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

Análise da relação entre tamanho e complexidade de um conjunto de dados

Mateus José dos Santos

CASCAVEL
2018

MATEUS JOSÉ DOS SANTOS

Análise da relação entre tamanho e complexidade de um conjunto de dados

Monografia apresentada como requisito parcial para obtenção 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. Dr. André Luiz Brun

CASCAVEL
2018

MATEUS JOSÉ DOS SANTOS

Análise da relação entre tamanho e complexidade de um conjunto de dados

Monografia apresentada como requisito parcial para obtenção 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. Dr. André Luiz Brun (Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

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

Prof[a]. M. Eng. Adriana Postal
Colegiado de Ciência da Computação,
UNIOESTE

Cascavel, 10 de dezembro de 2018

AGRADECIMENTOS

Agradeço primeiramente aos meus pais e meu irmão, mas principalmente aos meus pais Antônio e Adriana pelo apoio, por acreditarem em mim, por estarem ao meu lado nos momentos felizes e principalmente nos difíceis, por financiarem esses anos de graduação para que eu tivesse uma educação de qualidade.

A minha namorada, Carina, meu porto seguro, que durante os últimos três anos foi a pessoa que mais me apoiou, ajudou, acreditou em meu potencial. Por ser a pessoa que me fez amadurecer durante a graduação, e que nunca desistiu de mim. Pela motivação e por não me deixar desistir nos momentos mais complicados. Mas principalmente por sempre me dar um amor que me alegra, me abraça e faz me sentir o cara mais especial do mundo, me dando forças para sempre continuar em frente e nunca desistir.

Também a todos meus professores, pelo conhecimento adquirido, mas especialmente ao meu orientador André Luiz Brun, pela sua competência, por estar sempre presente nos momentos de dúvida e por nunca medir esforços para me auxiliar. Aos membros da minha banca Adair Santa Catarina e Adriana Postal pelo tempo gasto e pelos conhecimentos transmitidos.

A todos meus amigos, principalmente aos que fiz durante o tempo de graduação, os quais levarei para o resto da vida. Ao HU3, grupo de amigos da turma de 2013, amigos esses que guardarei no coração e os quais desejo todo sucesso do mundo e que tenhamos muitos momentos juntos mesmo após a graduação. Mas principalmente a um dos melhores amigos que tive dentro da graduação, Allysson Sanciani (Gordin, Sanci, Panci) por sempre me apoiar me ouvir, e me aconselhar e ao meu outro grande amigo Hamã (Hamã mesmo) pelos momentos felizes e tristes compartilhados, por ser meu ombro amigo nos momentos necessários, e por estar sempre presente.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

DCoL	*Data Complexity Library*
IA	Inteligência Artificial
KEEL	*Knowledge Extraction based on Evolutionary Learning*
KNN	*K-Nearest Neighbors*
LKC	*Ludmila Kuncheva Collection of Real Medical Data*
MST	*Minimum Spanning Tree*
NB	Naive Bayes
RNA	Rede Neural Artificial
RP	Reconhecimento de Padrões
SC	Subconjunto
SMC	Sistemas Multi Classificadores
SVM	*Support Vector Machine*
UCI	*University of California, Irvine*

Sumário

Resumo

Problemas de reconhecimento de padrões que apresentam classes muito similares com características pouco discriminantes são considerados complexos. Uma forma de tentar exprimir tal dificuldade são as métricas de complexidade as quais são sensíveis a alterações nos conjuntos de dados. Com base neste conceito, o objetivo deste trabalho foi efetuar a análise da relação entre o tamanho do conjunto usado no aprendizado de classificadores e seus respectivos índices de complexidade. Além disso, buscou-se identificar os descritores de complexidade que são mais robustos à variação do tamanho do conjunto. Para efetuar tal análise aplicou-se um protocolo experimental em que foram testadas várias dimensões (10%, 33%, 50% e 66%) de subconjunto de treino gerados pelos métodos de *Bagging* e *Boosting*. Para cada proporção foram gerados cem subconjuntos os quais tem sua assinatura de complexidade estimada. Levantou-se o índice de correlação para cada descritor de complexidade para cada proporção. Para validar os resultados, o protocolo foi aplicado sobre um conjunto de 26 bases de dados perante 20 repetições. Os resultados obtidos indicaram que a métrica de complexidade L1 possui um comportamento não muito definido, pois varia entre as faixas de correlação fraca e moderada. Já os descritores F1, F2, L2, N4, L3, T1, D2 e D3 não sofrem tanta influência do tamanho do conjunto onde são calculados, visto que o coeficiente de correlação obtido foi considerado fraco. Por outro lado, as medidas F3, F4, N1, N2 e N3 sofrem uma influência maior de acordo com a quantidade de instâncias presentes no conjunto uma vez que seu coeficiente de correlação apresentou comportamento moderado e forte.

Palavras-chave: Aprendizagem de Máquina, Reconhecimento de Padrões, Métricas de Complexidade, Correlação.

Capítulo 1

Introdução

O reconhecimento de padrões (RP) tem como uma de suas principais aplicações atribuir a um determinado objeto, uma classe entre várias possíveis. Este processo recebe o nome de classificação. O elemento responsável pela atribuição de um rótulo é chamado classificador. Estes classificadores são utilizados para apontar e descrever padrões ou objetos a partir de um conjunto de propriedades ou características. Por exemplo, podemos diferenciar dois animais pelo seu tamanho e seu habitat natural, tal como diferenciar uma baleia-azul de um urso-pardo. Ambos são mamíferos, porém a baleia-azul vive no mar e tem um tamanho médio de 25 metros enquanto um urso-pardo vive em terra e tem tamanho entre 70 e 150 cm.

O processo de preparar um classificador para o processo de reconhecimento é chamado de aprendizagem ou treinamento. O aprendizado de máquina é uma abordagem de análise de dados que, através de algoritmos, visa simular o conhecimento humano, ou seja, permitir que sistemas se adaptem de forma independente levando em consideração cenários anteriores, assim reproduzindo decisões e resultados confiáveis (TAN; STEINBACH; KUMAR, 2005).

Um sistema de aprendizado é uma estratégia geralmente desenvolvida em computador e que absorve conhecimento, com base em casos já conhecidos, e que toma decisões baseadas nas experiências acumuladas por meio de solução bem-sucedida de problemas anteriores.

Sabe-se que o desempenho de um sistema de reconhecimento é dependente do classificador utilizado. Da mesma forma, é consenso que o comportamento dos classificadores é dependente do conjunto de dados em que foram treinados. Por exemplo, digamos que se espera que um sistema seja capaz de identificar dois animais: cachorros e peixes. Como os peixes possuem características bastante distintas dos cachorros, o problema para separá-los é considerado simples.

Por outro lado, em um sistema de identificação entre gatos e cachorros, a tarefa torna-se

mais complexa, uma vez que neste caso, o problema envolve animais que possuem características similares, tais como o fato de ambos serem mamíferos, possuírem 4 patas, terem 2 olhos, focinho, pelos... etc.

Para tentar estimar o quão complexo são os dados e quão difícil será a tarefa do classificador, alguns estimadores foram propostos na literatura: as métricas de complexidades (HO, 1998) (SÁNCHEZ; MOLLINEDA; SOTOCA, 2007). Estas métricas tentam, através de índices, descrever o grau de complexidade do conjunto usado no treino.

Muitas vezes a tarefa de classificação envolve muita variabilidade ou complexidade, fazendo com que um classificador individual não seja capaz de aprender efetivamente sobre todo o espaço de busca ou ser capaz de identificar as classes com precisão. Uma estratégia para tentar resolver tal problema é a adoção de sistemas de múltiplos classificadores (SMC).

Os SMC podem ser considerados uma das técnicas de aprendizado mais robustas e precisas. São utilizados para melhorar a performance de classificadores não tão robustos (PONTI JR., 2011) através da combinação das opiniões de diversos indutores esperando que o resultado obtido seja mais acurado.

Uma boa estratégia para a tentativa de construção de classificadores, que sejam bons em diferentes regiões do espaço de busca, é treinar os classificadores em conjuntos distintos e de tamanhos variados. Por exemplo, um classificador pode ser treinado para identificar uma flor de acordo com o tamanho de sua pétala, pela cor da planta ou mesmo com base no formato do caule. Dentro do conjunto de classificadores treinados, haverá aqueles que serão mais hábeis em identificar a cor, diferenciar melhor o tamanho da pétala e outros o formato do caule. Espera-se então que, ao combinar-se os três tipos de especialidade, o resultado alcançado seja melhor.

Um SMC pode ser considerado homogêneo ou heterogêneo. Nos sistemas homogêneos, as mesmas técnicas de indução são utilizadas para modelar os classificadores, podendo variar o conjunto de características. Já nos sistemas de múltiplos classificadores heterogêneos as técnicas utilizadas para treinar os classificadores são diferentes, mas os dados são os mesmos (VRIESMANN, 2012). Dentre as abordagens homogêneas algumas técnicas destacam-se na geração de subconjuntos distintos para a etapa de treino. Entre elas podemos citar o *Bagging*, *Boosting* e *Randon Subspace*.

Os subconjuntos gerados pelo processo de *Bagging* e *Boosting* afetarão o desempenho do

classificador. Por exemplo, reutilizando o cenário onde se busca identificar dois animais: cachorros e gatos. Quando se utilizar uma das técnicas citadas para geração de subconjuntos com instâncias de ambos os animais, é possível que neste subconjunto estejam presentes instâncias com características análogas, tais como a presença de pelo, o tamanho (há gatos e cães com tamanhos similares) e o peso. Seguindo este exemplo, como todas as características citadas anteriormente são similares entre si, a dificuldade de o classificador identificar essas instâncias será maior, ou seja, diminuirá sua acurácia.

Esta dificuldade na classificação pode ser estimada pelas métricas de complexidade (HO, 1998) (SÁNCHEZ; MOLLINEDA; SOTOCA, 2007), através das quais tenta-se determinar o grau de dificuldade de uma solução para um determinado problema. Tais descritores, no entanto, são sensíveis às alterações nos conjuntos. Um ponto importante, que ainda carece de estudos, é tentar identificar quais índices de complexidade são mais robustos às alterações na dimensão do conjunto de treino.

Dessa forma, o objetivo principal deste trabalho foi tentar identificar se existe relação entre o tamanho do conjunto de treino com sua assinatura de complexidade. Por exemplo, verificou-se se um conjunto composto de 300 rosas e 300 margaridas seria mais simples ou mais complexo que um conjunto formado por 100 rosas e 100 margaridas. O intuito foi identificar quais métricas seriam mais suscetíveis às variações no tamanho do conjunto de treino e quais as mais estáveis.

O Capítulo 2 trata dos conceitos sobre reconhecimento de padrões e métricas de complexidade, apresentando a ideia básica e funcionamento dos mesmos. O Capítulo 3 detalha o *framework* desenvolvido para a realização do trabalho e alcance da proposta da pesquisa. Em seguida no Capítulo 4 serão apresentados os experimentos realizados e resultados obtidos e no Capítulo 5, as conclusões obtidas durante o desenvolvimento dos experimentos.

Capítulo 2

Fundamentação Teórica

Há muito levanta-se a questão de que se algum dia os computadores serão capazes, assim como os seres humanos, de aprender. Para tentar responder a esta questão são desenvolvidos algoritmos que tentam aproximar ao máximo o aprendizado da máquina ao aprendizado humano. Tais métodos não tem a mesma capacidade que uma pessoa tem para aprender, porém eles possuem uma certa eficiência em determinadas áreas (MITCHEL, 1997). Uma destas áreas é o reconhecimento de padrões.

Reconhecimento de padrões é uma área da Inteligência Artificial que tem como principal objetivo utilizar algoritmos que possam classificar determinados objetos em classes. Estes objetos podem ser seres vivos, veículos, formas de onda (musicas, vozes, ruídos) ou qualquer outro tipo de medida que possa ou precisa ser classificada (SILVA JR., 2015).

O classificador tem como objetivo atribuir uma classe, com base em um conjunto de características, a uma determinada instância. Para a realização da classificação das instâncias são utilizados os algoritmos de aprendizado de máquina (VRIESMANN, 2012)

A classificação é realizada com base em conjuntos de padrões previamente classificados, os quais são chamados de Conjunto de Treinamento. Esse aprendizado é chamado de "aprendizado supervisionado". Há também a classificação não supervisionada, na qual os conjuntos de treino não são previamente classificados, ou seja, não se conhece um padrão esperado de classificação.

2.1 Funcionamento do processo de classificação

Como dito anteriormente, para a realização da classificação das novas instâncias são utilizados estratégias de classificação. Alguns dos principais algoritmos são: KNN (*K-Nearest*

Neighbor), SVM (*Support Vector Machine*), RNA (Rede Neural Artificial), Naive Bayes, Árvores de Decisão, entre outros. Nas seções seguintes serão apresentados mais detalhes de cada abordagem.

2.1.1 K-Vizinhos Mais Próximos - KNN

Um dos algoritmos mais simples utilizados no aprendizado de máquina é o *K-Nearest Neighbor* ou KNN. Este algoritmo realiza a classificação de uma instância baseado nos vizinhos mais próximos em um espaço de características (ALTMAN, 1992). Esta classificação é supervisionada e realizada em duas etapas.

Na primeira etapa é necessário um conjunto de instâncias com suas características e, para cada instância, uma classe destino. Após o treinamento a classificação de cada instância será feita, de modo que os valores das instâncias serão testados e atribuídos para uma determinada classe.

O KNN irá encontrar todos os K elementos que estão mais próximos da instância desejada e atribuir a classe mais frequente dentro deste conjunto de K elementos. Dessa forma é preferível que K sempre seja um valor ímpar, pois assim haverá chance menor de ambiguidade na classificação. Por exemplo, se tomarmos K como 8 é possível que existam quatro vizinhos em uma classe e quatro em outra.

2.1.2 Árvore de Decisão

A classificação por árvore de decisão também utiliza treinamento de forma supervisionada. Este algoritmo é uma função que tem como entrada valores contínuos, discretos e nominais e tem como retorno uma decisão (RUSSELL; NORVIG, 2003). Esta decisão é definida de forma que ela consiga separar as classes da melhor forma possível.

Em uma árvore de decisão cada nodo interno correspondente a um valor de teste de um atributo de entrada A_i, de forma que cada folha do nodo A_i são seus possíveis valores, (RUSSELL; NORVIG, 2003).

Para cada atributo é definida uma regra de decisão. Aquela que for a mais eficiente na divisão das instâncias será a adotada (QUINLAN, 1986). Para determinar a eficiência da decisão emprega-se a entropia, que mede o grau de confusão de conjuntos.

Para características ruidosas este método é robusto, porém, quando os dados apresentam grande variabilidade, ela não tem o mesmo efeito, (SILVA JR., 2015).

2.1.3 Máquina de Vetores de Suporte - SVM

O algoritmo de Máquina de Vetores de Suporte (SVM, *Support Vector Machine*) é uma das abordagens mais populares para aprendizagem supervisionada (RUSSELL; NORVIG, 2003). O SVM padrão toma como entrada um conjunto de dados e estima, para cada entrada dada, qual de duas possíveis classes ela faz parte, o que o torna um classificador binário não probabilístico.

De maneira mais simples, o SVM encontra uma linha de separação, também conhecida como hiperplano entre os dados de duas classes. Essa fronteira procura maximizar a distância entre os pontos mais próximos de cada uma das classes.

Quando encontra-se um problema não linearmente separável, o conjunto de treinamento é reestruturado do seu espaço original para um novo espaço com uma dimensão maior. Para efetuar a transformação de espaço de características é necessário o uso de uma função Kernel ou *Kernel-Trick* que consiste na aplicação de uma função não linear (RUSSELL; NORVIG, 2003).

O objetivo do *Kernel-Trick* é transformar um problema que inicialmente era não linear em um novo problema que é linearmente separável , através de um hiperplano ótimo (SCHöLKOPF, 2000).

O SVM funciona muito bem com dados de alta dimensão, conseguindo suportar o problema da dimensionalidade. Outro ponto positivo desse algoritmo é o fato de que ele representa o limite de decisão usando um subconjunto dos exemplos de treinamento, conhecidos como vetores de suporte (TAN; STEINBACH; KUMAR, 2005).

2.1.4 Redes Neurais Artificiais

O algoritmo de Redes Neurais Artificiais (RNA) é baseado no sistema nervoso humano, de forma que as informações são processadas através de neurônios interligados (BISHOP, 1995). Desta maneira elas adquirem conhecimento através da experiência.

Uma RNA é composta por várias unidades de processamento (neurônios), as quais são normalmente conectadas por canais de comunicação que são associados a determinado peso. As

unidades realizam operações somente sobre seus dados locais, que são entradas recebidas pelas suas conexões.

As operação de uma unidade de processamento funciona de maneira que os sinais são apresentados as entradas, então cada sinal recebe um peso que indica sua influência na saída. Após essa atribuição de peso é realizada uma soma ponderada dos sinais que produz um nível de atividade. Por fim, caso o nível de atividade exceda um certo limiar, a unidade então produz uma determinada resposta de saída (McCULLOCH, 1943).

Uma RNA pode conter várias camadas entre as camadas de entradas e saídas. Essas camadas intermediárias são ditas *escondidas* e os nós presentes nestas, são chamados de *nós escondidos*.

As camadas intermediárias são geralmente utilizadas para a realização de um gargalo, de forma que a rede gere um modelo simples e que seja capaz de generalizar padrões desconhecidos (MITCHEL, 1997).

2.1.5 Naive Bayes

O Naive Bayes (NB) é um classificador probabilístico baseado no teorema de Bayes. Este algoritmo utiliza dados de treino para formar um modelo probabilístico baseado na evidência das características nos dados. Desta forma pode-se dizer que o NB se baseia na frequência das características (MICHIE et al., 1994).

De uma maneira um pouco menos formal, podemos dizer que o algoritmo evidencia que um recurso não está associado com a existência de uma característica particular em uma classe. Por exemplo, o fato de que um cachorro possui quatro patas, dois olhos, pelos, focinho e boca. Mesmo sabendo que cada exemplar destes atributos precisa dos outros, ainda sim cada um tem seu peso para a identificação do cachorro.

2.2 Classificador Monolítico

Os classificadores são utilizados para rotular ou descrever padrões ou objetos a partir de um conjunto de propriedades ou características.

Um sistema de classificação monolítico é aquele que possui apenas um único elemento, que é responsável por classificar todas as instâncias.

Na prática, no entanto, estes sistemas geralmente não conseguem absorver toda a variabilidade do problema, ou seja, em um caso de classificação onde há muito ruído, e poucos dados um único classificador acaba não sendo o suficiente (SILVA JR., 2015).

A criação de um único classificador para atender a variabilidade presente na maioria dos problemas de reconhecimento de padrões é uma tarefa desafiadora (BRITTO JR.; SABOURIN; OLIVEIRA, 2014). Para tentar absorver essa variabilidade nos problemas foram propostos os Sistemas de Múltiplos Classificadores.

2.3 Sistemas de Múltiplos Classificadores

Os sistemas de múltiplos classificadores ou SMC são considerados uma das abordagens de aprendizado mais robustas e precisas (PONTI JR., 2011). O SMC tem sido aplicado com sucesso em uma gama muito grande de problemas reais e é constantemente usado para melhorar a performance de classificadores mais fracos.

A ideia é formar um conjunto composto por diversos classificadores e, conforme são inseridas novas instâncias a serem classificadas, todos os elementos dão sua opinião sobre a classe do novo padrão. Essas opiniões são então combinadas segundo algum critério para formar um consenso quanto à classe da instância.

Um SMC pode ser construído por métodos homogêneos ou heterogêneos. Em um conjunto homogêneo utiliza-se a mesma técnica de classificação, variando-se o conjunto de instâncias usadas no treinamento ou o conjunto de características. Já os sistemas heterogêneos se utilizam de diferentes algoritmos de aprendizagem e mantêm os dados fixos, variando somente o classificador (KUNCHEVA, 2004).

Uma grande vantagem do SMC é a possibilidade de que, ao se utilizar opiniões diferentes, seja alcançada maior diversidade na classificação, para que então seja possível se obter uma maior precisão na classificação. Este processo de combinação é também chamado de *ensemble* e é normalmente utilizado em problemas mais complexos (KITTLER et al., 1998).

2.3.1 Métodos de Geração de Subconjuntos

Como constatado anteriormente, uma das estratégias de construir um *ensemble* de classificadores é através da abordagem homogênea. A seguir são apresentadas três estratégias para

formar um conjunto homogêneo de classificadores. As duas primeiras (*Bagging* e *Boosting*) focam na variação das instâncias de treino enquanto a terceira mantém todo o conjunto de treino e varia apenas o conjunto de características destes.

2.3.1.1 *Bagging*

O *Bagging* é uma abordagem proposta por Breiman (BREIMAN, 1996), que consiste em gerar, a partir de um conjunto de dados original, de forma aleatória e com reposição, subconjuntos de treino distintos. Assim é possível obter certa diversidade nos subconjuntos de treinos gerados.

O processo consiste em sortear vários subconjuntos com tamanho pré-definido. Cada um pode conter instâncias repetidas, as quais podem, inclusive, pertencerem a dois subconjuntos distintos.

A estratégia, apesar de aleatória, preserva a proporcionalidade entre as classes, mantendo a estratificação original dos dados.

Um exemplo do funcionamento do método é detalhado na Figura 2.1. A Figura 2.1-1 é o conjunto de treino, o qual possui 6 círculos e 4 triângulos. Cada um possui seu identificador (números para os círculos e letras para os triângulos). Após aplicação do *Bagging* (Figura 2.1-2) com 50% do tamanho do conjunto de treino, são gerados os subconjuntos (Figura 2.1-3) SC_1, SC_2 e SC_N sendo n o número total de conjuntos a serem gerados.

É possível notar que os subconjuntos gerados mantiveram a estratificação original dos dados, escolhendo de forma aleatória três círculos e 2 triângulos.

2.3.1.2 *Boosting*

O *Boosting* segue o mesmo princípio do Bagging, porém a diferença está no fato de que nele as escolhas são feitas a partir de um peso dado a cada instância.

A ideia é sortear um conjunto de elementos aleatoriamente onde inicialmente todos possuem o mesmo peso. Assim que a classificação das amostras sorteadas para formar um conjunto é feita, verifica-se quais foram classificadas incorretamente. Após esta verificação essas mesmas amostras terão seu peso aumentado de forma que em um sorteio seguinte, elas tenham mais chances de ser selecionadas (FREUND, 1996). Dessa forma, as instâncias mais difíceis terão mais foco na composição do conjunto de treinamento.

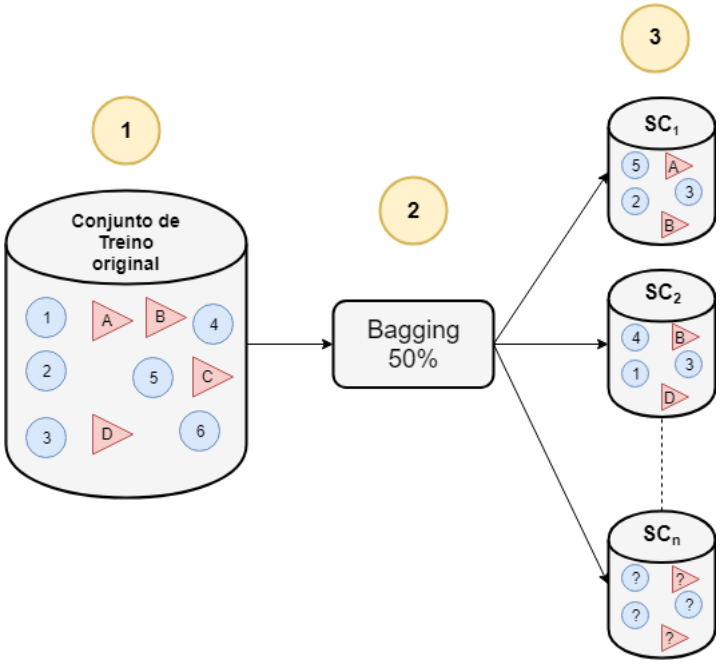

Figura 2.1: Exemplo do funcionamento do *Bagging*

É possível entender um pouco mais o funcionamento do *boosting* através da Figura 2.1. De modo que 2.2-1 é o mesmo conjunto de treino da Figura 2.1-1. Aplica-se então o método Bagging novamente (Figura 2.2-2 porém gerando um subconjunto por vez, assim temos o novo subconjunto gerado 2.2-3. Em seguida classifica-se o subconjunto formado (usando o KNN 2.1.1) Figura 2.2-4. Na sequência, identifica-se as instâncias que foram classificadas erroneamente (Figura 2.2-5) aumentando seu peso (Figura (2.2-6)) para próxima iteração do processo (a partir do Bagging Figura 2.2-2). A cada iteração obtém-se um novo subconjunto. Assim sendo, o algoritmo é executado N vezes, sendo N o número de classificadores a serem treinados.

2.3.1.3 *Random Subspace* - RSS

A estratégia proposta pelo *Random Subspace* (HO, 1998) possui ideia similar aos outros dois citados anteriormente, porém ao invés da seleção randômica sobre as instâncias presentes no conjunto de treino, ela será randômica sobre as características de cada instância.

Por exemplo, supondo que um conjunto de treino possua N instâncias com K atributos, aplicando o RSS pode-se criar X subconjuntos com todas as N instâncias em todos os X subconjuntos. Todavia, cada instância terá um número pré definido de características e essas serão selecionadas randomicamente.

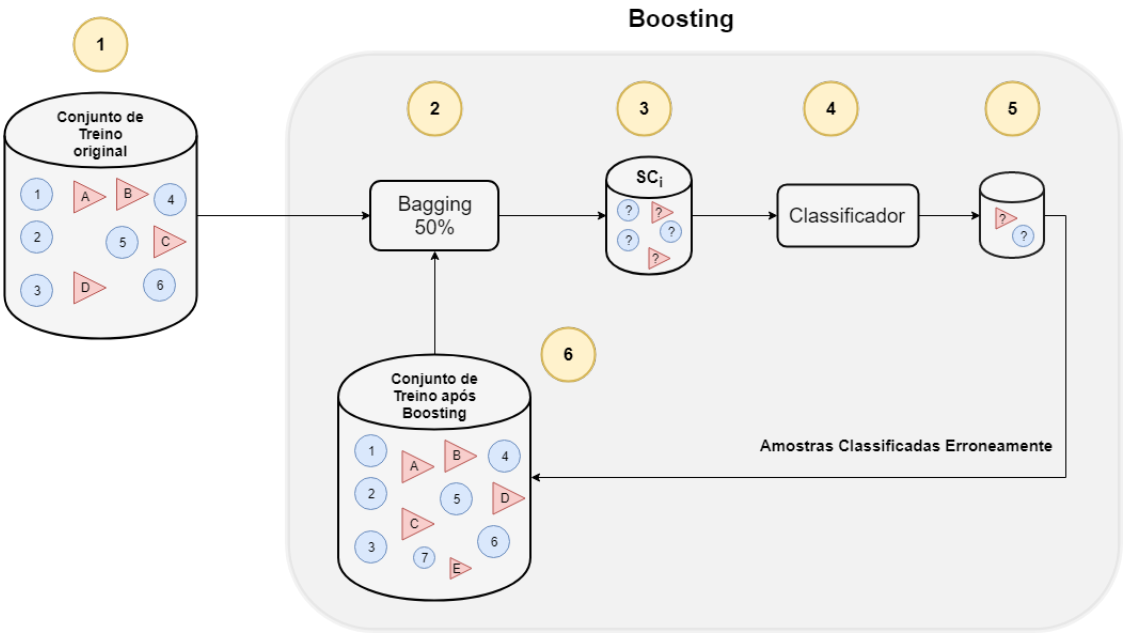

Figura 2.2: Exemplo do funcionamento do *Boosting*

Uma representação simbólica para este algoritmo é apresentada na Figura 2.3. Na ilustração tem-se um conjunto de treino (Figura 2.3-A) com duas instâncias de classes diferentes (quadrado e círculo) com quatro atributos cada (A_1, A_2, A_3 e A_4). É então aplicado o RSS com proporção 50% (Figura 2.3-B). São gerados N subconjuntos com combinações dos atributos de cada instância.

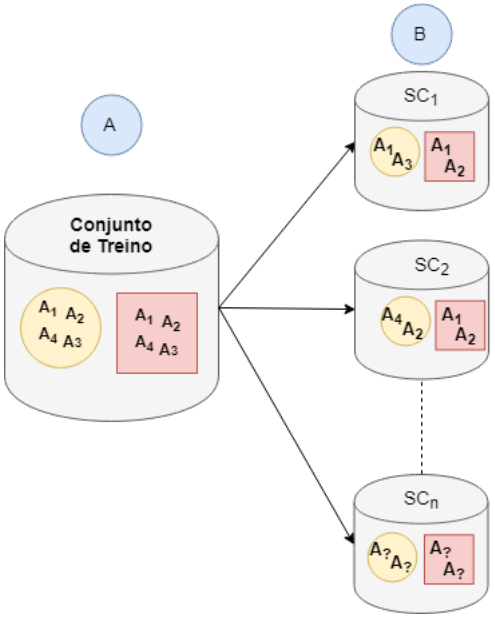

Figura 2.3: Exemplo do funcionamento do *Random Subspace*

Os dados do conjunto do treino têm uma grande influência no comportamento do classificador. Caso esses dados sejam extraídos de forma inapropriada, ao final pode-se ter uma acurácia baixa na classificação (MACIà; ORRIOLS-PUIG; BERNADó-MANSILLA, 2010).

Uma abordagem para tentar descrever melhor o comportamento dos dados seria analisar a complexidade do problema. Para tanto, foram propostos na literatura diversos estimadores com o intuito de tentar caracterizar nível o de dificuldade do problema em estudo. São chamadas medidas de complexidade, as quais serão exploradas com maior detalhamento na seção seguinte.

2.4 Métricas de Complexidade

A complexidade de um problema está relacionada à dificuldade de classificação do mesmo, baseado nas instâncias e atributos que o compõem. Dessa maneira, a complexidade tem relação com o desempenho da classificação, pois caso os dados tenham uma complexidade grande tem-se a possibilidade de que os classificadores não consigam classificar os novos padrões de forma efetiva.

Por exemplo, considere um conjunto de dados do qual é necessário escolher 5 instâncias de cada classe (5 círculos e 5 “x”, como apresentado na Figura 2.4 (a).

Digamos, no primeiro caso, que o *Bagging* ou *Boosting* escolham as instâncias circuladas em azul, como na Figura (2.4 (b)) para formar um subconjunto de treino.

Neste caso, o problema será simples e o classificador terá boa acurácia. Pode-se perceber que os atributos comprimento e peso das instâncias selecionadas, são bastante distintos entre os dois grupos. Tal discrepância será estimada através de um descritor de complexidade.

No terceiro exemplo, Figura (2.4 (c)), são selecionadas instâncias que estão bem mais próximas no espaço de características (Comprimento x Peso). Este problema é mais complexo e, consequentemente, a acurácia será menor. O índice de complexidade deste grupo mostrará valores mais acentuados, caracterizando o problema como mais difícil.

Visando estimar a complexidade de um problema foram propostas as medidas de complexidade, as quais são divididas em três categorias (HO; BASU, 2000) (HO; BASU, 2002) (SÁNCHEZ; MOLLINEDA; SOTOCA, 2007), que são: Sobreposição (*overlap*) das classes (F1, F2, F3, F4), Separabilidade das classes (L1, L2, N1, N2, N3) e Medidas de geometria, topologia e densidade (L3, N4, T1, T2, D1, D2, D3). Cada um dos grupos será mais bem detalhado a

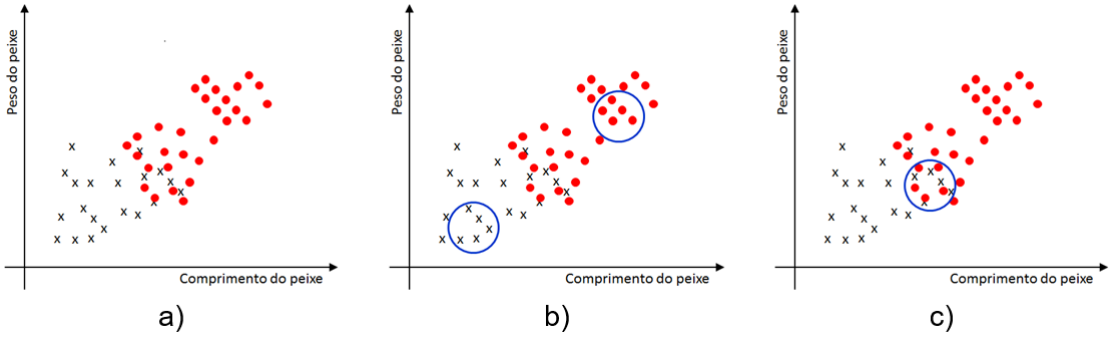

Figura 2.4: Exemplo de estimação de complexidade

seguir.

Todas as medidas de complexidade apresentadas realizam a estimação apenas para problemas de duas classes. Em cenários onde estão presentes mais de duas classes é necessária a adoção da estratégia OVA (*One Vs All*). Esta abordagem separa uma classe das demais, tornando o problema em um cenário dicotômico. Por exemplo, em um problema onde estão presentes as classe A, B e C, a estratégia OVA forma três novos problemas: A versus BC, B versus AC e C versus AB.

2.4.1 Medidas de Sobreposição

Medidas de sobreposição são aquelas com foco na eficácia na separação das classes com características de uma dimensão (SÁNCHEZ; MOLLINEDA; SOTOCA, 2007). Elas analisam o intervalo e a distribuição de valores no conjunto de dados em relação a cada recurso e estimam a sobreposição entre diferentes classes.

2.4.1.1 Relação Máxima do Discriminante de Fischer (F1)

Esta medida de sobreposição estima o quão separados são duas classes de acordo com características específicas (SÁNCHEZ; MOLLINEDA; SOTOCA, 2007).

É possível interpretar F1 como a distância entre o centro de duas classes. A separação entre elas é definida através de um índice, de forma que quanto maior ele for, mais distantes espera-se que as classes sejam uma da outra (LANDEROS, 2008).

Para calcular-se o índice é realizado um cálculo que compara as médias e desvio-padrões das classes para cada um de seus atributos, assim é possível medir a distância entre elas.

Os cálculos são realizados utilizando a Equação 2.1, onde μ_1, μ_2 e α_1, α_2 são as médias das duas classes e suas variâncias, respectivamente (SÁNCHEZ; MOLLINEDA; SOTOCA, 2007).

$$F1_i = \frac{(\mu_{1_i} - \mu_{2_i})^2}{{\sigma_{1_i}}^2 + {\sigma_{2_i}}^2} \tag{2.1}$$

Porém a Equação 2.1 é utilizada para casos onde há somente duas classes. Para casos mais gerais, onde existem N (N>2) classes utiliza-se a Equação 2.2.

$$N = \frac{\sum_{i=1}^{C} n_i.\delta(\mu, \mu_i)}{\sum_{i=1}^{C} \sum_{j=1}^{n_i} \delta(x_j^i, \mu_i)} \tag{2.2}$$

Onde n_i denota o número de instâncias da classe *i*, δ é uma medida (normalmente utiliza-se a distância Euclidiana), μ_i é a média da classe e x_j^i representa o elemento *j* que pertence a classe *i*.

A Figura 2.5 representa de maneira simples a ideia do funcionamento do Discriminante de Fischer. É possível identificar o centroide das classes vermelha e azul e então a medida (d_1) da distância entre o centro de ambas as classes.

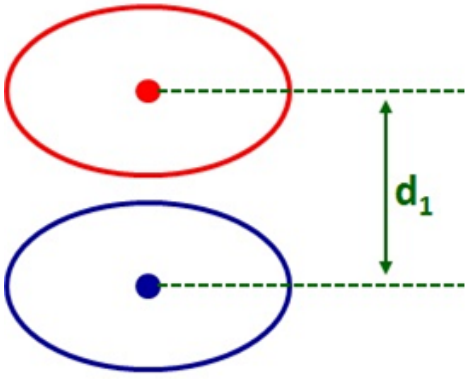

Figura 2.5: Funcionamento do Discriminante de Fischer (D1) em classes linearmente separáveis, adaptado de Landeros(2008)

2.4.1.2 Sobreposição de Atributos por Classe (F2)

O descritor F2 estima a sobreposição de duas classes considerando apenas uma característica por vez. É possível calcular esta sobreposição encontrando, para cada característica, seus valores máximos e mínimos. Em seguida calcula-se o tamanho da distância da região que é com-

partilhado pelas duas classes (sobrepostas) dividido pelo valor total entre mínimo e máximo do atributo (HO; BASU, 2002).

Para que a sobreposição total de duas classes possam ser determinadas, é necessário calcular F2 para cada uma das características do conjunto e então multiplicá-las, como mostrados na Equação 2.3.

$$F2 = \prod_{i=1}^{d} \frac{MIN(max(f_i, c_1), max(f_i, c_2)) - MAX(min(f_i, c_1), min(f_i, c_2))}{MAX(max(f_i, c_1), max(f_i, c_2)) - MIN(min(f_i, c_1), min(f_i, c_2))} \tag{2.3}$$

Onde *i* é o número da característica sendo verificadas, *d* indica a quantidade de atributos, f_i corresponde à característica *i* e c_i à classe i.

Para que o valor de F2 seja zero, é necessário que não haja sobreposição para um atributo, levando em conta o fato de a Equação 2.3 ser um produtório (HO; BASU, 2002).

2.4.1.3 Eficiência Máxima por Atributo Individual (F3)

Em problemas de alta dimensionalidade recomenda-se entender como as informações discriminantes são distribuídas de acordo com os atributos. Baseado nisso, F3 é considerado uma medida de eficiência de características individuais que descrevem o quanto cada uma contribui para a separação de duas classes (HO; BASU, 2002).

A eficiência de cada característica é estimada pelo percentual de instâncias que podem ser separadas de acordo com ela. O índice F3 será definido pela maior separabilidade obtida entre todos os atributos do problema em estudo (HO; BASU, 2002). Quanto maior o valor, espera-se que melhor separadas estejam as classes.

2.4.1.4 Eficiência Coletiva dos Atributos (F4)

A ideia desta medida é similar àquela apresentada pela medida F3, porém leva em consideração o poder discriminante de todo o conjunto de atributos.

Para calcular o poder discriminante coletivo, o seguinte procedimento é realizado: seleciona-se o atributo que consegue separar o maior número de instâncias de uma classe. Dessa forma, todas as instâncias que puderam ser separadas são removidas do conjunto de dados. Em seguida, o próximo atributo mais discriminante é então selecionado e é realizada a separação dos elementos classificados, os quais são removidos do conjunto. O processo se repete até que

todas as instâncias possam ser classificadas ou até que todos os atributos tenham sidos analisados (MACIà; ORRIOLS-PUIG; BERNADó-MANSILLA, 2010). O valor de F4 é obtido pelo percentual de instâncias do conjunto que puderam ser discriminadas.

2.4.2 Medidas de Separabilidade

As medidas de separabilidade avaliam o quanto duas classes são separadas, ou seja, o quão complexa é a região de fronteira entre duas classes. Para isso, tais métricas descrevem a complexidade do comportamento dos conjuntos nessa região.

2.4.2.1 Soma Minimizada da Distância de Erro de um Classificador Linear (L1)

L1 verifica o quão linearmente separáveis são os dados do conjunto (HO; BASU, 2002) (MOLLINEDA; SÁNCHEZ; SOTOCA, 2006).

Para realizar esta checagem primeiramente é construído um classificador linear ótimo (SVM) para que sejam minimizados os erros na separação das duas classes. Após a criação do classificador, calcula-se L1 através da soma das distâncias das amostras classificadas erroneamente até a fronteira linear construída pelo classificador (HO; BASU, 2000), (MOLLINEDA; SÁNCHEZ; SOTOCA, 2006) assim como mostra a Equação 2.4.

$$L1 = \sum \delta(C(x_i^-), x_i) \tag{2.4}$$

Onde δ é a distância euclidiana entre a fronteira construída pelo classificador linear *C* e a instância x_i, classificada de forma incorreta pelo classificador ($C(x_i^-)$), tal como mostrado na Figura 2.6 a qual tem o classificador linear representado pela reta em azul, evidenciando duas instâncias que foram classificadas erroneamente.

Caso L1 tenha valor igual a zero então as classes são linearmente separáveis. De outra forma, quanto maior o valor de L1 mais complexo é o conjunto de amostras, tornando assim a separação linear menos eficiente.

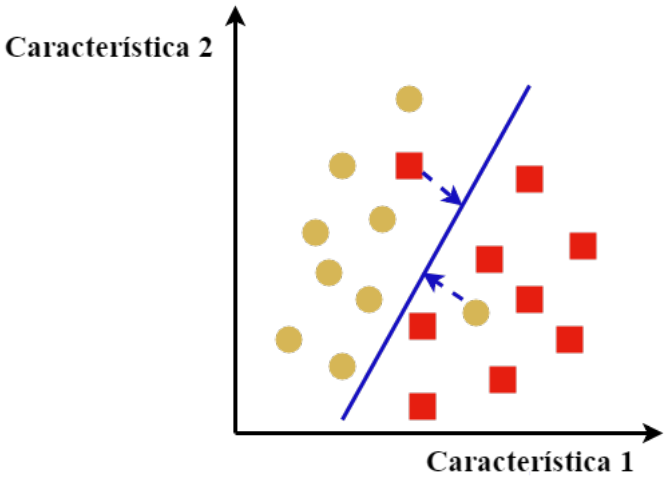

Figura 2.6: Separação linear ineficiente devido a classificação errônea das duas instâncias em destaque, adaptado de Landeros(2008)

2.4.2.2 Taxa de Erro de um Classificador Linear sobre o Treino (L2)

Esta medida é utilizada para representar a taxa de erro obtida utilizando-se de um classificador linear ótimo (SVM) sobre os dados de treino (HO; BASU, 2000), (MOLLINEDA; SÁNCHEZ; SOTOCA, 2006). O princípio do índice L2 é verificar quantas amostras estão posicionadas na região correspondente a uma classe utilizando o mesmo classificador criado para L1. Calcula-se o índice dividindo o número de elementos classificados erroneamente pelo número total de instâncias, tal como a Equação 2.5.

$$L2 = \frac{contagem(C(c_i^-))}{n} \tag{2.5}$$

Se o valor de L2 for zero significa que as classes são linearmente separáveis. Por outro lado, quanto mais próximo de 1 for o valor de L2, menos linearmente separáveis elas serão (BRUN, 2017).

2.4.2.3 Fração de Pontos na Região de Fronteiras (N1)

Este método é baseado na construção de uma árvore de cobertura miníma (*MST - Minimum Spanning Tree*), a qual conecta todos os pontos do conjunto de dados de forma a minimizar a soma das distâncias. Caso um ponto de uma classe esteja conectado a outra classe então diz-se que esse ponto é um elemento de fronteira entre as classes.

Calcula-se o valor de N1 através da relação de elementos conectados a outra classe pelo número de elementos total do conjunto como mostra a Equação 2.6 (HO; BASU, 2002).

$$N1 = \frac{contagem(x_i \neq x_j)}{n} \tag{2.6}$$

Onde $x_i \neq x_j$ são os elementos que estão ligados com instâncias de classes diferentes, enquanto *n* refere-se a quantidade de elementos presentes no conjunto.

2.4.2.4 Proporção das Distâncias Intra/Inter classes até o vizinho mais próximo (N2)

Esta medida compara a distância média de todos os elementos mais próximos contidos em uma classe com a distância dos vizinhos mais próximos que não estão contidos na classe (BRUN, 2017).

O intuito é calcular, a partir de cada elemento da classe, sua distância euclidiana até o vizinho mais próximo dentro da mesma classe e sua distância até o mais próximo que não está contido na classe. Então as distâncias entre os elementos de mesma classe são somadas e posteriormente divididas pela soma das distâncias entre os elementos de outra classe (HO; BASU, 2002), (MOLLINEDA; SÁNCHEZ; SOTOCA, 2006). A Equação 2.7 demonstra como o cálculo é realizado.

$$N2 = \frac{\sum_{i=1}^{n} \delta(N_1^{=}(x_i), x_i)}{\sum_{i=1}^{n} \delta(N_1^{\neq}(x_i), x_i)} \tag{2.7}$$

Onde $\delta(N_1^{=}(x_i), x_i)$ é a distância entre a instância *i* e seu vizinho de mesma classe o qual está o mais próximo, e $\delta(N_1^{\neq}(x_i), x_i)$ representa a distância da instância *i* até o elemento mais próximo que está contido em outra classe.

2.4.2.5 Taxa de erro do classificador KNN pela abordagem *Leave-One-Out* (N3)

A medida N3 corresponde à taxa de erro de um classificador KNN (*K-Nearest-Neighbor*) usando uma vizinhança de uma unidade sobre a classe (MOLLINEDA; SÁNCHEZ; SOTOCA, 2006).

Quanto mais próximo forem os pontos de classes diferentes maior tende a ser a porcentagem de erro de um classificador que utiliza o método do vizinho mais próximo (MOLLINEDA; SÁNCHEZ; SOTOCA, 2006). O percentual de erro é estimado pelo método *leave-one-out*.

Dessa forma quando os valores de N3 forem baixos tem-se uma lacuna entre os elementos das bordas das classes. Entretanto, quando os valores forem altos considera-se que haverá sobreposição nas regiões fronteiriças.

O *K-fold* (K-subconjuntos) é um método de validação cruzada que consiste na divisão do conjunto total de dados em dois novos subconjuntos mutuamente exclusivos, onde um é usado para testes e os $K - 1$ restantes para treinamento. O processo é realizado K vezes. Ao fim de todas as iterações, é calculada a acurácia sobre os erros encontrados (Equação 2.8). O *leave-one-out* é um método derivado do *K-fold* onde K é igual ao número total de dados N. Para esta abordagem realizam-se N cálculos de erro, um para cada dado.

$$Acc = \frac{1}{k} \sum_{j=1}^{k} \sum_{i=1}^{\frac{n}{k}} (\omega_i, \hat{\omega}_i) \tag{2.8}$$

Onde k é o número de *folds*, $\frac{n}{k}$ corresponde à quantidade de instâncias em cada *fold*, ω_i refere-se à classe real da instância e, por fim, $\hat{\omega}_i$ é a classe predita pelo classificador.

2.4.3 Medidas de Geometria, Topologia e Densidade

As medidas de Geometria, Topologia e Densidade visam descrever a geometria ou a forma das variações abrangidas por cada classe de forma a oferecer compreensão mais espacial do relacionamento das classes.

2.4.3.1 Fração de Esferas de Cobertura Máxima (T1)

Esta medida conta o número de círculos necessários para cobrir cada classe, onde cada círculo é centralizado em cada uma das instâncias e crescem até um ponto onde tocam instâncias de outra classe (HO; BASU, 2002), (MOLLINEDA; SÁNCHEZ; SOTOCA, 2006).

Assim que todos os círculos terminam de crescer, aqueles redundantes (círculos totalmente contidos dentro de outros) são removidos. Após essa remoção realiza-se a contagem de círculos necessários para cobrir cada uma das classes. O resultado desta contagem será dividido pelo número de instâncias presentes no conjunto. O resultado dessa divisão é T1.

O tamanho e o número das esferas indicam o quanto as instâncias tendem a ser agrupadas em hiperesferas ou distribuídas em estrutura menores. Conjuntos com pontos que estão muito próximos entre as classes inferem em esferas menores e uma maior quantidade das mesmas para

que toda a classe seja coberta. Dessa forma o valor de T1 é maior indicando que existem mais regiões de sobreposição entre as classes (HO; BASU, 2002).

A Figura 2.7 representa a ideia da adoção de T1, de forma que as circunferências tracejadas são utilizadas como delimitadores das classes. Neste caso, as demais circunferências estariam todas posicionadas dentro daquelas representadas por linhas tracejadas. No exemplo ilustrado o índice corresponderia à relação da quantidade de esferas remanescentes (4) perante o total de instâncias do conjunto (17).

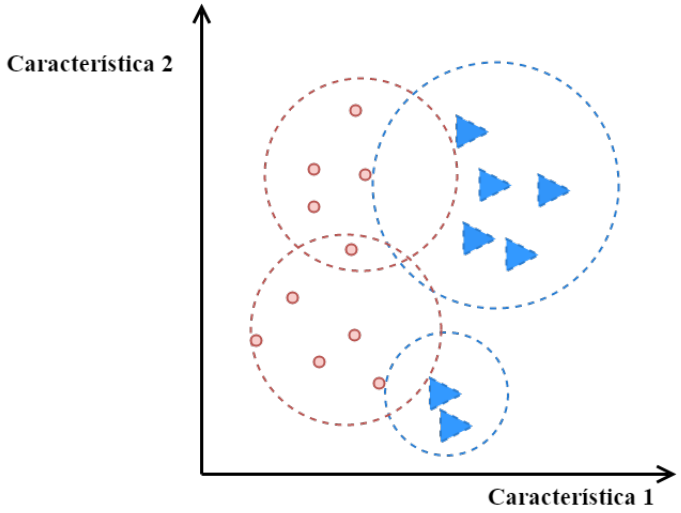

Figura 2.7: Representação do número de esferas necessárias para cobrir duas classes. Adaptado de Brun(2017)

2.4.3.2 Número médio de pontos por dimensão (T2)

Usada para investigar a influência da dimensionalidade de cada base de dados, T2 descreve a densidade da distribuição espacial de amostras através da divisão do número de amostras do conjunto pelo número de características (HO; BASU, 2002), (MOLLINEDA; SÁNCHEZ; SOTOCA, 2006). T2 é apontada pelos autores como uma medida não muito eficiente quanto a separabilidade das classes onde estas possuem base em classificadores lineares, porém, fornece informações relevantes em casos onde os classificadores são não lineares, tal como o KNN.

Landeros (2008) propôs uma variação para T2, em que obtém-se o valor da métrica pela razão entre a raiz n-ésima de i (quantidade de elementos presentes na base de dados) perante o número de atributos.

$$N = \frac{\sqrt[n]{i}}{n} \tag{2.9}$$

2.4.3.3 Não-Linearidade de um Classificador Linear (L3)

Segundo Hoekstra e Duin (1996), Mollineda, Sánchez e Sotoca (2006), L3 é uma medida para a não linearidade de um classificador em relação a um dado conjunto de dados. Com base em um conjunto de treino, primeiramente forma-se um conjunto de teste por meio da interpolação linear entre pares escolhidos randomicamente dentro de uma classe pertencente ao conjunto de treino com atributos também definidos por pesos randômicos. Desse modo, L3 será o valor da taxa de erro dos dados de treino em relação ao conjunto de testes, de forma que seja aplicado um classificador linear, do mesmo modo que é feito em L1.

Para uma melhor compreensão a Figura 2.8 mostra o processo de geração do conjunto de teste. A Figura 2.8(a) representa o conjunto de treino original. Com base no conjunto de treino original são gerados através de sorteios elementos dentro da mesma classe e o peso de cada elemento na formação da nova instância. Então as instâncias selecionadas para a geração do novo padrão são ligadas por linhas de modo que as marcações entre as instâncias são os pesos que cada "pai" tem sobre o novo elemento Figura 2.8(b). A Figura 2.8(c) apresenta o novo conjunto de teste gerado, a partir deste conjunto será calculado o valor do índice.

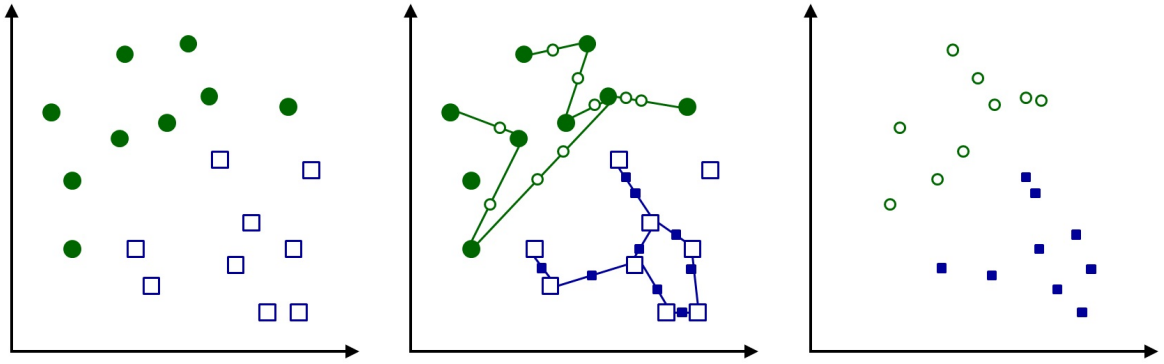

Figura 2.8: Processo de geração do conjunto de teste adotado em L3. Fonte (BRUN, 2017)

2.4.3.4 Não-Linearidade de um Classificador KNN (N4)

A medida N4 segue a mesma ideia da criação do conjunto de testes a qual se baseia L3 (HO; BASU, 2002), (MOLLINEDA; SÁNCHEZ; SOTOCA, 2006). Porém, para o cálculo da taxa

de erro sobre o conjunto de teste, usa-se um classificador de vizinhos mais próximos e não uma abordagem linear.

2.4.3.5 Densidade (D1)

De acordo com Sotoca, Sànchez & Mollineda (2006) a medida de densidade pode ser descrita como o número médio de amostras por unidade de volume onde os pontos são distribuídos. Obtém-se o valor do volume pelo produto da variação total de todas as características de todas as classes.

2.4.3.6 Volume de Vizinhança Local (D2)

O descritor D2 representa o volume médio ocupado pelos K vizinhos mais próximos de cada instância de treinamento (SÁNCHEZ; MOLLINEDA; SOTOCA, 2007).

Considerando $N_k(x_i)$ o conjunto de *k* vizinhos mais próximos de um dado exemplo (x_i), então o volume pode ser definido tal como a Equação 2.10:

$$\nu_i = \prod_{h-1}^{d} (max(f_h, N_k(x_i)) - min(f_h, N_k(x_i))) \tag{2.10}$$

onde $max(f_h, N_k(x_i))$ e $min(f_h, N_k(x_i))$ representam os valores máximos e mínimos da característica f_h entre os *k* vizinhos mais próximos da instância x_i.

A partir disso, o volume de uma vizinhança local pode ser representado como o valor médio de V_i para as *n* instâncias de treino. Tal valor para a vizinhança pode ser expressado pela Equação 2.11

$$D2 = \frac{1}{n} \sum_{i=1}^{n} V_i \tag{2.11}$$

2.4.3.7 Densidade da Classe na Região de Sobreposição (D3)

D3 visa determinar a densidade relativa de cada classe na região de sobreposição das classes. No geral, estas regiões contém os casos mais críticos para a classificação de tarefas e consequentemente origina a maioria dos erros de classificação (SÁNCHEZ; MOLLINEDA; SOTOCA, 2007).

A ideia desta medida é primeiramente encontrar os *k* vizinhos mais próximos de cada exemplo x_i, assim se a maioria dos *k* vizinhos pertencerem a uma classe que não seja a qual ele pertence, diz-se que o elemento faz parte de uma região de sobreposição. Para uma determinada classe ω obtém-se o valor de D3 através da relação do número de elementos na região de justaposição pelo total de instâncias pertencentes à classe.

Quanto menor for o número de exemplos presentes em uma região de sobreposição de determinada classe menor será o valor de D3.

Além da taxonomia das métricas de complexidade em medidas de Separabilidade, Sobreposição e Topologia e Geometria, outra categorização que pode ser feita seria a apresentada na Tabela 2.1. Dessa forma, pode-se perceber quem são os descritores que empregam em seu *core* um classificador linear, quais são os que adotam outros classificadores e aqueles que se baseiam apenas nos valores dos atributos.

Tabela 2.1: Taxonomia das métricas de complexidade

Taxonomia	Métricas
Dependência de um classificador linear	L1, L2, L3
Dependência de um classificador não linear	N3, N4
Dependência dos atributos das instâncias	F1, F2, F3, F4, N1, N2, T1, D1, D2, D3

Sabendo-se que os dados que formam os conjuntos de treino têm influência direta no desempenho do classificador, destaca-se a importância em analisar o comportamento desses dados em termos de complexidade, perante o comportamento dos classificadores. Para tanto, fez-se o levantamento teórico apresentado neste capítulo para embasar o método desenvolvido na pesquisa e que é apresentado no capítulo seguinte.

Capítulo 3

Metodologia

Este trabalho teve como objetivo analisar se há existência de relação entre o tamanho do conjunto de treino com sua assinatura de complexidade. Para tanto, foi necessária a construção de um *framework* que fosse capaz de obter, com base em um conjunto genérico de entrada, suas informações de complexidade. A estrutura implementada é ilustrada na Figura 3.1.

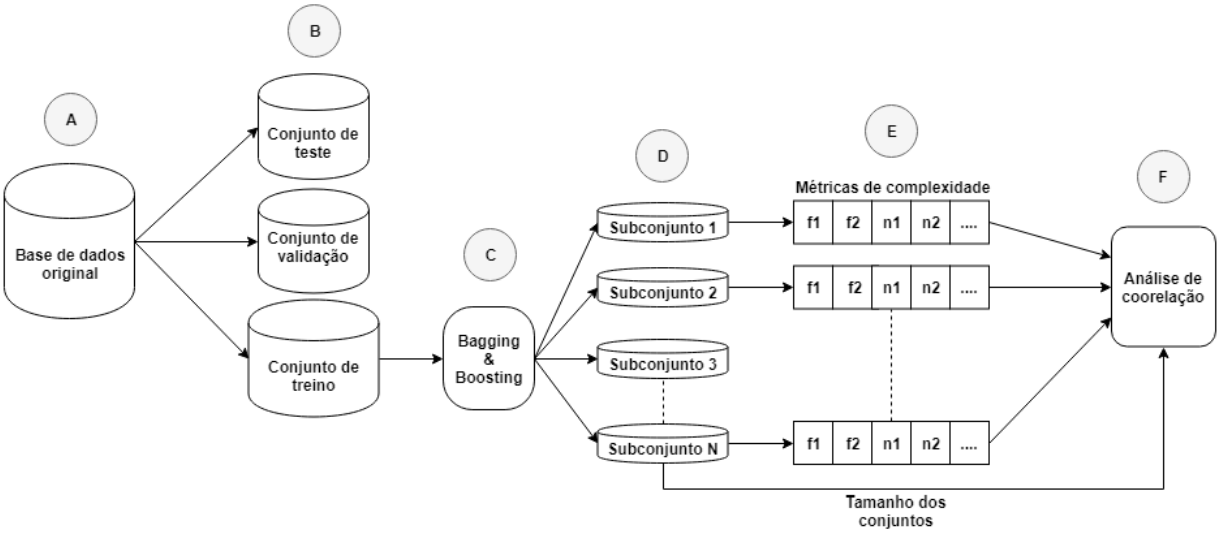

Figura 3.1: Estrutura geral do *framework* construído

Inicialmente, foi necessária a divisão das bases de dados em conjuntos de treino, teste e validação (Figura 3.1-A, B). Após a divisão foram gerados subconjuntos para o treinamento utilizando o *Bagging* (2.3.1.1) e *Boosting* (2.3.1.2) (Figura 3.1 - C, D). Com base nos subconjuntos formados foi possível efetuar a análise de complexidade de cada um (Figura 3.1-E). Estimadas as métricas de complexidade pôde-se efetuar a análise de correlação entre tamanho e assinatura de complexidade de cada subconjunto (Figura 3.1-F). Cada uma destas etapas é

melhor detalhada nas seções seguintes.

Os métodos implementados neste trabalho foram desenvolvidos em linguagem Python, em conjunto com as bibliotecas Pandas e *scikit-learn*. Pandas é uma biblioteca de software livre, que permite a análise de dados e estrutura de dados em alta performance (AUGSPURGER et al., 2018). Em Python, para facilitar o uso de algoritmos de classificação, regressão e agrupamento, utilizou-se a biblioteca *scikit-learn* (PEDREGOSA et al., 2011).

3.1 Base de Dados

A primeira parte do processo consiste em realizar a divisão das bases em três novos subconjuntos: treino, teste e validação. Esta divisão é feita de forma aleatória de maneira que as proporções das classes do conjunto de dados original sejam mantidas. O primeiro conjunto é empregado no aprendizado do classificador. O segundo conjunto, Validação, é usado na definição de parâmetros do classificador. Já o conjunto de teste é utilizado na avaliação de acurácia do classificador.

3.1.1 Geração de Subconjuntos

Com a realização da divisão das bases de dados, é possível obter-se os conjuntos de treinos a partir dos quais serão gerados subconjuntos. Com esse intuito, dois métodos de geração de subconjuntos foram implementados: *Bagging* e *Boosting*.

Um exemplo da geração de subconjuntos é representado na Figura 3.2, onde tem-se inicialmente o conjunto de treino (resultante da divisão das bases de dados a partir da base de dados Iris) com dezoito instâncias, sendo seis da classe Setosa, seis da classe Versicolor e outras seis da classe Virgínica. Após a aplicação do *Bagging* e *Boosting* com proporção de 50%, obtém-se a geração dos subconjuntos SC_1, SC_2 e $SC_3...SC_N$ (em que N corresponde à quantidade total de subconjuntos) mantendo-se a proporção das classes do conjunto original. Assim, SC_1, SC_2 e $SC_3...SC_N$ terão 50% da quantidade de cada classe do conjunto de treino original, neste caso, três instâncias de cada.

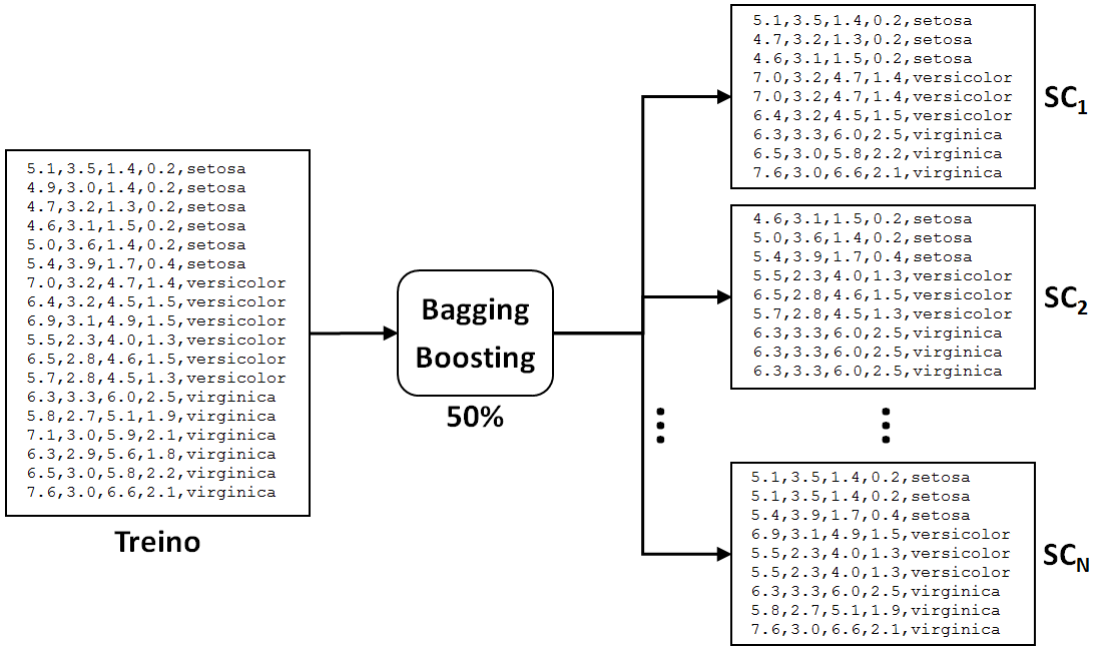

Figura 3.2: Construção de Subconjuntos de Treino

3.2 Estimação das Métricas de Complexidade

Uma vez construídos os subconjuntos fez-se necessária a estimação de sua assinatura de complexidade. Esta parte do processo efetua a estimação de complexidade utilizando a DCoL (ORRIOLS-PUIG; MACIà; HO, 2010). DCoL (*Data Complexity Library*) é uma biblioteca de aprendizado de máquina, implementada em C++ pela *Dell Laboratories*, que implementa o cálculo de quatorze descritores de complexidade, dos quais apenas doze serão empregados neste trabalho. Tais descritores são apresentados na Tabela 3.1.

Tabela 3.1: Métricas presentes na biblioteca DCoL

Sigla	**Métricas de Complexidade**
F1	Relação Máxima do Discriminante de Fischer
F2	Sobreposição de Atributos por Classe
F3	Eficiência Máxima por Atributo Individual
F4	Eficiência Coletiva dos Atributos
L1	Soma Minimizada da Distância de Erro de um Classificador Linear
L2	Taxa de Erro de um Classificador Linear sobre o Treino
L3	Não-Linearidade de um Classificador
N1	Fração de Pontos na Região de Fronteiras
N2	Proporção das Distâncias Intra/Inter Classes até o vizinho mais próximo
N3	Taxa de erro do classificador KNN pela abordagem Leave-One-Out
N4	Não-Linearidade de um Classificador KNN
T1	Fração de Esfera de Cobertura Máxima

Para a utilização da DCoL é necessário que todos os arquivos dos subconjuntos estejam em formato ".arff" (*Attribute-Relation Format File*), que são usados como entrada para a biblioteca. A saída consiste de um vetor com atributos numéricos gravados em arquivo ".txt" para cada uma das entradas.

Além das métricas disponíveis na DCoL são empregados outros dois descritores de implementação própria: D2 seção (2.4.3.6) e D3 seção (2.4.3.7).

3.3 Análise de Correlação

Estimadas as assinaturas de complexidade de cada subconjunto, o passo seguinte consiste em analisar o comportamento de cada métrica de complexidade perante a variação do tamanho do subconjunto, ou seja, o tamanho do subconjunto em relação a cada métrica calculada. Este processo foi realizado para todos os subconjuntos gerados por *Bagging* e *Boosting*. Para a análise da correlação foi utilizado o coeficiente de correlação de Pearson, que é definido na Equação 3.1.

$$P = \frac{n\sum(x_i y_i) - (\sum x_i)(\sum y_i)}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \tag{3.1}$$

Onde x_i corresponde a cada amostra do primeiro conjunto (métrica de complexidade), y_i refere-se a cada amostra do segundo conjunto (tamanho do subconjunto gerado por *Bagging* ou *Boosting*) e n é o número de elementos de cada conjunto.

O coeficiente de correlação de Pearson (r) também conhecido como correlação produto-momento, é utilizado para realizar a medição do grau da correlação linear entre duas variáveis quantitativas. Seu valor está contido entre -1 e 1, onde -1 corresponde a uma relação negativa, enquanto o valor 1 indica uma relação positiva. Se o valor for 0, entende-se que não há relação (FILHO; SILVA JR., 2010).

Dentro do limite da correlação pode obter-se alguns níveis intermediários (VIEIRA, 2015), conforme apresentado na Tabela 3.2.

Tabela 3.2: Faixas de interpretação da correlação de Pearson

Valor de ρ (+ ou -)	Interpretação
0.00 a 0.25	Uma correlação pequena ou nula
0.25 a 0.50	Uma correlação fraca
0.50 a 0.75	Uma correlação moderada
0.75 a 1.00	Uma correlação forte ou perfeita

Capítulo 4

Resultados Experimentais

Enquanto o capítulo anterior teve como objetivo explanar, de uma forma genérica, quais são os métodos que foram utilizados na pesquisa, nesta seção são detalhadas quais os parâmetros aplicados ao experimento, bem como a forma com que ocorreu a validação do protocolo e análise dos resultados alcançados.

4.1 Bases de dados

Visando obter resultados mais consistentes na avaliação entre o tamanho do conjunto de treino com a sua assinatura de complexidade, optou-se em utilizar um conjunto composto de vinte e seis bases bases de dados, as quais são apresentadas na Tabela 4.1. Quatorze são originárias do repositório da UCI (BACHE; LICHMAN, 2013), duas são procedentes do repositório KEEL (*Knowledge Extraction based on Evolutionary Learning*) (ALCALá-FDEZ et al., 2011), outras quatro pertencentes à LKC (*Ludmila Kuncheva Collection of Real Medical Data*) (KUNCHEVA, 2004), quatro provenientes do projeto STATLOG (KING; FENG; SUTHERLAND, 1995) e duas bases geradas artificialmente com o *toolbox PRTools* do Matlab.

A primeira parte do processo consiste em realizar a divisão das bases. Para a segmentação, cada uma das vinte e seis bases foi dividida aleatoriamente em três conjuntos: treino, teste e validação. Neste processo, o conjunto de treino ficou com 50% do tamanho total da base, o de teste 25% e o conjunto de validação 25%. A divisão foi feita mantendo a proporção das classes do conjunto original. O primeiro conjunto é empregado no aprendizado do classificador. O segundo conjunto, Validação, é usado na definição de parâmetros do classificador. Já o conjunto de teste é utilizado na avaliação de acurácia do classificador.

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
Faults	1941	971	485	485	27	7	UCI
German	1000	500	250	250	24	2	STATLOG
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
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

O número total de instâncias de cada subconjunto é apresentado nas colunas “Treino”, “Teste” e “Validação” da Tabela 4.1. Além disso, são especificados também o número de classes e atributos de cada uma das bases.

O processo de divisão dos dados de entrada, geração dos subconjuntos e demais etapas da validação do protocolo foram submetidos a 20 repetições, de forma a construir uma avaliação robusta dos métodos.

4.2 Geração de Subconjuntos

Com base nos conjuntos de treinos formados foram gerados subconjuntos de tamanhos variados, proporcionais ao conjunto base. Estes subconjuntos, que foram formados através dos algoritmos de *Bagging* e *Boosting*, possuem 10%, 33%, 50% e 66% do tamanho do conjunto de treino. Para cada proporção são gerados cem subconjuntos para cada base de dados, totalizando assim, dez mil e quatrocentos subconjuntos para cada um dos métodos de geração em cada uma

das repetições.

Para a geração baseada em *Bagging* não foi necessária a especificação de parâmetros, uma vez que o método consiste basicamente em sortear as instâncias com reposição.

No caso do *Boosting*, durante a fase de geração, é preciso efetuar a classificação das instâncias que foram geradas em etapas anteriores. Inicialmente, todas as instâncias do conjunto possuem o mesmo peso. No entanto, a partir da segunda iteração, as instâncias tem seu peso aumentado com base nos erros de um classificador. Para tal processo utilizou-se o classificador KNN (K-Vizinhos Mais Próximos) com valor de K igual a cinco.

4.3 Estimação de Complexidade

Para a estimação de complexidade, além das 12 métricas disponibilizadas na DCoL, duas foram implementadas: D2 e D3. Para a implementação das métricas D2 e D3 foi necessário a utilização do KNN, visto que tal métrica verifica se um elemento é de uma região de fronteira, de acordo com as instâncias mais próximas. Para este processo, adotou-se um valor para K igual a sete.

Dentre as 12 métricas que estão presentes na DCoL cinco são parametrizáveis, sendo elas: L1, L2, L3, N3 e N4. Para as métricas L1, L2 e L3 é utilizado o classificador SVM com Kernel Linear. Nos dois descritores restantes, utiliza-se o KNN com K igual a um.

4.4 Análise da Correlação

Cada subconjunto gerado resultou em um vetor de descritores de complexidade. Assim, para cada uma das métricas levantadas presentes no vetor, analisou-se seu comportamento perante as dimensões dos subconjuntos gerados pelos métodos de *Bagging* e *Boosting*.

Visando ilustrar o processo de correlação o gráfico da Figura 4.1 apresenta um exemplo do cálculo efetuado para uma métrica de complexidade genérica. Na imagem, no eixo das abscissas são representadas as proporções, em termos de tamanho (10%, 33%, 50%, 66%), que cada subconjunto pode possuir. No eixo das ordenadas são representados os valores da métrica de complexidade referente à cada subconjunto.

O cálculo do coeficiente de correlação foi realizado para cada uma das bases, seguindo a ideia representada na Figura 4.1. Assim a correlação é estimada para um conjunto composto de

quatrocentos elementos (cem para cada uma das proporções).

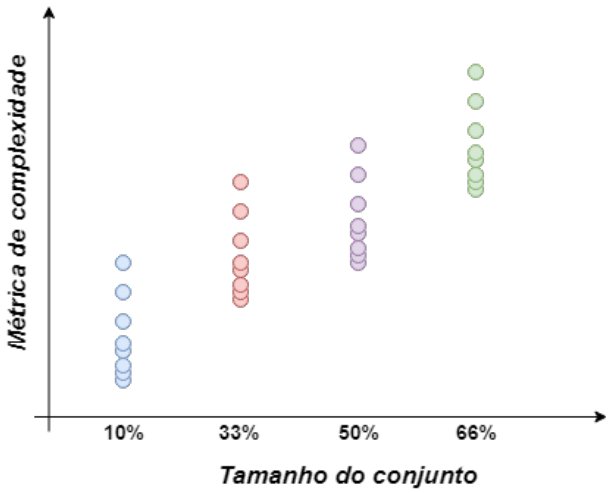

Figura 4.1: Gráfico representativo da análise de correlação entre métricas e proporção do subconjunto

Visando representar de forma gráfica as análises realizadas, foram plotados gráficos de dispersão que exemplificam o comportamento dos dados em relação ao tamanho dos conjuntos.

Na Figura 4.2 é representada uma correlação negativa observada ao longo dos experimentos. O gráfico foi gerado a partir da base Faults com relação à métrica F4, na primeira das vinte repetições executadas. O valor da correlação observado neste caso foi de -0.9681, indicando uma forte correlação negativa (representada pela reta vermelha pontilhada). Analisando os valores é possível notar que, quanto maior o tamanho do conjunto, menor é o valor de F4, logo, menos discriminantes são os dados.

Para melhor visualizar a variação do descritor F4 perante os quatrocentos subconjuntos gerados, a Figura 4.3 detalha como estão distribuídos os valores do descritor para cada um dos conjuntos. Os primeiros cem elementos (representados em azul) correspondem aos subconjuntos com proporção de 10%, os elementos seguintes (em amarelo) referem-se aos subconjuntos de tamanho 33%. Na cor verde são apresentados os valores da métrica obtidos a partir dos subconjuntos com 50% do tamanho do treino e, por fim, em laranja são ilustrados os elementos com 66% de proporção.

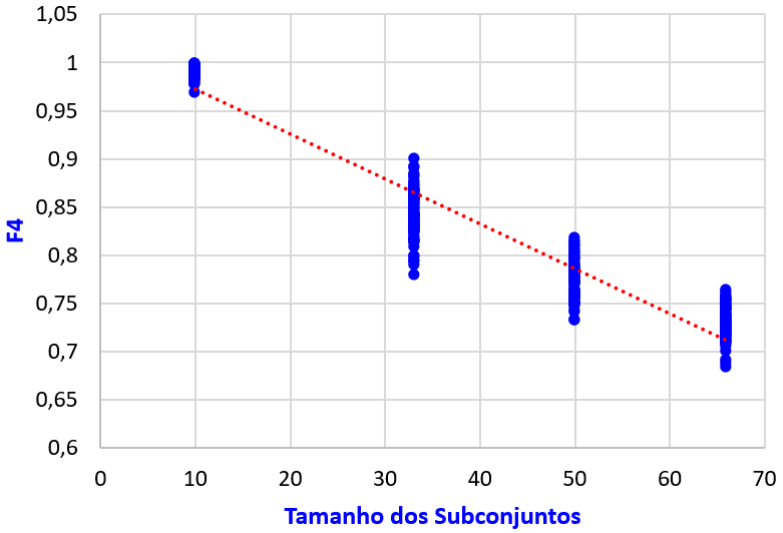

Figura 4.2: Correlação negativa para a métrica F4 em relação a variação dos subconjuntos

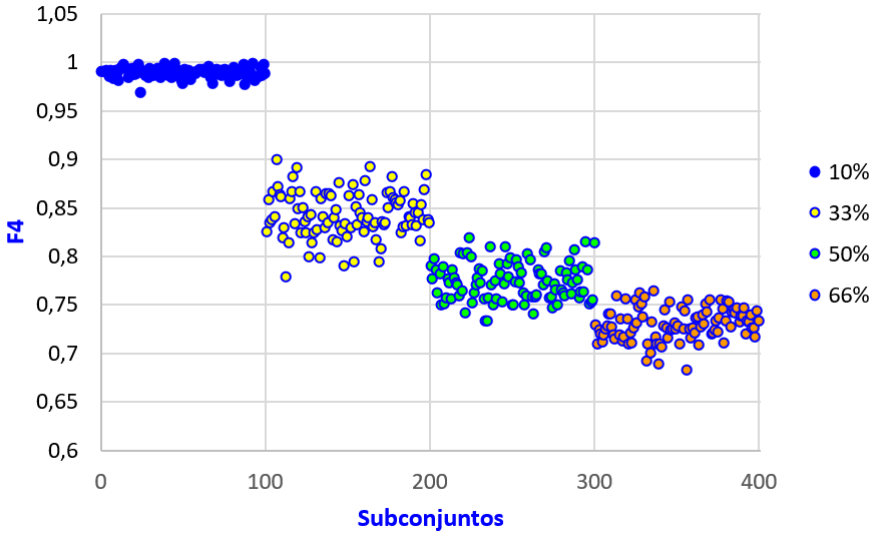

Figura 4.3: Dispersão dos dados entre a métrica F4 e a variação dos subconjuntos

Na Figura 4.4 é representada uma correlação positiva obtida durante as execuções dos experimentos. O gráfico ilustra um exemplo da base Faults com relação à métrica L1, na primeira das vinte repetições executadas. O valor observado para a correlação neste caso foi de 0.9215, caracterizando uma forte correlação positiva (representada pela reta vermelha pontilhada). Analisando os valores é possível notar que, quanto maior o tamanho do conjunto, maior é o valor de L1, o que indica que o classificador linear está cometendo mais erros. Além de aumentar o

percentual de erros este fato implica em um valor maior para a soma das instâncias classificadas incorretamente até a fronteira de separação.

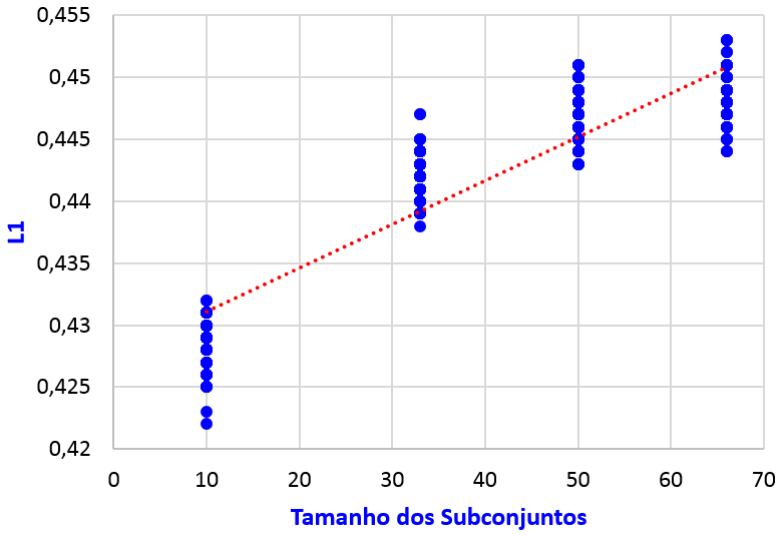

Figura 4.4: Correlação positiva de L1 em relação a variação dos subconjuntos

A Figura 4.5 apresenta uma exposição mais detalhada dos valores de L1 ilustrados na Figura 4.4. Nesta representação são exibidos os valores de complexidade para cada um dos subconjuntos. A representação visual segue o mesmo padrão de cores da Figura 4.3. Nota-se o aumento nos valores de L1 conforme aumenta-se o tamanho dos subconjuntos.

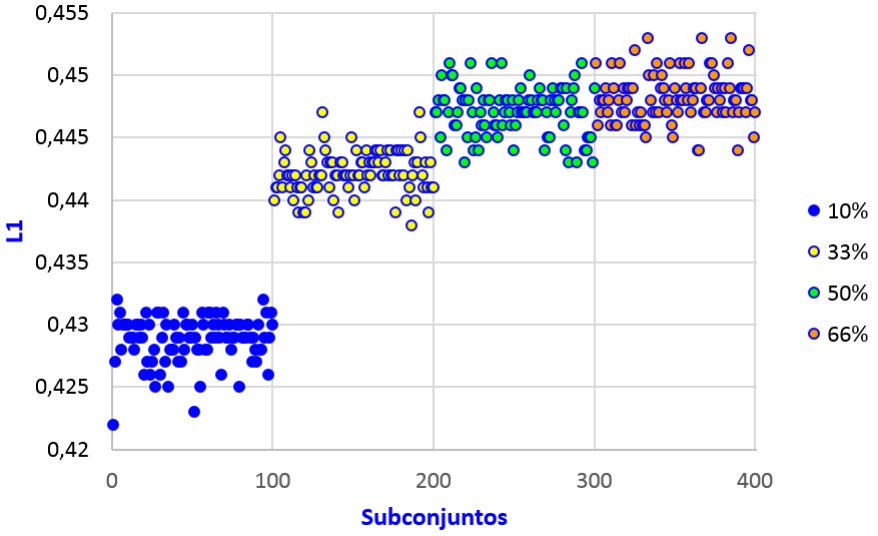

Figura 4.5: Dispersão dos dados obtidos entre a métrica L1 e a variação dos subconjuntos

Para exemplificar uma correlação nula, utilizou-se à repetição (5) da base de dados Heart

perante o comportamento da métrica T1. Neste cenário, o coeficiente de correlação apresentou valor de 0,0035. Ilustrando tal comportamento, na Figura 4.6 é representada a relação observada entre a variação no tamanho dos subconjuntos e o valor da métrica. Notamos que a variação do tamanho do conjunto não possui influência no resultado da métrica, ou seja, o número de esferas usadas na cobertura das classes não é influenciado por variações no tamanho dos subconjuntos.

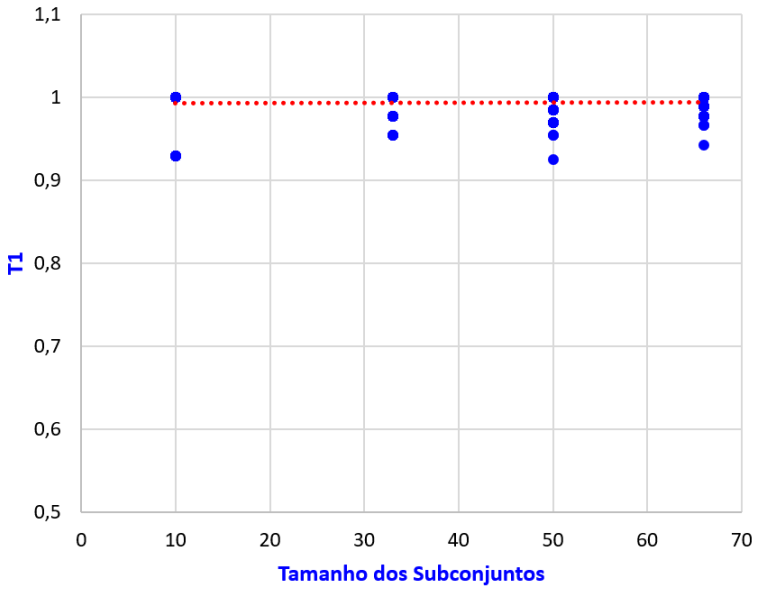

Figura 4.6: Correlação nula da métrica T1 em relação a variação dos subconjuntos

A Figura 4.7 apresenta uma distribuição mais detalhada dos valores de T1 ilustrados na figura anterior. Nesta representação são exibidos os valores de complexidade para cada um dos quatrocentos subconjuntos. A interpretação visual segue o mesmo padrão de cores da Figura 4.3. Observando-se a variação dos valores de T1 no gráfico é possível perceber que a métrica apresentou variação bastante pequena, onde os índices do descritor sempre estiveram próximos de 1.

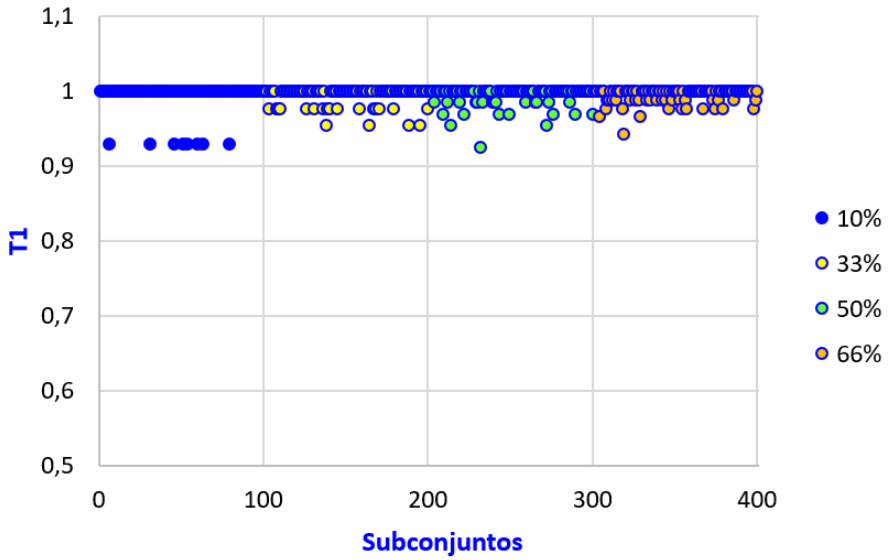

Figura 4.7: Dispersão dos dados entre a métrica T1 e a variação dos subconjuntos

Para cada repetição, obteve-se um valor de correlação entre cada métrica de complexidade e cada uma das bases de teste. Assim, o coeficiente final foi obtido pela média de todas as 20 repetições. A partir destas médias foram geradas duas tabelas, uma para o *Bagging* (Tabela 4.2) e outra para o *Boosting* (Tabela 4.3). Nas tabelas, cada linha corresponde a uma base de dados, enquanto as colunas remetem às métricas de complexidade.

Buscando um melhor entendimento das correlações obtidas entre cada uma das métricas e o tamanho de cada subconjunto, realizou-se uma análise individual para cada um dos descritores de complexidade em relação às médias apresentadas nas Tabelas 4.2 e 4.3. Tal análise teve por objetivo identificar métricas suscetíveis à variação no tamanho de um subconjunto.

Tabela 4.2: Coeficiente de Correlação médio entre os descritores de complexidade e o tamanho dos subconjuntos gerados por ***bagging*** ao longo das 20 repetições para cada uma das 26 bases

Base	F1	F2	F3	F4	L1	L2	L3	N1	N2	N3	N4	T1	D2	D3
Adult	-0,1666	0,1778	-0,7217	-0,8567	-0,7158	-0,3463	-0,4155	-0,6035	-0,9038	-0,6033	0,5101	0,0999	-0,0003	-0,1806
Banana	-0,0959	0,5236	-0,5264	-0,7009	-0,8450	-0,5647	-0,5281	-0,7825	-0,9413	-0,6209	0,2190	-0,1636	-0,8921	-0,2971
Blood	-0,3054	0,2172	-0,7054	-0,7961	-0,0804	-0,0987	0,0000	-0,6621	-0,8348	-0,7072	0,4065	0,2283	-0,3657	-0,0371
CTG	-0,2680	0,1673	-0,6598	-0,9063	0,5016	0,0917	0,0000	-0,8329	-0,9587	-0,8164	0,2605	0,3105	-0,0003	-0,7028
Diabetes	-0,2881	0,5470	-0,7394	-0,8976	0,0799	-0,1049	-0,0140	-0,6807	-0,9108	-0,7183	0,6062	0,1267	-0,5359	-0,4024
Faults	-0,4295	0,3462	-0,8442	-0,9634	0,8788	-0,0728	-0,1839	-0,9080	-0,9532	-0,8947	0,7990	-0,0784	-0,1348	-0,8620
German	-0,5372	0,4220	-0,7345	-0,8579	0,7750	-0,3964	-0,3197	-0,7331	-0,9446	-0,7838	0,7753	0,0207	0,0000	-0,3006
Haberman	-0,4225	0,6915	-0,7662	-0,8638	-0,4942	-0,5554	0,0000	-0,5400	-0,7788	-0,5807	0,5496	0,1915	-0,6790	0,2597
Heart	-0,1524	0,6060	-0,7769	-0,3171	-0,7461	-0,7118	-0,7347	-0,5325	-0,8380	-0,4800	0,5117	0,0963	-0,4222	-0,5224
ILPD	-0,5320	-0,0856	-0,7454	-0,8970	-0,2104	-0,3017	0,0000	-0,6365	-0,8906	-0,6893	0,5983	0,1590	-0,0063	-0,1006
Ionosphere	-0,4001	0,0000	-0,7784	0,4359	-0,6867	-0,7401	-0,7299	-0,7231	-0,8430	-0,6749	0,2512	0,2480	0,0000	-0,7163
Laryngeal1	-0,3957	0,0000	-0,5351	0,6226	-0,2212	-0,3009	-0,2114	-0,5413	-0,8143	-0,4829	0,2528	0,1591	-0,0974	-0,7298
Laryngeal3	-0,5589	0,2689	-0,8478	0,4292	-0,1127	-0,0950	0,0443	-0,6872	-0,8413	-0,6578	0,5161	0,3040	-0,0926	-0,4508
Lithuanian	-0,0998	0,5119	-0,5813	-0,7450	-0,9146	-0,6667	-0,6845	-0,8137	-0,9452	-0,7027	0,0111	-0,0321	-0,8963	-0,3104
Liver	-0,5571	0,4012	-0,7886	-0,8905	0,1128	0,0401	0,0484	-0,6031	-0,8501	-0,6612	0,5845	0,1000	-0,1313	-0,5337
Mammo	-0,3922	0,3452	-0,6544	-0,7519	-0,6031	-0,3418	-0,3982	-0,4430	-0,8027	-0,5643	0,4138	-0,0353	-0,2779	-0,1427
Monk	-0,3170	0,6805	-0,4383	0,5237	-0,8622	-0,7107	-0,7158	-0,7373	-0,8861	-0,7162	0,2643	-0,1131	-0,8173	-0,5625
Phoneme	-0,3578	0,4672	-0,5051	-0,7851	0,9211	-0,3071	-0,2769	-0,8946	-0,9652	-0,9109	0,4092	0,1475	-0,8343	-0,8036
Segmentation	-0,2751	0,0000	-0,7827	-0,5002	0,8714	-0,0157	0,0036	-0,9293	-0,8766	-0,9002	-0,3781	-0,2567	0,0000	-0,8864
Sonar	-0,5421	0,0000	-0,8539	0,5676	0,0271	-0,0811	-0,1683	-0,6893	-0,8622	-0,6853	0,4700	0,0000	0,0000	-0,6064
Thyroid	-0,3893	0,0000	-0,6264	0,5775	0,7666	-0,1015	0,0000	-0,5391	-0,8914	-0,4540	0,2931	0,2809	-0,1109	-0,5822
Vehicle	-0,5696	-0,0356	-0,8330	-0,8592	-0,3024	-0,3859	0,0000	-0,7881	-0,9215	-0,8150	0,5926	0,3799	-0,0181	-0,7963
Vertebral	-0,3910	0,3594	-0,5938	0,1212	0,2200	0,0325	0,0106	-0,5812	-0,8147	-0,5483	0,3409	0,1796	-0,4436	-0,4909
WBC	-0,3880	0,0000	-0,5927	0,5466	0,1771	-0,5736	-0,5798	-0,6285	-0,8886	-0,5151	0,0483	0,2334	-0,0847	-0,3858
WDVG	-0,4866	0,7295	-0,8382	-0,9030	0,3449	0,1062	0,0000	-0,8761	-0,9855	-0,9029	0,5969	0,0857	-0,7926	-0,5030
Weaning	-0,3787	0,2091	-0,7109	0,4298	-0,5844	-0,1646	-0,0851	-0,6763	-0,8703	-0,6448	0,5149	0,0214	-0,4439	-0,6665

Tabela 4.3: Coeficiente de Correlação médio entre os descritores de complexidade e o tamanho dos subconjuntos gerados por ***boosting*** ao longo das 20 repetições para cada uma das 26 bases

	F1	F2	F3	F4	L1	L2	L3	N1	N2	N3	N4	T1	D2	D3
Adult	-0,2345	0,2596	-0,7433	-0,8894	-0,3567	-0,2983	-0,3855	-0,7472	-0,9170	-0,8140	0,5747	0,0408	-0,0518	-0,0638
Banana	-0,1369	0,2859	-0,3521	-0,5094	-0,7842	-0,5681	-0,5851	-0,8267	-0,9334	-0,7959	0,0824	-0,1787	-0,8880	-0,2758
Blood	-0,4002	0,2284	-0,6323	-0,7604	-0,0499	0,0276	0,1060	-0,8640	-0,8457	-0,8336	0,5328	-0,2587	-0,4212	0,1061
CTG	0,0254	0,2127	-0,5690	-0,8660	0,1646	0,0486	-0,1389	-0,8604	-0,8602	-0,8799	0,3358	0,4052	-0,0256	-0,1402
Diabetes	-0,5073	0,5670	-0,7421	-0,8881	-0,0133	0,1337	0,2029	-0,7875	-0,9233	-0,8213	0,7012	0,0989	-0,5592	0,0163
Faults	-0,5168	0,3790	-0,7966	-0,9296	0,3466	0,3117	-0,0137	-0,8161	-0,9528	-0,9068	0,7501	0,2408	-0,1313	-0,5549
German	-0,4278	0,4932	-0,7398	-0,8155	0,0855	-0,3508	-0,4012	-0,7806	-0,9401	-0,8531	0,8121	0,0109	0,0000	0,0401
Haberman	-0,4849	0,6871	-0,7352	-0,8562	0,0666	0,3131	0,2895	-0,7102	-0,8209	-0,7143	0,6133	0,1379	-0,6969	-0,0756
Heart	-0,4276	0,6380	-0,7903	-0,5819	-0,2709	-0,0646	-0,0780	-0,6265	-0,8661	-0,6761	0,6630	0,0560	-0,3872	-0,1986
ILPD	-0,5900	0,0264	-0,7413	-0,9140	0,1253	0,0624	0,0132	-0,7397	-0,8986	-0,7893	0,7009	0,1371	-0,0175	0,0666
Ionosphere	-0,5793	0,0000	-0,6814	0,3996	-0,3315	-0,3519	-0,0892	-0,8066	-0,8710	-0,8046	0,3864	0,1883	0,0000	-0,4344
Laryngeal1	-0,4471	-0,0005	-0,7249	0,4358	0,3318	0,2521	0,0600	-0,6214	-0,8326	-0,6814	0,4439	0,1079	-0,0575	-0,4517
Laryngeal3	-0,5655	0,2965	-0,8401	0,4227	0,1718	0,2282	0,1569	-0,7082	-0,8533	-0,7301	0,6820	0,2461	-0,1033	-0,3247
Lithuanian	-0,1464	0,1593	-0,4579	-0,6085	-0,7948	-0,5352	-0,5476	-0,8381	-0,9329	-0,8256	0,0605	0,0448	-0,8815	-0,1735
Liver	-0,5607	0,4585	-0,7889	-0,8969	0,1874	0,2311	0,1824	-0,6792	-0,8603	-0,7217	0,6101	0,1278	-0,2242	-0,1783
Mammo	-0,4123	0,2898	-0,6007	-0,6962	-0,7424	-0,5044	0,2451	-0,8222	-0,8759	-0,8164	0,5355	-0,0233	-0,2715	-0,4376
Monk	-0,4164	0,6826	-0,1174	0,2509	-0,6093	-0,0990	-0,0363	-0,7425	-0,8858	-0,7698	0,4160	-0,1393	-0,7958	-0,3059
Phoneme	-0,3405	0,3718	-0,4561	-0,6955	0,2872	0,2573	-0,1855	-0,8906	-0,9589	-0,9384	0,4245	0,1905	-0,8364	-0,1134
Segmentation	-0,3452	0,0000	-0,4459	-0,7134	0,2256	0,1154	-0,0356	-0,9098	-0,8580	-0,8972	-0,0036	-0,0602	0,0000	-0,8481
Sonar	-0,5369	0,0000	-0,8533	0,5373	0,1193	0,1987	0,2083	-0,6805	-0,8668	-0,7149	0,4661	0,0234	0,0000	-0,4352
Thyroid	-0,2775	0,0000	-0,4926	0,3887	0,3158	0,1412	0,0000	-0,7059	-0,9022	-0,7301	0,3275	0,1540	-0,0769	-0,2718
Vehicle	-0,6382	0,0227	-0,8551	-0,9022	0,5691	0,5681	0,0000	-0,8106	-0,9257	-0,8458	0,6951	0,3848	0,0212	-0,6232
Vertebral	-0,4580	0,4307	-0,6854	-0,3815	0,2416	0,2123	0,0422	-0,6828	-0,8542	-0,7057	0,4819	0,1404	-0,4648	-0,1674
WBC	-0,4126	0,0000	-0,6306	0,4629	0,0433	-0,3187	-0,3727	-0,7074	-0,8631	-0,6515	0,1672	0,1507	-0,0764	-0,1888
WDVG	-0,4107	0,6256	-0,7825	-0,8808	0,6545	-0,2341	0,0000	-0,7420	-0,9377	-0,9229	0,4654	0,0689	-0,7579	0,2148
Weaning	-0,4350	0,1988	-0,7704	0,4127	0,1238	0,1354	0,0374	-0,7044	-0,8755	-0,7291	0,6132	0,0006	-0,4076	-0,3427

4.4.1 F1

A partir dos dados apresentados na Tabela 4.2, é perceptível o fato de que, para todas as bases, o valor final da média da correlação para F1 é sempre negativo, ou seja, utilizando o algoritmo de *Bagging*, o crescimento dos subconjuntos é inversamente proporcional ao valor de F1, de modo que, quanto maior o tamanho do subconjunto, menor é o valor da métrica e, possivelmente, menos separáveis serão as classes do subconjunto.

O algoritmo de *Boosting* apresenta comportamento similar. Todas as bases compreendem valores negativos para a correlação, implicando no mesmo resultado apresentado para o algoritmo de *Bagging*.

4.4.2 F2

Apesar da variação do tamanho dos subconjuntos, a métrica F2 teve uma correlação positiva tanto para o *Bagging* quanto para o *Bossting* em grande parte das bases, com algumas exceções (ILPD, Ionosphere, Laryngeal1, Segmentation, Sonar, Thyroid, Vehicle e WBC), onde os valores da correlação foram zero ou próximos a zero, ou seja, uma correlação pequena ou nula.

Para as outras bases a correlação foi positiva com média 0,281, caracterizando assim, uma correlação fraca. Como para F2, quanto maior seu valor maior sobreposição, infere-se que quanto maior o subconjunto mais alto é o valor da métrica.

4.4.3 F3

Analisando os valores apresentados pelas médias do *Bagging* e *Boosting* para o descritor F3, percebe-se que o valor da correlação obtida é negativa para todas as bases. Para esta métrica, a média da correlação foi de -0,658, que indica uma correlação moderada.

Como para F3 valores próximos a 0 remetem a um problema de alta complexidade, entende-se que, a medida que o tamanho do subconjunto aumenta mais complexo se torna o problema, já que o valor de F3 diminui.

4.4.4 F4

Uma vez que F4 reflete o percentual de instâncias que podem ser separadas pelo conjunto de atributos, espera-se que o aumento do número de instâncias implique em um problema mais

complexo e, consequentemente, este percentual decaia.

O valor médio observado da correlação indica que conforme o tamanho dos subconjuntos aumenta o valor de F4 diminui. No entanto, para algumas bases observou-se correlação moderada positiva, o que sugere um aumento no valor de F4 em relação ao crescimento dos subconjuntos. Analisando-se as bases em que o fato ocorreu notou-se que praticamente todas apresentam apenas duas classes e, geralmente, possuem os maiores números de atributos.

4.4.5 L1

O comportamento de L1 mostrou-se bastante variado em relação às dimensões dos subconjuntos. Em parte das bases, o aumento da dimensão dos subconjuntos implicou em diminuição do valor de complexidade. Este fato foi observado para as bases compostas por duas classes e, em sua maioria por poucos atributos. Por outro lado em bases onde o número de atributos era maior e o número de classes era superior a dois a correlação foi positiva, ou seja, a soma da distância dos erros aumentou. Acredita-se que tal correlação tenha se mostrado positiva pois um número maior de atributos e classes impactam em valores maiores para a distância dos erros até a fronteira de classificação.

4.4.6 L2

Para os subconjuntos gerados por *bagging* observou-se que a taxa de erro do classificador linear apresentou certa diminuição conforme o tamanho dos subconjuntos aumentou. Esta correlação negativa no entanto, mostrou-se fraca, com valores inferiores a 0,3.

A relação dos subconjuntos gerados por *boosting* perante L2, no entanto, mostrou-se muito próxima de zero, indicando que, neste caso, a variação do tamanho dos subconjuntos não causou variação na taxa de acerto do classificador linear.

4.4.7 L3

O comportamento apresentado pelo descritor L3 foi bastante similar ao observado para L2. Tal fato já era esperado uma vez que ambos baseiam-se na taxa de erro de um classificador linear.

4.4.8 N1

O aumento no tamanho dos subconjuntos teve influência direta no comportamento da métrica N1. Os experimentos mostraram que, conforme o tamanho do subconjunto aumenta, o nível da complexidade estimado para a métrica de complexidade diminui.

Como N1 reflete o número de vizinhos na região de sobreposição, acredita-se que conforme o número de instâncias foi aumentando, a maioria delas foi sendo distribuída nas regiões onde não há tanta sobreposição e, por isso, o valor da complexidade foi diminuindo, já que a proporção de instâncias na fronteira foi se tornando menor.

4.4.9 N2

Assim como ocorreu para a métrica anterior, N2 apresentou uma alta correlação negativa perante o crescimento dos subconjuntos. O valor da correlação apresentou média aproximada a -0,89 para os grupos gerados por *Bagging* e por *Boosting*.

Tal comportamento indica que, conforme a dimensão dos subconjuntos aumentou, a distância entre os elementos de uma mesma classe tornaram-se menores, aumentando a coesão de cada classe. Essa diminuição implica em valores menores para F2 que mede a relação intra classes perante a inter classes.

4.4.10 N3

Considerando que N3 é a taxa de erro de um classificador KNN, era esperado que o valor da métrica diminuísse conforme o tamanho do conjunto de treino (subconjuntos) aumentasse. Logo, os resultados obtidos foram os esperados. N3 apresentou média de 0,791 em seu coeficiente de correlação tanto para o *Bagging* quanto ao *Boosting*. Tal resultado representa uma correlação forte ou perfeita.

4.4.11 N4

Analisando-se o comportamento de N4 perante a variação do tamanho dos subconjuntos observou-se uma correlação positiva fraca (0,43). O que sugere que o classificador 1NN comete mais erros quando os subconjuntos possuem mais instâncias. Acredita-se que este fato ocorre

pois conforme os subconjuntos vão sendo aumentados, a dificuldade do classificador também aumenta.

4.4.12 T1

O valor da correlação de T1 perante a variação do tamanho dos subconjuntos foi considerado pequeno ou nulo (aproximadamente 0,156), o que indica que a métrica é pouco suscetível à oscilação na dimensão dos subconjuntos formados, tanto por *bagging* quanto por *boosting*.

4.4.13 D2

De acordo com a variação do tamanho dos subconjuntos o valor de D2 manteve-se negativo ou nulo. O valor médio da correlação de D2 calculado foi de -0,311 para o *Bagging* e -0,314 para o *Boosting*. Ambos os valores mostram que a métrica tem um valor mais próximo a zero à medida que os subconjuntos crescem, resultando em uma correlação fraca.

4.4.14 D3

Para D3, quanto maior o número de elementos que pertencem a uma região de sobreposição, maior será seu valor. Levando em conta esta afirmação e analisando os valores coletados, nota-se uma correlação negativa fraca na utilização do algoritmo de *boosting*, pois o valor médio levantado para D3 foi de -0,2711 o que indica que a maior parte das instâncias dos conjuntos maiores são distribuídas em regiões de menor sobreposição.

Já para o *bagging* houve uma correlação pequena, devido ao fato de que o valor de D3 foi de -0,1558.

4.4.15 *Bagging* vs *Boosting*

Com o objetivo de comparar o comportamento das métricas de complexidade perante às variações do tamanho dos subconjuntos gerados por *bagging* e *boosting*, levantou-se o valor médio absoluto das correlações para as duas técnicas de geração, os quais são apresentados nas Tabela 4.4. O objetivo foi tentar mensurar o quanto o tamanho dos subconjuntos interfere nas medidas de complexidade, independente de ser de forma positiva ou negativa.

Tabela 4.4: Média dos valores absolutos para a correlação entre as medidas de complexidade e o tamanho dos subconjuntos

Métricas	Bagging	Boosting
F1	0,373	0,413
F2	0,300	0,281
F3	0,699	0,655
F4	0,683	0,658
L1	0,502	0,308
L2	0,304	0,252
L3	0,237	0,170
N1	0,695	0,762
N2	0,885	0,889
N3	0,682	0,791
N4	0,430	0,483
T1	0,156	0,139
D2	0,311	0,314
D3	0,494	0,271

Analisando-se os valores absolutos dos coeficientes de cada uma das métricas, nota-se que, em sua maioria, os algoritmos de geração não apresentaram discrepâncias consideráveis nas correlações apresentadas.

Por outro lado, observou-se que em quatro casos específicos (L1, N1, N3 e D3), a utilização do *bagging* e *boosting* teve influência direta nos resultados, ou seja, notou-se certa diferença entre as correlações.

Para L1 a diferença entre as abordagens foi de aproximadamente 20 pontos percentuais. Segundo o valor obtido pelo *bagging* a métrica se enquadra na faixa de correlação moderada, já pelo *boosting*, o índice está na faixa de correlação fraca.

Com relação ao descritor N1, tem-se uma diferença menor, em torno de 10 pontos percentuais. No entanto, apesar da diferença ser menor, a correlação de cada estratégia de geração cai em uma faixa diferente de interpretação: para o *bagging* a relação entre tamanho e complexidade é apenas moderada, enquanto para o *boosting*, ela é considerada forte ou perfeita.

Como ocorreu para N1, o descritor N3, apresentou uma correlação moderada no uso do *bagging* e uma correlação forte ou perfeita no uso do *boosting*. Contudo, para D3, tanto para o *bagging* quanto para o *boosting* a correlação se mantém na faixa moderada, mas com uma diferença próxima aos 20 pontos percentuais entre as duas estratégias de geração.

Um dos objetivos do trabalho foi tentar identificar quais métricas de complexidade são mais

suscetíveis à variação do tamanho dos conjuntos de dados. De forma a tentar responder esta pergunta, são apresentados na Tabela 4.5 o conjunto de descritores de complexidade analisados, divididos por faixa de correlação para os subconjuntos gerados pelo *Bagging* e *Boosting*.

Tabela 4.5: Comportamento das métricas de complexidade perante às faixas de classificação do coeficiente de correlação de Pearson

Faixa da Correlação	Bagging	Boosting
Pequena ou Nula	L3 e T1	L3 e T1
Fraca	F1, F2, L2, N4, D2 e D3	F1, F2, L1, L2, N4, D2 e D3
Moderada	F3, F4, L1, N1 e N3	F3 e F4
Forte ou Perfeita	N2	N1, N2 e N3

Analisando-se os dados apresentados na tabela percebe-se que as medidas L3, T1, F1, F2, L2, N4, D2 e D3 são menos influenciadas pelo tamanho do conjunto em que é calculada. Entretanto, os descritores F3, F4, N1, N2 e N3 sofrem mais influência da quantidade de instâncias presentes no conjunto. A medida L1 apresentou um comportamento mais indefinido, variando entre correlação fraca e moderada.

Com o objetivo de analisar se os métodos de geração apresentam comportamento similar em termos de sentido da correlação entre a variação no tamanho dos subconjuntos perante os valores das métricas de complexidade são apresentados na Tabela 4.6 os valores médios das correlações obtidas.

Analisando-se os valores apresentados, observou-se que, tanto para o *Bagging* quanto para o *Boosting*, a correlação apresenta o mesmo sentido, ou seja, quando um índice de complexidade aumenta para o *Bagging* o mesmo aumenta para o *Boosting*. Este fato também é observado no sentido contrário, quando a métrica diminui para os dois métodos de geração.

Além disso, percebeu-se que as métricas baseadas em classificador de vizinhança (N1, N2, N3 e N4) são mais suscetíveis ao *Boosting* enquanto as medidas de complexidade baseadas em um classificador linear (L1, L2 e L3) tem maior suscetibilidade aos conjuntos gerados pelo *Bagging*.

Tabela 4.6: Titulo temporário

Sentido	Métricas	Bagging	Boosting
⇑	F1	-0,373	-0,411
⇓	F2	0,290	0,281
⇑	F3	-0,699	-0,655
⇑	F4	-0,355	-0,403
⇓	L1	-0,065	-0,283
⇓	L2	-0,283	-0,003
⇓	L3	-0,228	-0,051
⇓	N1	-0,695	-0,762
⇓	N2	-0,885	-0,889
⇓	N3	-0,682	-0,791
⇓	N4	0,401	0,482
⇓	T1	0,104	0,088
⇑	D2	-0,311	-0,312
⇓	D3	-0,474	-0,223

Na primeira coluna da Tabela 4.6 são apresentados indicadores visuais que representam o comportamento desejado para cada métrica. Neste sentido, quando a seta está voltada para cima entende-se que a medida que o índice aumenta, a complexidade do conjunto diminui. Já as setas que apontam para baixo indicam que quando o valor métrica decresce, a complexidade do conjunto também diminui.

Capítulo 5

Conclusões

Neste trabalho realizou-se uma análise com objetivo de estudar a relação entre as assinaturas de complexidade de um conjunto de dados perante o tamanho do conjunto. Esperava-se, através da pesquisa, identificar quais métricas são mais suscetíveis à variação do tamanho do conjunto de dados.

Para realizar tal análise foram implementados dois algoritmos de geração de subconjuntos: *Bagging* e *Boosting*. Para cada conjunto de dados gerado estimou-se sua assinatura de complexidade. De forma a avaliar se o tamanho do conjunto influencia nesta assinatura, as proporções adotadas nos métodos de gerações foram variadas em 10%, 33%, 50% e 66%. Com o intuito de formar um protocolo experimental robusto, aplicou-se o processo a 26 bases, submetidas a 20 replicações.

Analisando-se os resultados alcançados pôde-se perceber que algumas métricas são mais suscetíveis a variações no tamanho do conjunto. Outro ponto notado foi de que alguns descritores foram mais influenciados pela dimensionalidade das bases ou mesmo pelo número de classes. Tais fatos foram observados em cenários em que a correlação para uma mesma métrica apresentou valores positivos e negativos.

Os descritores de complexidade que mostraram maior correlação com o número de instâncias que compõe os conjuntos foram justamente aquelas que apresentaram apenas correlação positivas ou negativas para todas as bases. Assim, pode-se afirmar que as métricas que sofrem mais influência do tamanho do conjunto são: F3, F4, N1, N2 e N3, enquanto as menos suscetíveis foram: L3, T1, F1, F2, L2, N4, D2 e D3. Ou seja, estas últimas são mais indicadas em cenários em que pode haver variação do número de instâncias do conjunto.

Observou-se, através dos experimentos realizados, que as correlações existentes entre os

algoritmos de *Bagging* e *Boosting* foram bastante similares, de forma que em apenas alguns descritores houve uma diferença significativa nos resultados obtidos da correlação.

Para uma análise mais detalhada entre os algoritmos de geração de subconjuntos, seria interessante a comparação entre *Bagging*, *Boosting* e *Random Subspace*. Dessa forma poderiam ser analisados fatores como a quantidade de atributos dos dados no comportamento das assinaturas de complexidade.

Poderia ser explorado também o comportamento das instâncias de um conjunto dentro do espaço de características e como estes atributos afetam a complexidade do conjunto.

Seria interessante também a proposição de novos descritores de complexidade, focando principalmente em uma métrica que não seja suscetível ao tamanho do conjunto de dados e que também não varie de acordo com o número de atributos.

Referências Bibliográficas

ALCALá-FDEZ, J. et al. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, v. 17, n. 2-3, p. 255–287, 2011. Cited By 275.

ALTMAN, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, v. 46, n. 3, p. 175–185, May 1992.

AUGSPURGER, T. et al. *Pandas*. 2018. Consultado na INTERNET: http://pandas.pydata.org/, 2018. Acesso em: 03 maio 2018.

BACHE, K.; LICHMAN, M. *UCI Machine Learning Repository*. 2013. Disponível em: <http://archive.ics.uci.edu/ml>. Acesso em: 16 junho 2018.

BISHOP, C. M. *Natural Networks for Pattern Recognition*. 1. ed. New York: Oxfor University Press Inc, 1995.

BREIMAN, L. Bagging predictors. machine learning. *Machine Learning*, United States, v. 24, n. 2, p. 123–140, August 1996.

BRITTO JR., A. S.; SABOURIN, R.; OLIVEIRA, L. E. S. Dynamic selection of classifiers - a comprehensive review. *Pattern Recognition*, v. 47, n. 11, p. 3665 – 3680, 2014. ISSN 0031-3203.

BRUN, A. L. *Geração e Seleção de classificadores com base na Complexidade do Problema*. Tese (Doutorado) — Pontifica Universidade Católica do Paraná, 2017.

FILHO, D. B. F.; SILVA JR., J. A. Desvendando os mistérios do coeficiente de correlação de pearson (r). *Revista Política Hoje-ISSN: 0104-7094*, Pernambuco, v. 18, n. 1, p. 115–146, January 2010.

FREUND, R. E. S. Y. Experiments with a new boosting algorithm. *ICML'96 Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, Bari, Italy, v. 24, n. 2, p. 148–156, July 1996.

HO, L. T. T. K. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell., IEEE Computer Society, Washington*, Washington DC, United States, v. 20, n. 8, p. 832–844, August 1998.

HO, T. K.; BASU, M. Measuring the complexity of classification problems. In: *Pattern Recognition, 2000. Proceedings. 15th International Conference on*. [S.l.: s.n.], 2000. v. 2, p. 43–47 vol.2. ISSN 1051-4651.

HO, T. K.; BASU, M. Complexity measures of supervised classification problems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 24, n. 3, p. 289–300, Mar 2002. ISSN 0162-8828.

HOEKSTRA, A.; DUIN, R. P. W. On the nonlinearity of pattern classifiers. *Pattern Recognition, International Conference on*, IEEE Computer Society, Los Alamitos, CA, USA, v. 4, p. 271, 1996. ISSN 1051-4651.

KING, R. D.; FENG, C.; SUTHERLAND, A. *StatLog: Comparison of Classification Algorithms on Large Real-World Problems*. 1995.

KITTLER, J. et al. On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 20, n. 3, p. 226–239, 1998. ISSN 0162-8828.

KUNCHEVA, L. L. *Combining Pattern Classifiers*. 1. ed. Hoboken, New Jersey: JOHN WILEY & SONS, INC, 2004.

LANDEROS, A. I. *Data complexity and classifier selection*. 2008. 180 p. Copyright - Copyright ProQuest, UMI Dissertations Publishing 2008; Última atualização em - 2014-01-20; Primeira página - n/a; M3: Ph.D. Disponível em: <https://search.proquest.com/docview/304682789>. Acesso em: 18 maio 2018.

MACIà, N.; ORRIOLS-PUIG, A.; BERNADó-MANSILLA, E. In search of targeted-complexity problems. In: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2010. (GECCO '10), p. 1055–1062. ISBN 978-1-4503-0072-8.

McCULLOCH, W. H. P. W. S. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, v. 5, n. 4, p. 115–133, 1943.

MICHIE, D. et al. (Ed.). *Machine Learning, Neural and Statistical Classification*. Upper Saddle River, NJ, USA: Ellis Horwood, 1994. ISBN 0-13-106360-X.

MITCHEL, T. M. *Machine Learning: An artificial intelligence approach*. 1. ed. New York: McGraw-Hill Science/Engineering/Math, 1997.

MOLLINEDA, R. A.; SÁNCHEZ, J.; SOTOCA, J. A meta-learning framework for pattern classification by means of data complexity measures. v. 10, p. 31–38, December 2006.

ORRIOLS-PUIG, A.; MACIà, N.; HO, T. K. *Documentation for the Data Complexity Library in C++*. Barcelona, Spain, 2010. Disponível em: <http://dcol.sourceforge.net/>. Acesso em: 26 maio 2018.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, n. 12, p. 2825–2830, October 2011.

PONTI JR., M. P. Combining classifiers: From the creation of ensembles to the decision fusion. In: *Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2011 24th SIBGRAPI Conference on*. [S.l.: s.n.], 2011. p. 1–10.

QUINLAN, J. R. Induction of decision trees. *Machine Learning*, v. 1, n. 1, p. 81–106, Mar 1986. ISSN 1573-0565.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 1. ed. Englewood Cliffs, New Jersey 07632: Prentice Hall, 2003.

SCHöLKOPF, B. The kernel trick for distances. In: *Proceedings of the 13th International Conference on Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2000. (NIPS'00), p. 283–289. Disponível em: <http://dl.acm.org/citation.cfm?id=3008751.3008793>. Acesso em: 18 maio 2018.

SILVA JR., E. J. da. *Método de Classificação em Cascata de Dois Níveis: Uma Alternativa para a Redução do Custo de Sistemas Baseados em Múltiplos Classificadores*. Tese (Doutorado) — Pontifica Universidade Católica do Paraná, june 2015.

SÁNCHEZ, J. S.; MOLLINEDA, R. A.; SOTOCA, J. M. An analysis of how training data complexity affects the nearest neighbor classifiers. *Pattern Anal. Appl.*, Springer-Verlag, London, UK, UK, v. 10, n. 3, p. 189–201, jul. 2007. ISSN 1433-7541.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining*. Us ed. [S.l.]: Addison Wesley, 2005. Hardcover. ISBN 0321321367.

VIEIRA, S. *Introdução a Bioestatistica*. [S.l.]: Elsevier Editora Ltda., 2015. ISBN 9788535283990.

VRIESMANN, L. M. *Seleção dinâmica de subconjunto de classificadores: Abordagem baseada em acurácia local.* Tese (Doutorado) — Pontifica Universidade Católica do Paraná, june 2012.