7610±0.01	0.9396 ±0.01	0.7673 ±0.01
Laryngeal3	0.6591±0.01	0.6591±0.03	0.6439±0.01	**0.6629±0.02**	0.6591±0.01	0.6326±0.01	0.6477±0.01	0.6532±0.01	**0.6629±0.02**	0.9006 ±0.00	0.6439 ±0.00
Lithuanian	0.9740±0.00	0.8853±0.00	0.9727±0.01	0.9740±0.00	**0.9760±0.00**	0.9733±0.00	0.9740±0.00	0.9747±0.00	0.9740±0.00	0.9940 ±1.35	0.9780 ±0.00
Liver	0.6324±0.06	0.5465±0.01	**0.6899±0.02**	**0.6899±0.03**	0.6550±0.05	0.6860±0.02	**0.6899±0.03**	0.6434±0.06	**0.6899±0.03**	0.9225 ±0.02	0.6279 ±0.11
Magic	0.8616±0.00	0.7939±0.00	0.8259±0.00	0.8584±0.00	0.8601±0.00	0.8259±0.00	0.8472±0.00	**0.8692±0.00**	0.8584±0.00	0.9434 ±0.00	0.8720 ±0.00
Mammo	0.8035±0.01	0.7874±0.01	0.8116±0.01	0.8100±0.00	0.8003±0.00	0.8100±0.01	**0.8132±0.01**	0.8020±0.01	0.8100±0.00	0.9018 ±0.00	0.7504 ±0.01
Monk	0.9198±0.01	0.7963±0.03	**0.9784±0.01**	0.9691±0.00	0.9198±0.01	**0.9784±0.01**	**0.9784±0.01**	0.9392±0.01	0.9691±0.00	1.0000 ±0.00	0.7958 ±0.00
Phoneme	0.8299±0.00	0.7849±0.00	0.8344±0.00	0.8364±0.00	0.8347±0.00	0.8344±0.00	**0.8369±0.00**	0.8330±0.00	0.8364±0.00	0.9645 ±0.00	0.8189 ±0.00
Segmentation	0.9411±0.00	0.7597±0.05	0.9024±0.00	0.9382±0.00	**0.9515±0.00**	0.9249±0.00	0.9336±0.01	0.9472±0.01	0.9382±0.00	0.9936 ±0.00	0.9469 ±0.01
Sonar	0.7179±0.01	0.5833±0.02	0.6987±0.04	0.7115±0.05	**0.7244±0.04**	0.6987±0.04	0.7051±0.04	0.7154±0.01	0.7115±0.05	0.9731 ±0.01	0.6154 ±0.01
Thyroid	0.9518±0.00	0.9326±0.00	0.9711±0.00	0.9538±0.00	0.9557±0.00	**0.9711±0.00**	0.9634±0.00	0.9615±0.01	0.9538±0.00	0.9923 ±0.00	0.9345 ±0.00
Vehicle	0.6671±0.00	0.5371±0.02	0.6256±0.01	0.6777±0.01	**0.7030±0.01**	0.6414±0.01	0.6603±0.00	0.6572±0.00	0.6777±0.01	0.9194 ±0.02	0.7836 ±0.02
Vertebral	0.8456±0.03	0.7956±0.00	0.8711±0.00	**0.8711±0.01**	0.8622±0.02	**0.8711±0.00**	0.8667±1.35	0.8400±0.03	**0.8711±0.01**	0.9556 ±0.00	0.8533 ±0.02
Wbc	0.9507±0.00	0.7676±0.13	**0.9577±0.00**	0.9554±0.00	0.9507±0.01	**0.9577±0.00**	0.9531±0.00	0.9496±0.00	0.9554±0.00	0.9836 ±0.00	0.9343 ±0.00
Wdvg	0.8632±0.00	0.7981±0.00	0.8435±0.00	0.8643±0.00	**0.8677±0.00**	0.8445±0.00	0.8560±0.00	0.8662±0.01	0.8643±0.00	0.9571 ±0.00	0.8592 ±0.00
Weaning	0.8844±0.00	0.8756±0.01	**0.9156±0.02**	0.9067±0.01	0.8800±0.01	**0.9156±0.02**	0.9111±0.02	0.8753±0.00	0.9067±0.01	0.9778 ±0.01	0.8978 ±0.00
Wine	0.9091±0.04	0.5833±0.01	0.9621±0.01	**0.9773±0.0**	0.9697±0.01	0.9621±0.01	0.9621±0.01	0.9357±0.04	**0.9773±0.00**	1.0000 ±0.00	0.9470 ±0.02
Médias	0.8105	0.7337	0.7913	0.8025	0.8019	0.7926	0.7973	0.8217	0.8025	0.9474	0.7843

As estratégias apresentadas nas duas últimas colunas são utilizadas como referência para determinar se o sistema possui resultados próximos aos apresentados, visto que o oráculo é a máxima chance de acerto de um *pool* de classificadores. O ideal é que o sistema sempre seja o mais próximo possível dessa métrica. Também espera-se que o sistema de classificadores seja mais eficiente que o *Single Best*, visto que ele sempre é composto por apenas um classificador definido *a priori* (definido durante a validação) e pode ser interpretado como um limite inferior.

Fazendo um comparativo entre as acurácias alcançadas pelos método de combinação e as métricas de *groundtruth*, percebeu-se que, de forma geral, os resultados das combinações foram superiores aos valores obtidos pelo SB.

Em cinco das trinta bases testadas o SB apresentou a maior acurácia média (Adult, CTG, Lithuanian, Magic e Vehicle). Em dois destes casos o desempenho do SB foi consideravelmente superior ao melhor método de combinação, obtendo acurácia de aproximadamente 3 e 8 pontos percentuais para as bases CTG e Vehicle, respectivamente.

Por outro lado, ao comparar-se as acurácias obtidas com o oráculo, pode se observar que ainda há margem para melhora. Em diversos casos a diferença observada é inferior a 5 pontos percentuais. No entanto, há casos, como das bases German, ILPD, Liver e Sonar, em que a diferença foi superior a 20 pontos percentuais.

Além disso, notou-se que o comportamento das abordagens da média e soma foram idênticas para todas as bases. Isso deve-se ao fato de que, uma maior soma de probabilidade representa, consequentemente, uma maior média. Assim, em todos os cenários em que uma foi melhor, por consequência, a outra também. Afora os empates entre a soma e média, se observou que, em treze situações ocorreram empates entre duas ou mais estratégias.

Visando ilustrar o desempenho dos métodos de combinação perante os 30 problemas, construiu-se o histograma representado na Figura 4.3. Cada barra corresponde ao número de bases em que o método apresentou o melhor desempenho. Em casos de empate, contabilizou-se vitória para ambos os métodos. Analisando-se os resultados foi possível observar que a estratégia do voto majoritário obteve o menor número de vitórias (2). No entanto, perante as demais estratégias, não houve uma que tenha se sobressaído perante as demais.

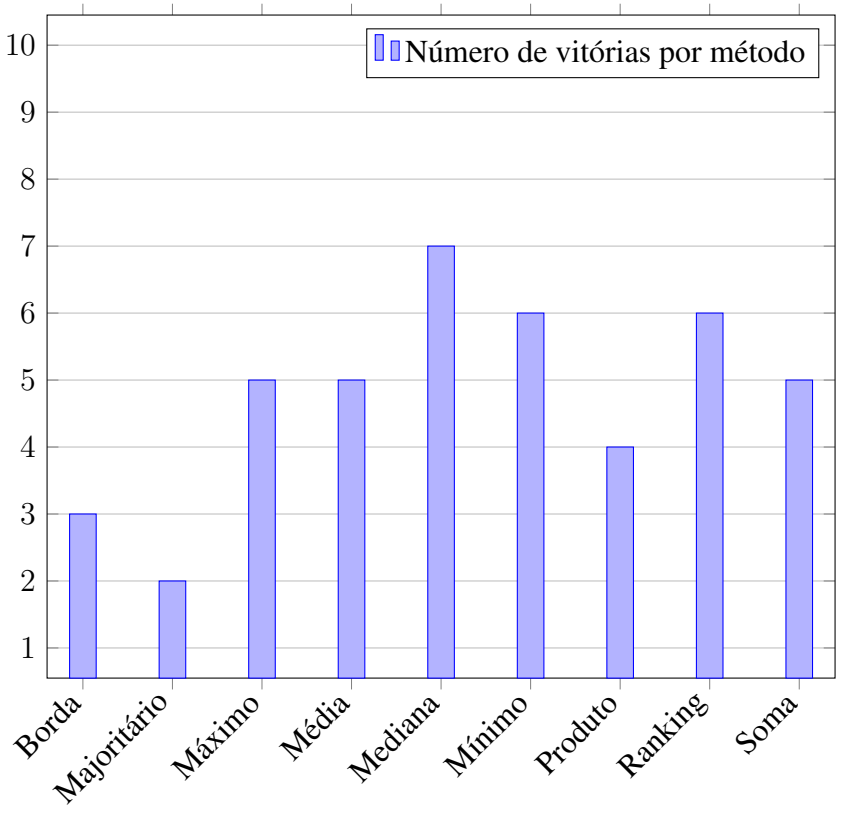

Figura 4.3: Número de vitórias por método

4.5 Validação Estatística

De forma a verificar se existe diferença significativa entre as acurácias de cada estratégia de combinação foi aplicada avaliação estatística para cada uma das bases do conjunto. Neste cenário, a hipótese nula afirma não haver diferença significa entre as abordagens. Já a hipótese alternativa defende a existência de diferença significante entre pelo menos duas estratégias de combinação.

Inicialmente utilizou-se o teste estatístico de postos de Kruskal-Wallis com 95% de confiança. Este teste avalia se existe diferença estatística entre mais que dois conjuntos de dados simultaneamente. Por tratar-se de um método não paramétrico, não existe a necessidade de que os dados apresentem distribuição normal.

Os resultados do teste de Kruskal-Wallis apontaram para rejeição da hipótese nula em 23 das 30 bases. As exceções foram: Adult, Ionosphere, Laryngeal1, Mammo, Sonar, Thyroid, Vertebral e WBC, para as estratégias de combinação utilizadas.

Após os resultados obtidos com o teste de Kruskal-Wallis, foi aplicado o teste de Nemenyi, que considera o desempenho de todas as estratégias de combinação com o intuito de verificar seu desempenho médio em termos de ranques. A Figura 4.4 apresenta uma representação gráfica dos resultados colhidos.

O teste de Nemenyi, ao comparar os ranques alcançados pelos métodos de combinação, obteve uma distância crítica de 2.473. Tal limiar está representado na Figura 4.4 pela barra vermelha. A interpretação indica que, se a distância entre duas abordagens for superior o limiar, considera-se que há diferença significativa entre os métodos.

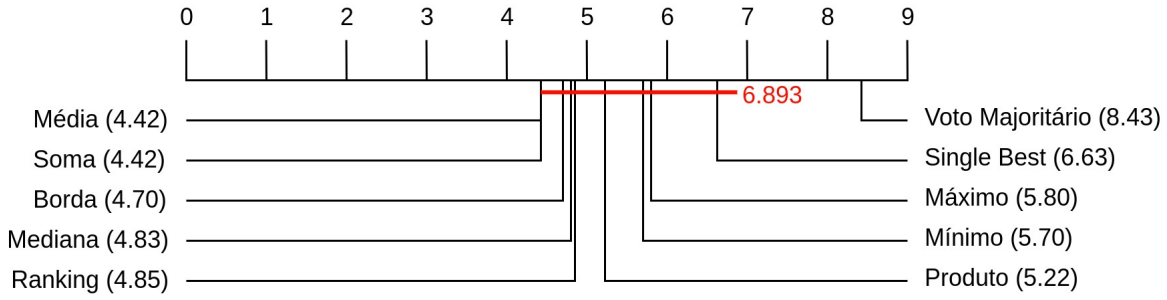

Figura 4.4: Resultado do teste de Nemenyi com *Bagging* em 50%

Analisando-se os ranques médios obtidos pelas estratégias pode-se notar que a Soma e, consequentemente a Média alcançaram o melhor ranque (4.42). Na sequência, os métodos que apresentaram as melhores posição foram Borda Count (4.7), Mediana (4.83), Ranking (4.85), Produto (5.22), Mínimo (5.70), Máximo (5.80), SB (6.63) e Voto Majoritário (8.43).

Com relação à distância crítica, existe diferença significativa entre o Voto Majoritário e todas as abordagens de combinação, considerando que o SB é apenas uma métrica de balizagem. Isso mostra que o desempenho médio do método foi bastante inferior se comparado aos outros.

4.6 Variações do *Bagging*

Visando avaliar a acurácia dos métodos de combinação de classificadores treinados em conjuntos de diferentes tamanhos foram executados os mesmos testes descritos nas seções anteriores, em que o *Bagging* gerou conjunto com tamanho igual a 20% do treino. Os valores médios de acurácia e seus respectivos desvios padrão são apresentados na Tabela 4.3. Os melhores desempenhos para cada base de dados estão destacados em negrito.

Tabela 4.3: Acurácias médias das estratégias de combinação com *Bagging* em 20%

Bases	Borda	Majoritário	Máximo	Média	Mediana	Mínimo	Produto	Ranking	Soma	Oráculo	*Single Best*
Adult	**0.8721 ±0.00**	0.8663 ±0.00	0.8663 ±0.00	0.8663 ±0.00	0.8663 ±0.00	0.8663 ±0.00	0.8663 ±0.00	0.8714 ±0.00	0.8663 ±0.00	0.8895 ±0.01	0.8488 ±0.01
Banana	0.9440 ±0.01	0.8780 ±0.00	**0.9600 ±0.00**	0.9540 ±0.00	0.9480 ±0.01	**0.9600 ±0.00**	0.9560 ±0.00	0.9292 ±0.01	0.9540 ±0.00	0.9840 ±0.00	0.9820 ±0.00
Blood	0.7487 ±0.00	0.7487 ±2.27	**0.7540 ±0.01**	0.7487 ±0.00	0.7487 ±0.00	**0.7540 ±0.01**	0.7487 ±0.00	0.7487 ±0.00	0.7487 ±0.00	0.8396 ±0.02	0.7021 ±0.00
Ctg	0.8889 ±0.00	0.8249 ±0.01	0.8795 ±0.00	0.8927 ±0.00	**0.8983 ±0.00**	0.8795 ±0.00	0.8776 ±0.00	0.8836 ±0.00	0.8927 ±0.00	0.9642 ±0.00	0.8983 ±0.01
Diabetes	0.7115 ±0.00	0.6759 ±0.01	0.7092 ±0.00	0.7076 ±0.00	**0.7123 ±0.00**	0.7089 ±0.00	0.7081 ±0.00	0.7103 ±0.00	0.7076 ±0.00	0.8945 ±0.00	0.6982 ±0.01
Ecoli	0.4423±0.04	0.7446±0.04	0.4167±2.27	0.4167±2.27	0.4167±2.27	0.4167±2.27	0.4167±2.27	**0.7464±0.03**	0.4167±2.27	0.9071 ±0.01	0.4780 ±0.00
Faults	0.7093 ±0.00	0.6041 ±0.01	0.6433 ±0.01	0.7113 ±0.01	**0.7237 ±0.01**	0.7052 ±0.01	0.7155 ±0.01	0.7124 ±0.00	0.7113 ±0.01	0.9175 ±0.00	0.6433 ±0.01
German	0.7560 ±0.01	0.3640 ±0.05	0.6000 ±0.03	**0.7720 ±0.01**	0.7360 ±0.01	0.6000 ±0.03	0.7600 ±0.03	0.7345 ±0.01	**0.7720 ±0.01**	0.9640 ±0.01	0.7160 ±0.02
Glass	**0.6792 ±0.01**	0.4151 ±0.03	0.6415 ±0.02	0.6226 ±0.02	0.6604 ±0.01	0.5472 ±0.01	0.6226 ±0.01	0.6642 ±0.00	0.6226 ±0.01	0.9057 ±0.01	0.7170 ±0.01
Haberman	**0.7368 ±0.01**	0.6842 ±0.01	**0.7368 ±0.01**	**0.7368 ±0.00**	**0.7368 ±0.01**	**0.7368 ±0.01**	**0.7368 ±0.00**	**0.7368 ±0.01**	**0.7368 ±0.00**	0.8026 ±0.01	0.7368 ±0.01
Heart	0.7612 ±0.02	0.7761 ±0.03	**0.7910 ±0.02**	0.7612 ±0.02	0.7612 ±0.02	**0.7910 ±0.02**	0.7612 ±0.02	0.7852 ±0.01	0.7612 ±0.02	0.9104 ±0.01	0.7910 ±0.02
Ilpd	0.7260 ±0.02	0.7123 ±0.02	0.7192 ±0.01	0.7192 ±0.02	0.6986 ±0.01	0.7192 ±0.01	0.6986 ±0.02	**0.7285 ±0.02**	0.7192 ±0.02	0.9521 ±0.01	0.6986 ±0.02
Ionosphere	0.7614 ±0.03	**0.8977 ±0.03**	0.7614 ±0.03	0.7955 ±0.03	0.7273 ±0.03	0.7614 ±0.03	0.7841 ±0.03	0.7598 ±0.02	0.7955 ±0.03	0.9318 ±0.01	0.7500 ±0.04
Laryngeal1	**0.8868 ±0.02**	0.7547 ±0.04	**0.8868 ±0.02**	**0.8868 ±0.02**	**0.8868 ±0.02**	**0.8868 ±0.02**	**0.8868 ±0.02**	**0.8868 ±0.02**	**0.8868 ±0.02**	0.9811 ±0.01	0.8302 ±0.04
Laryngeal3	0.7386 ±0.02	0.6136 ±0.04	**0.7614 ±0.17**	**0.7614 ±0.18**	0.7386 ±0.18	**0.7614 ±0.18**	**0.7614 ±0.18**	0.7386 ±0.02	**0.7614 ±0.18**	0.8750 ±0.01	0.7386 ±0.01
Lithuanian	0.9598 ±0.00	0.8720 ±0.00	0.9580 ±0.00	0.9600 ±0.00	**0.9620 ±0.00**	0.9580 ±0.00	0.9600 ±0.00	**0.9620 ±0.00**	0.9600 ±0.00	0.9780 ±0.00	0.9560 ±0.00
Liver	0.6047 ±0.03	0.5698 ±0.00	**0.7093 ±0.02**	**0.7093 ±0.02**	0.6163 ±0.04	**0.7093 ±0.02**	**0.7093 ±0.02**	0.6267 ±0.03	**0.7093 ±0.02**	0.8605 ±0.02	0.5698 ±0.03
Magic	0.8421 ±0.01	0.7941 ±0.01	0.8123 ±0.01	0.8442 ±0.01	0.8435 ±0.01	0.8123 ±0.01	0.8338 ±0.01	**0.8452 ±0.01**	0.8442 ±0.01	0.9368 ±0.01	0.8605 ±0.00
Mammo	0.7971 ±0.01	0.7826 ±0.01	**0.8019 ±0.01**	**0.8019 ±0.01**	**0.8019 ±0.01**	**0.8019 ±0.01**	**0.8019 ±0.01**	0.7962 ±0.01	**0.8019 ±0.01**	0.8986 ±0.01	0.8019 ±0.01
Monk	0.8981 ±0.02	0.8519 ±0.02	**0.9815 ±2.27**	0.9722 ±0.02	0.8981 ±0.02	**0.9815 ±2.27**	0.9815 ±2.27	0.8981 ±0.02	0.9722 ±0.02	0.9815 ±0.00	0.9815 ±0.00
Phoneme	**0.8335 ±0.00**	0.7905 ±0.00	0.8187 ±0.00	0.8253 ±0.00	**0.8335 ±0.00**	0.8187 ±0.00	0.8142 ±0.00	0.8245 ±0.00	0.8253 ±0.00	0.9563 ±0.00	0.8520 ±0.00
Segmentation	0.9620 ±0.00	0.6590 ±0.00	0.9506 ±0.00	0.9644 ±0.00	**0.9672 ±0.00**	0.9559 ±0.00	0.9643 ±0.00	0.9620 ±0.00	0.9644 ±0.00	0.9975 ±0.00	0.5990 ±0.00
Sonar	0.6923 ±0.03	0.6154 ±0.01	0.6346 ±0.03	0.6346 ±0.03	0.6731 ±0.05	0.6346 ±0.03	0.6346 ±0.02	**0.6986 ±0.02**	0.6346 ±0.03	0.9038 ±0.03	0.6538 ±0.05
Thyroid	0.9422 ±0.00	0.9422 ±0.00	0.9480 ±0.00	0.9422 ±0.00	0.9480 ±0.00	0.9480 ±0.00	**0.9538 ±0.00**	0.9500 ±0.00	0.9422 ±0.00	0.9711 ±0.00	0.9480 ±0.00
Vehicle	0.6682 ±0.02	0.4218 ±0.09	0.6303 ±0.01	**0.6872 ±0.01**	0.6682 ±0.02	0.6635 ±0.01	**0.6872 ±0.01**	0.6654 ±0.02	**0.6872 ±0.01**	0.9194 ±0.02	0.7299 ±0.02
Vertebral	**0.8400 ±0.01**	0.7467 ±0.01	0.7600 ±0.02	0.8133 ±0.01	**0.8400 ±0.01**	0.7600 ±0.02	0.7867 ±0.01	**0.8400 ±0.01**	0.8133 ±0.01	0.9067 ±0.01	0.7600 ±0.02
Wbc	0.9437 ±0.01	0.8732 ±0.08	0.9507 ±0.00	**0.9648 ±0.00**	0.9577 ±0.01	0.9507 ±0.00	0.9577 ±0.01	0.9357 ±0.01	**0.9648 ±0.00**	0.9930 ±0.00	0.9296 ±0.05
Wdvg	0.8608 ±0.00	0.7864 ±0.00	0.8464 ±0.00	0.8616 ±0.00	**0.8632 ±0.00**	0.8472 ±0.00	0.8576 ±0.00	0.8573 ±0.00	0.8616 ±0.00	0.9664 ±0.00	0.8600 ±0.00
Weaning	0.8800 ±0.02	0.8800 ±0.02	**0.9200 ±0.02**	0.8933 ±0.02	0.8800 ±0.02	**0.9200 ±0.02**	**0.9200 ±0.02**	0.9015 ±0.01	0.8933 ±0.02	0.9600 ±0.00	0.9200 ±0.01
Wine	0.8409 ±0.05	0.6136 ±0.10	**0.9318 ±0.01**	**0.9318 ±0.01**	**0.9318 ±0.02**	0.9091 ±0.01	**0.9318 ±0.01**	0.8817 ±0.03	**0.9318 ±0.01**	0.9773 ±0.01	0.9545 ±0.02
Médias	0.7976	0.7253	0.7927	0.8053	0.7981	0.7922	0.8032	0.8094	0.8053	0.9309	0.7868

A duas últimas colunas da Tabela 4.3 correspondem aos valores de acurácia média e seus respectivos desvios padrão para o Oráculo e SB referentes às execuções com 20% do *Bagging*.

Para o teste de Nemenyi, obteve-se uma distância crítica de 2.473. Tal limiar está representado na Figura 4.5 pela barra vermelha.

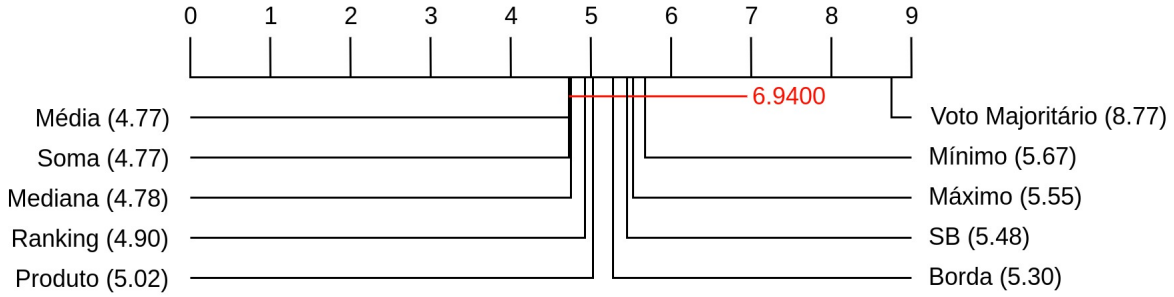

Figura 4.5: Resultado do teste de Nemenyi com *Bagging* em 20%

Analisando-se os ranques médios obtidos pelas estratégias, é possível observar certa semelhança com a Tabela 4.4, onde a Soma e Média obtiveram os melhores ranques (4.77) e o Voto Majoritário continuou em último lugar (8.77), houve mudanças entre os métodos que ficaram entre os primeiros e último, seguindo a sequência do ranque vieram Mediana (4.78), Ranking (4.90), Produto (5.02), Borda Count (5.30), SB (5.48), Máximo (5.55), Mínimo (5.67).

Visando analisar o desempenho obtido pela métricas de combinação em termos de acurácia perante as duas dimensões adotadas pelo *Bagging*, comparou-se o desempenho de cada uma das estratégias para cada uma das bases, obtendo assim o histograma apresentado na Figura 4.6, onde são computados os números de vitórias e empates, par-a-par, de cada abordagem de combinação. Neste cenário, a comparação não levou em conta o desempenho geral da estratégia.

Analisando-se as contagens representadas na Figura 4.6 é possível notar que, para o *Single Best*, utilizar 20% ou 50% houve diferenças significativas em alguns casos, sendo a abordagem de 50% obteve sempre um número maiores de vitórias, no entanto, o número de vitórias tanto para 20% quanto para 50% ficou equilibrado. Por outro lado, o Oráculo, com 50% foi bastante superior em número de vitórias comparado ao se adotar 20%.

A comparação entre todas as combinações levou a ocorrência de poucos empates, mostrando que as duas abordagens normalmente levam a acurácias distintas.

De forma geral, as estratégias de combinação apresentaram proporcionalidade no número de vitórias, com exceção das técnicas de Ranking, Borda Count e Mediana. Nestes casos, o

número de vitórias dos classificadores treinados em conjuntos de 50% foi bastante superior aos de 20%. Acredita-se que tal fato esteja relacionado ao grande número de empates observados na Tabela 4.3 onde, para várias bases, houve empate múltiplo entre as estratégias. Estes empates, provavelmente, se dão por conta de que estas estratégias baseiam-se nos ranques dos classificadores e não em suas confianças diretamente, o que pode fazer com que o mesmo classificador seja escolhido diversas vezes.

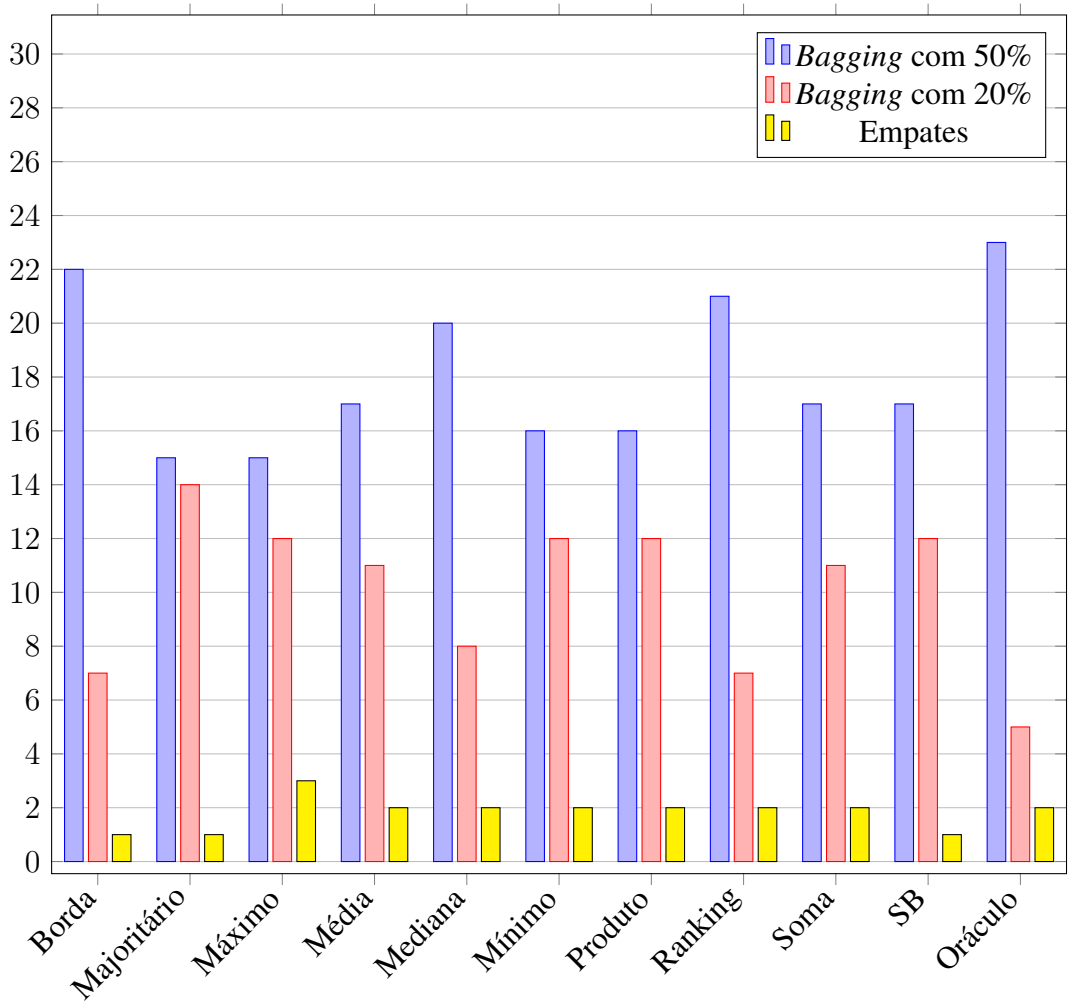

Figura 4.6: Comparação entre diferentes tamanhos do conjunto gerado pelo *Bagging*

De forma geral, as acurácias dos métodos de combinação de classificadores treinados em conjuntos com 50% do tamanho do conjunto de treino foi superior ao se adotar apenas 20%. Acredita-se que os classificadores treinados nos conjuntos maiores tendem a ser mais precisos, dado seu treinamento mais robusto, mas ainda mantêm certa diversidade entre si.

Capítulo 5

Conclusão

Este trabalho teve como objetivo a implementação e validação de um SMC heterogêneo com o intuito de estimar a acurácia de cada uma das estratégias de combinação. Para a validação do sistema foi utilizado um protocolo robusto composto de 30 bases de dados bastante comuns na avaliação de sistemas de reconhecimento na literatura e que não precisavam de tratamento ou segmentação de dados. Além disso, cada experimento foi executado 20 vezes, garantindo embasamento aos resultados obtidos.

A arquitetura do SMC implementado pode ser dividida em 5 etapas principais. A primeira destinada à divisão das bases em treino, validação e teste com 50%, 25% e 25% respectivamente para cada um dos subconjuntos. A etapa seguinte foi a geração dos subconjuntos de treino utilizado o *Bagging* com 50% e 20% do tamanho do conjunto de treino. Em seguida, fez-se o treinamento dos classificadores que foram submetidos à etapa de validação, para definir os melhores parâmetros para cada um dos classificadores em cada uma das bases de dados. A penúltima etapa consiste em levantar as porcentagens de certeza de cada uma das possíveis classes para todas as amostras do conjunto de teste. A última etapa constituiu na combinação das porcentagens de certeza de cada classificador obtidas na etapa anterior. Para tanto, foram utilizados e avaliadas regras de combinação (Borda Count, Média, Mediana, Ranking, Soma, Produto, Mínimo, Máximo e Voto Majoritário), as quais tiveram sua acurácia estimada.

Analisando os resultados obtidos nas execuções do *Bagging* com tamanho igual a 50% e 20%, foi possível observar que não houve mudanças significativas entre as escolhas dos melhores parâmetros dos classificadores para cada base, portanto, os resultados colhidos e apresentados na Tabela A (1, 2, 3, 4), apresentam-se os mesmos para as duas execuções. No entanto, não podemos afirmar se esse comportamento é constante para qualquer tamanho que podem ser ado-

tados pelo *Bagging*. Para tanto, seriam necessários estudos mais aprofundados que tratassem de um número maior de proporções para obter conclusões mais concretas.

Analisando cada métrica de combinação em termos de acurácia, pudemos observar que para os resultados alcançados utilizando o *Bagging* com 50%, as melhores estratégias foram a Soma e a Média, empatadas em primeiro lugar. Em segundo e terceiro lugar, respectivamente, encontraram-se Borda Count e Mediana, seguidos de Ranking, Mínimo, Máximo, e por último o Voto Majoritário, que obteve o pior ranque no teste de Nemenyi. A Soma e por sua vez a Média, já eram estratégias esperadas para ficar nas melhores posições pois, como já foi apontado por Kittler (1998), essa é uma métrica de combinação muito resiliente a presença de eventuais erros, o que provê melhores resultados, quando comparados às demais estratégias de combinação. Não muito diferente dos resultado obtidos com *Bagging* em 50%, com 20% a Soma e a Média obtiveram os melhores ranques, seguidos pelas técnicas da Mediana, Ranking, Produto, Borda Count, Máximo, Mínimo e o Voto Majoritário que manteve-se em último lugar.

Como trabalho futuros, se propõe a divisão dos conjuntos gerados pelo *Bagging* ser feita com proporções distintas. Durante o desenvolvimento deste trabalho, utilizou-se 50% e 20%, no entanto, outros trabalhos podem explorar e avaliar se existe diferença significativa entre essas variações de tamanho e de que forma elas interferem, tanto na definição dos melhores parâmetros quanto na acurácia.

Pode ser interessante a implementação de critérios de desempante dentro das estratégias de combinação, de forma a evitar a frequente seleção de um mesmo classificador. Além disso, seria mais adequado a adoção do voto ponderado como critério de desempate, visto que o voto majoritário simples mostrou-se o menos efetivo em termos de acurácia.

Os códigos esta disponíveis em <https://github.com/jcanabarro/classifiers>, sob a licença General Public License 3.0 (GPL)[1], a qual garante liberdade de uso e modificação, comercial ou não, desde que se mantenha uma cópia da licença e os devidos créditos aos contribuidores junto ao produto.

[1]https://www.gnu.org/lienses/gpl-3.0.pt-br.html

Apêndice A

Melhores Parâmetros

Durante os experimentos foram definidos diversos parâmetros para os métodos de aprendizagem de máquina. Visando mapear aqueles que se mostraram mais adequados a cada cenário, fez-se um levantamento dos valores mais frequentes.

Esse apêndice é dedicado ao detalhamento mais aprofundado dos melhores parâmetros obtidos nas execuções de cada base. Conforme descrito nos Capítulos 3 e 4, foram executadas 20 vezes todo o processo para cada base de dados. Durante a execução do processo, além dos resultados, as melhores configurações de cada classificador também foram salvas.

Como os subconjuntos utilizados durante a execução mudam sempre, em alguns momentos puderam ser observadas pequenas varições de parâmetros entre uma execução e outra. Neste caso, foram apresentados apenas os que apareceram com mais frequência. No caso de empate entre o número de vezes que o parâmetro apareceu, dois ou mais valores foram apresentados na Tabela, como os classificadores possuem diversos parâmetros, não foi possível gerar combinação de todos os possíveis. Para o classificador Naive Bayes não houve variação de parâmetros uma vez que as probabilidades são definidas diretamente sobre o conjunto de dados de entrada.

As Tabelas A1, A2, A3 e A4 apresentam apenas os dados que foram detalhados na Seção 4.3.

Base	SVM	KNN	DT	MLP
Adult	kernel: rbf gamma: 0.01 C: 0.2	neighbors: 5	criterion: gini splitter: best min_samples_split: 20 min_samples_leaf: 5 max_depth: $None$ max_leaf_nodes: $None$	learning_rate: constant hidden_layer_sizes: (60,) max_iter: 100
Banana	kernel: rbf gamma: 0.01 C: 0.2	neighbors: 7	criterion: gini splitter: random min_samples_split: 2 min_samples_leaf: 1 max_depth: $None$ max_leaf_nodes: $None$	learning_rate: constant hidden_layer_sizes: (40,) max_iter: 100
Blood	kernel: linear gamma: $auto$ C: 1.0	neighbors: 3	criterion: gini splitter: best min_samples_split: 1 min_samples_leaf: 2 max_depth: $None$ max_leaf_nodes: $None$	learning_rate: constant hidden_layer_sizes: (5, 2) max_iter: 200
CTG	kernel: poly gamma: $auto$ C: 0.9	neighbors: 5	criterion: gini splitter: best min_samples_split: 2 min_samples_leaf: 1 max_depth: 10 max_leaf_nodes: $None$	learning_rate: constant hidden_layer_sizes: (80,) max_iter: 100
Diabetes	kernel: linear gamma: $auto$ C: 1.0	neighbors: 7	criterion: gini splitter: random min_samples_split: 2 min_samples_leaf: 10 max_depth: 10 ou 20 max_leaf_nodes: 5 ou 20	learning_rate: constant hidden_layer_sizes: (20, 20, 20) max_iter: 200
Ecoli	kernel: linear gamma: $auto$ C: 0.8	neighbors: 1	criterion: entropy splitter: random min_samples_split: 2 min_samples_leaf: 1 max_depth: 10 ou 20 max_leaf_nodes: 10 ou $None$	learning_rate: constant hidden_layer_sizes: (20,) max_iter: 100
Faults	kernel: rbf gamma: 0.1 C: 0.8	neighbors: 5	criterion: entropy splitter: random min_samples_split: 2 min_samples_leaf: 1 max_depth: 10 max_leaf_nodes: $None$	learning_rate: constant hidden_layer_sizes: (80,) max_iter: 100
German	kernel: rbf gamma: 0.01 C: 0.2	neighbors: 17	criterion: gini splitter: random min_samples_split: 10 min_samples_leaf: 10 max_depth: 5 ou $None$ max_leaf_nodes: 10	learning_rate: constant hidden_layer_sizes: (40,) max_iter: 100

Tabela A.1: Melhores parâmetros - Parte 1

Base	SVM	KNN	DT	MLP
Glass	kernel: linear gamma: $auto$ C: 0.4	neighbors: 1	criterion: gini splitter: best min_samples_split: 2 min_samples_leaf: 1 max_depth: 5 ou $None$ max_leaf_nodes: $None$	learning_rate: constant hidden_layer_sizes: (20, 20, 20) max_iter: 100
Haberman	kernel: rbf gamma: 0.1 C: 0.1	neighbors: 9	criterion: entropy splitter: best ou random min_samples_split: 2 min_samples_leaf: 1 max_depth: $None$ max_leaf_nodes: 5	learning_rate: constant hidden_layer_sizes: (20,) max_iter: 300
Heart	kernel: rbf gamma: 0.01 C: 0.6	neighbors: 7	criterion: gini splitter: best ou random min_samples_split: 2 min_samples_leaf: 1 max_depth: 5 ou $None$ max_leaf_nodes: 20	learning_rate: constant hidden_layer_sizes: (80,) max_iter: 100
ILPD	kernel: rbf gamma: 0.1 C: 0.4	neighbors: 5	criterion: entropy splitter: random min_samples_split: 10 ou 20 min_samples_leaf: 10 max_depth: 10 ou 20 max_leaf_nodes: 10	learning_rate: constant hidden_layer_sizes: (40,) max_iter: 100
Ionosphere	kernel: rbf gamma: 0.1 C: 0.7	neighbors: 1	criterion: gini splitter: random min_samples_split: 10 ou 20 min_samples_leaf: 1 max_depth: 10 max_leaf_nodes: 20	learning_rate: constant hidden_layer_sizes: (100, 100) max_iter: 100
Laryngeal1	kernel: rbf gamma: 0.1 C: 0.2	neighbors: 3	criterion: entropy splitter: random min_samples_split: 2 ou 10 min_samples_leaf: 10 max_depth: 10 max_leaf_nodes: $None$	learning_rate: constant hidden_layer_sizes: (20, 20) max_iter: 100
Laryngeal3	kernel: poly gamma: $auto$ C: 0.7	neighbors: 13	criterion: gini splitter: random min_samples_split: 2 min_samples_leaf: 5 max_depth: $None$ max_leaf_nodes: $None$	learning_rate: constant hidden_layer_sizes: (20,) max_iter: 100
Lithuanian	kernel: rbf gamma: 0.01 C: 0.5	neighbors: 5	criterion: gini splitter: best min_samples_split: 2 min_samples_leaf: 5 max_depth: $None$ max_leaf_nodes: $None$	learning_rate: constant hidden_layer_sizes: (40,) max_iter: 100

Tabela A.2: Melhores parâmetros - Parte 2

Base	SVM	KNN	DT	MLP
Liver	kernel: rbf gamma: 0.001 C: 0.5	neighbors: 5	criterion: gini splitter: random min_samples_split: 2 min_samples_leaf: 1 max_depth: $None$ max_leaf_nodes: 10	learning_rate: constant hidden_layer_sizes: (100,) max_iter: 100
Magic	kernel: rbf gamma: 0.1 C: 0.9	neighbors: 7	criterion: gini splitter: best min_samples_split: 2 min_samples_leaf: 1 max_depth: $None$ max_leaf_nodes: 20	learning_rate: constant hidden_layer_sizes: (20, 20, 20) max_iter: 100
Mammo	kernel: linear gamma: $auto$ C: 0.6	neighbors: 17	criterion: gini splitter: random min_samples_split: 2 min_samples_leaf: 5 max_depth: 5 max_leaf_nodes: 20	learning_rate: constant hidden_layer_sizes: (20,) max_iter: 100
Monk	kernel: rbf gamma: 0.1 C: 1.0	neighbors: 13	criterion: gini splitter: random min_samples_split: 2 min_samples_leaf: 1 max_depth: 5 max_leaf_nodes: 5	learning_rate: constant hidden_layer_sizes: (40, 40) max_iter: 100
Phoneme	kernel: rbf gamma: 0.1 C: 0.9	neighbors: 7	criterion: gini splitter: best min_samples_split: 2 min_samples_leaf: 1 max_depth: 5 max_leaf_nodes: $None$	learning_rate: constant hidden_layer_sizes: (40,) max_iter: 200
Segmentation	kernel: linear gamma: $auto$ C: 0.3	neighbors: 1	criterion: gini splitter: best min_samples_split: 2 min_samples_leaf: 1 max_depth: $None$ max_leaf_nodes: 20	learning_rate: constant hidden_layer_sizes: (60, 60, 60) max_iter: 200
Sonar	kernel: linear gamma: $auto$ C: 0.4	neighbors: 1	criterion: entropy splitter: random min_samples_split: 20 min_samples_leaf: 1 ou 10 max_depth: 10 max_leaf_nodes: $None$	learning_rate: constant hidden_layer_sizes: (20, 20) max_iter: 100
Thyroid	kernel: linear gamma: $auto$ C: 0.9	neighbors: 7	criterion: gini splitter: best min_samples_split: 2 min_samples_leaf: 1 max_depth: 5 max_leaf_nodes: $None$	learning_rate: constant hidden_layer_sizes: (40,) max_iter: 100

Tabela A.3: Melhores parâmetros - Parte 3

Base	SVM	KNN	DT	MLP
Vehicle	kernel: linear gamma: $auto$ C: 0.4	neighbors: 5	criterion: gini splitter: random min_samples_split: 2 min_samples_leaf: 1 max_depth: $None$ max_leaf_nodes: 10	learning_rate: constant hidden_layer_sizes: (60, 60, 60) max_iter: 400
Vertebral	kernel: linear gamma: $auto$ C: 0.4	neighbors: 11	criterion: gini splitter: random min_samples_split: 10 min_samples_leaf: 1 max_depth: 10 max_leaf_nodes: $None$	learning_rate: constant hidden_layer_sizes: (100, 100, 100) max_iter: 100
WBC	kernel: poly gamma: $auto$ C: 0.1	neighbors: 9	criterion: gini splitter: random min_samples_split: 2 min_samples_leaf: 5 max_depth: 5 max_leaf_nodes: 10	learning_rate: constant hidden_layer_sizes: (100,) max_iter: 100
WDVG	kernel: rbf gamma: 0.01 C: 0.5	neighbors: 17	criterion: gini splitter: random min_samples_split: 2 min_samples_leaf: 10 max_depth: 10 max_leaf_nodes: $None$	learning_rate: constant hidden_layer_sizes: (40,) max_iter: 100
Weaning	kernel: rbf gamma: 0.01 C: 0.7	neighbors: 5	criterion: gini splitter: random min_samples_split: 20 min_samples_leaf: 1 max_depth: 5 max_leaf_nodes: 10 ou 20	learning_rate: constant hidden_layer_sizes: (60,) max_iter: 100
Wine	kernel: linear gamma: $auto$ C: 0.2	neighbors: 1	criterion: gini splitter: random min_samples_split: 10 min_samples_leaf: 1 max_depth: 5 max_leaf_nodes: 10	learning_rate: constant hidden_layer_sizes: (20, 20, 20) max_iter: 200

Tabela A.4: Melhores parâmetros - Parte 4

Bibliografia

[Alcalá-Fdez et al. 2008]ALCALá-FDEZ, J. et al. Keel: A software tool to assess evolutionary algorithms for data mining problems. *Soft Comput.*, Springer-Verlag, Berlin, Heidelberg, v. 13, n. 3, p. 307–318, out. 2008. ISSN 1432-7643. Disponível em: <http://dx.doi.org/10.1007/s00500-008-0323-y>.

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

[Breiman 1996]BREIMAN, L. Bagging predictors. *Mach. Learn.*, 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>.

[Dheeru e Taniskidou 2017]DHEERU, D.; TANISKIDOU, E. K. *UCI Machine Learning Repository*. 2017. Disponível em: <http://archive.ics.uci.edu/ml>.

[Gilpin e Dunlavy 2009]GILPIN, S. A.; DUNLAVY, D. M. *Heterogeneous Ensemble Classification*. [S.l.], 2009. Disponível em: <http://www.cs.sandia.gov/CSRI/Proceedings/>.

[Gunes et al. 2003]GUNES, V. et al. Combination, cooperation and selection of classifiers: A state of the art. *IJPRAI*, v. 17, n. 8, p. 1303–1324, 2003. Disponível em: <https://doi.org/10.1142/S0218001403002897>.

[Hassanat et al. 2014]HASSANAT, A. B. et al. Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach. *CoRR*, abs/1409.0919, 2014. Disponível em: <http://arxiv.org/abs/1409.0919>.

[Hsu, Chang e Lin 2010]HSU, C. wei; CHANG, C. chung; LIN, C. jen. *A practical guide to support vector classification*. 2010.

[Jain, Duin e Mao 2000]JAIN, A. K.; DUIN, R. P. W.; MAO, J. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 22, n. 1, p. 4–37, Jan 2000. ISSN 0162-8828.

[King, Feng e Sutherland 1995]KING, R.; FENG, C.; SUTHERLAND, A. Statlog: comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence*, v. 9, n. 3, p. 289–333, 5 1995.

[Kittler et al. 1998]KITTLER, J. et al. On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 20, n. 3, p. 226–239, 1998. Disponível em: <https://doi.org/10.1109/34.667881>.

[Kuncheva 2002]KUNCHEVA, L. I. A theoretical study on six classifier fusion strategies. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 24, n. 2, p. 281–286, fev. 2002. ISSN 0162-8828. Disponível em: <https://doi.org/10.1109/34.982906>.

[Kuncheva e Whitaker 2003]KUNCHEVA, L. I.; WHITAKER, C. J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, v. 51, n. 2, p. 181–207, 2003. Disponível em: <https://doi.org/10.1023/A:1022859003006>.

[Michie, Spiegelhalter e Taylor 1994]MICHIE, D.; SPIEGELHALTER, D. J.; TAYLOR, C. C. *Machine Learning, Neural and Statistical Classification*. [S.l.]: Ellis Horwood, 1994. ISBN 0-13-106360-X.

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

[Ponti Jr. 2011]Ponti Jr., M. P. Combining classifiers: From the creation of ensembles to the decision fusion. In: *24th SIBGRAPI Conference on Graphics, Patterns and Images - Tutorials, SIBGRAPI-T 2011, Alagoas, Brazil, August 28-30, 2011*. [s.n.], 2011. p. 1–10. Disponível em: <https://doi.org/10.1109/SIBGRAPI-T.2011.9>.

[Ranawana e Palade 2006]RANAWANA, R.; PALADE, V. Multi-classifier systems: Review and a roadmap for developers. *Int. J. Hybrid Intell. Syst.*, v. 3, n. 1, p. 35–61, 2006. Dis-

ponível em: <http://content.iospress.com/articles/international-journal-of-hybrid-intelligent-systems/his00022>.

[Ruta e Gabrys 2000]RUTA, D.; GABRYS, B. An overview of classifier fusion methods. *Computing and Information systems*, Bournemouth University, Fern Barrow, Poole, Dorset, BH12 5BB, UK, v. 7, n. 1, p. 1–10, 2000.

[Silva Jr. 2015]Silva Jr., E. J. da. *A two-step cascade classification method*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Paraná, 2015.

[Tan et al. 2009]TAN, P. et al. *Introdução ao datamining: Mineração de dados*. Ciencia Moderna, 2009. ISBN 9788573937619. Disponível em: <https://books.google.com.br/books?id=69d6PgAACAAJ>.

[Utgoff 1989]UTGOFF, P. E. Incremental induction of decision trees. *Machine Learning*, v. 4, n. 2, p. 161–186, Nov 1989. ISSN 1573-0565. Disponível em: <https://doi.org/10.1023/A:1022699900025>.

[Vriesmann 2013]VRIESMANN, L. M. *Combining overall and local class accuracies in an oracle-based method for dynamic ensemble selection*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Paraná, 2013.

[Weston e Watkins 1999]WESTON, J.; WATKINS, C. *Support Vector Machines for Multi-Class Pattern Recognition*. 1999.