

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

MATHEUS ALVES

**Social-Training: Aprendizado  
Semi-Supervisionado Utilizando Funções de  
Escolha Social**

Dissertação apresentada como requisito parcial para  
a obtenção do grau de Mestre em Ciência da  
Computação

Orientador: Profa. Dra. Ana Lúcia Cetertich Bazzan

Porto Alegre  
2017

## CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Alves, Matheus

Social-Training: Aprendizado Semi-Supervisionado Utilizando Funções de Escolha Social / Matheus Alves. – Porto Alegre: PPGC da UFRGS, 2017.

84 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2017. Orientador: Ana Lúcia Cetertich Bazzan.

I. Bazzan, Ana Lúcia Cetertich. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof<sup>a</sup>. Jane Fraga Tutikian

Pró-Reitor de Pós-Graduação: Prof. Celso Giannetti Loureiro Chaves

Diretora do Instituto de Informática: Prof<sup>a</sup>. Carla Maria Dal Sasso Freitas

Coordenador do PPGC: Prof. João Luiz Dihl Comba

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

## RESUMO

Dada a grande quantidade de dados gerados atualmente, apenas uma pequena porção dos mesmos pode ser rotulada manualmente por especialistas humanos. Isso é um desafio comum para aplicações de aprendizagem de máquina. Aprendizado semi-supervisionado aborda este problema através da manipulação dos dados não rotulados juntamente aos dados rotulados. Entretanto, se apenas uma quantidade limitada de exemplos rotulados está disponível, o desempenho da tarefa de aprendizagem de máquina (e.g., classificação) pode ser não satisfatória. Diversas soluções abordam este problema através do uso de uma *ensemble* de classificadores, visto que essa abordagem aumenta a diversidade dos classificadores. Algoritmos como o co-training e o tri-training utilizam múltiplas partições de dados ou múltiplos algoritmos de aprendizado para melhorar a qualidade da classificação de instâncias não rotuladas através de concordância por maioria simples. Além disso, existem abordagens que estendem esta ideia e adotam processos de votação menos triviais para definir os rótulos, como eleição por maioria ponderada, por exemplo. Contudo, estas soluções requerem que os rótulos possuam um certo nível de confiança para serem utilizados no treinamento. Conseqüentemente, nem toda a informação disponível é utilizada. Por exemplo: informações associadas a níveis de confiança baixos são totalmente ignoradas. Este trabalho propõe uma abordagem chamada social-training, que utiliza toda a informação disponível na tarefa de aprendizado semi-supervisionado. Para isto, múltiplos classificadores heterogêneos são treinados com os dados rotulados e geram diversas classificações para as mesmas instâncias não rotuladas. O social-training, então, agrega estes resultados em um único rótulo por meio de funções de escolha social que trabalham com agregação de rankings sobre as instâncias. Especificamente, a solução trabalha com casos de classificação binária. Os resultados mostram que trabalhar com o ranking completo, ou seja, rotular todas as instâncias não rotuladas, é capaz de reduzir o erro de classificação para alguns conjuntos de dados da base da UCI utilizados.

**Palavras-chave:** Aprendizado semi-supervisionado. sistemas de múltiplos classificadores. funções de escolha social.

## **Social-Training: Semi-Supervised Learning Using Social Choice Functions**

### **ABSTRACT**

Given the huge quantity of data currently being generated, just a small portion of it can be manually labeled by human experts. This is a challenge for machine learning applications. Semi-supervised learning addresses this problem by handling unlabeled data alongside labeled ones. However, if only a limited quantity of labeled examples is available, the performance of the machine learning task (e.g., classification) can be very unsatisfactory. Many solutions address this issue by using a classifier ensemble because this increases diversity. Algorithms such as co-training and tri-training use multiple views or multiple learning algorithms in order to improve the classification of unlabeled instances through simple majority agreement. Also, there are approaches that extend this idea and adopt less trivial voting processes to define the labels, like weighted majority voting. Nevertheless, these solutions require some confidence level on the label in order to use it for training. Hence, not all information is used, i.e., information associated with low confidence level is disregarded completely. An approach called social-training is proposed, which uses all information available in the semi-supervised learning task. For this, multiple heterogeneous classifiers are trained with the labeled data and generate diverse classifications for the same unlabeled instances. Social-training then aggregates these results into a single label by means of social choice functions that work with rank aggregation over the instances. The solution addresses binary classification cases. The results show that working with the full ranking, i.e., labeling all unlabeled instances, is able to reduce the classification error for some UCI data sets used.

**Keywords:** semi-supervised learning, classifier ensembles, social choice functions.

## LISTA DE ABREVIATURAS E SIGLAS

AER *Average Error Rate* - Taxa de erro médio

ML *Machine Learning* - Aprendizado de máquina

SCFs *Social Choice Functions* - Funções de escolha social

## LISTA DE FIGURAS

Figura 3.1 Comparação com abordagens similares .....	20
Figura 4.1 Fluxograma Social-Training .....	29

## LISTA DE TABELAS

Tabela 2.1 Exemplo de Ranking de Preferência .....	16
Tabela 3.1 Comparação de abordagens relacionadas de acordo com os seguintes critérios: (i): tipo de aprendizado (semi: semi-supervisionado; sup: supervisionado); (ii): classificador(s) homogêneos? (sim ou não); (iii) métodos de classificação reportados nos testes; (iv): SCF usada (B: Borda; C: Copeland; f: footrule; m: maioria simples; wm: maioria ponderada; ra: método de agregação de rankings proposto pelo trabalho MPBM; fa: concordância total; e: escolha de expert); (v): tipo de partição (h: horizontal; v: vertical; n: sem partição) .....	18
Tabela 4.1 Parâmetros de Entrada e Saída Social-Training .....	24
Tabela 4.2 Medidas de Avaliação Obtidas .....	26
Tabela 4.3 Tecnologias Utilizadas para o Desenvolvimento do Protótipo .....	27
Tabela 5.1 Organização das medidas de avaliação.....	32
Tabela 5.2 Características dos conjuntos de dados utilizados pelo tri-training.....	33
Tabela 5.3 Comparação com Tri-Training. (AER $\pm$ desvio padrão) .....	33
Tabela 5.4 Resultados individuais <i>diabetes</i> 80%. ( $\pm$ desvio padrão).....	36
Tabela 5.5 Resultados individuais <i>diabetes</i> 60%. ( $\pm$ desvio padrão).....	37
Tabela 5.6 Resultados individuais <i>diabetes</i> 40%. ( $\pm$ desvio padrão).....	38
Tabela 5.7 Resultados individuais <i>diabetes</i> 20%. ( $\pm$ desvio padrão).....	39
Tabela 5.8 Resultados individuais <i>german</i> 80%. ( $\pm$ desvio padrão).....	40
Tabela 5.9 Resultados individuais <i>german</i> 60%. ( $\pm$ desvio padrão).....	41
Tabela 5.10 Resultados individuais <i>german</i> 40%. ( $\pm$ desvio padrão).....	42
Tabela 5.11 Resultados individuais <i>german</i> 20%. ( $\pm$ desvio padrão).....	43
Tabela 5.12 Resultados individuais <i>wdbc</i> 80%. ( $\pm$ desvio padrão).....	44
Tabela 5.13 Resultados individuais <i>wdbc</i> 60%. ( $\pm$ desvio padrão).....	45
Tabela 5.14 Resultados individuais <i>wdbc</i> 40%. ( $\pm$ desvio padrão).....	46
Tabela 5.15 Resultados individuais <i>wdbc</i> 20%. ( $\pm$ desvio padrão).....	47
Tabela 5.16 Resultados individuais <i>kr-vs-kp</i> 80%. ( $\pm$ desvio padrão) .....	49
Tabela 5.17 Resultados individuais <i>kr-vs-kp</i> 60%. ( $\pm$ desvio padrão) .....	50
Tabela 5.18 Resultados individuais <i>kr-vs-kp</i> 40%. ( $\pm$ desvio padrão) .....	51
Tabela 5.19 Resultados individuais <i>kr-vs-kp</i> 20%. ( $\pm$ desvio padrão) .....	52
Tabela 5.20 Características dos conjuntos de dados utilizados pelo DCL.....	53
Tabela 5.21 Comparação com Democratic Co-Learning (DCL). (AER $\pm$ desvio padrão) ....	54
Tabela 5.22 Resultados individuais <i>vote</i> 80%. ( $\pm$ desvio padrão).....	55
Tabela 5.23 Resultados individuais <i>vote</i> 60%. ( $\pm$ desvio padrão).....	56
Tabela 5.24 Resultados individuais <i>vote</i> 40%. ( $\pm$ desvio padrão).....	57
Tabela 5.25 Resultados individuais <i>vote</i> 20%. ( $\pm$ desvio padrão).....	58
Tabela 5.26 Comparações entre grupos de classificadores (AER (Algoritmos de aprendizado)) .....	60
Tabela 5.27 Resultados individuais <i>ionosphere</i> 80%. ( $\pm$ desvio padrão).....	61
Tabela A.1 Resultados individuais <i>bupa</i> 80%. ( $\pm$ desvio padrão) .....	71
Tabela A.2 Resultados individuais <i>tic-tac-toe</i> 80%. ( $\pm$ desvio padrão).....	72
Tabela A.3 Resultados individuais <i>bupa</i> 60%. ( $\pm$ desvio padrão) .....	74
Tabela A.4 Resultados individuais <i>ionosphere</i> 60%. ( $\pm$ desvio padrão).....	75
Tabela A.5 Resultados individuais <i>tic-tac-toe</i> 60%. ( $\pm$ desvio padrão).....	76
Tabela A.6 Resultados individuais <i>bupa</i> 40%. ( $\pm$ desvio padrão) .....	78

Tabela A.7 Resultados individuais <i>ionosphere</i> 40%. ( $\pm$ desvio padrão).....	79
Tabela A.8 Resultados individuais <i>tic-tac-toe</i> 40%. ( $\pm$ desvio padrão).....	80
Tabela A.9 Resultados individuais <i>bupa</i> 20%. ( $\pm$ desvio padrão) .....	82
Tabela A.10 Resultados individuais <i>ionosphere</i> 20%. ( $\pm$ desvio padrão).....	83
Tabela A.11 Resultados individuais <i>tic-tac-toe</i> 20%. ( $\pm$ desvio padrão).....	84

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>10</b>
<b>2 EMBASAMENTO TEÓRICO</b> .....	<b>13</b>
2.1 Tarefas de classificação em ML .....	13
2.2 Funções de escolha social.....	15
<b>3 TRABALHOS RELACIONADOS</b> .....	<b>17</b>
<b>4 SOLUÇÃO PROPOSTA</b> .....	<b>23</b>
4.1 Algoritmo .....	23
4.2 Medidas de avaliação .....	26
4.3 Tecnologias.....	27
4.4 Considerações Finais .....	28
<b>5 EXPERIMENTOS</b> .....	<b>30</b>
5.1 Configuração de Instanciação.....	30
5.2 Organização dos Resultados Individuais dos Algoritmos de Aprendizado .....	31
5.3 Comparação com Tri-Training .....	32
5.4 Comparação com Democratic Co-Learning.....	53
5.5 Testes para prova de conceito .....	59
5.6 Considerações Finais .....	62
<b>6 CONCLUSÃO</b> .....	<b>63</b>
<b>REFERÊNCIAS</b> .....	<b>66</b>
<b>APÊNDICE A — MÉTRICAS INDIVIDUAIS DOS CLASSIFICADORES</b> .....	<b>69</b>
A.0.1 Taxa de dados não rotulados de 80% .....	71
A.0.2 Taxa de dados não rotulados de 60% .....	74
A.0.3 Taxa de dados não rotulados de 40% .....	78
A.0.4 Taxa de dados não rotulados de 20% .....	82

## 1 INTRODUÇÃO

A quantidade de informação digital armazenada em diversos tipos de bancos de dados aumenta diariamente enquanto observa-se um crescimento vertiginoso no uso de dispositivos móveis. Isto traz mais e mais desafios para a área de aprendizado de máquina (*machine learning* - ML) em geral, e para tarefas de classificação em particular, pois a alta frequência com a qual informações são geradas impede que especialistas humanos avaliem e rotulem cada instância de um conjunto de dados manualmente, ou pelo menos torna este processo muito custoso. Para endereçar este problema, utiliza-se aprendizado semi-supervisionado. Neste tipo de aprendizado, os classificadores são treinados em conjuntos de dados que contenham uma quantidade de dados não rotulados juntamente aos dados rotulados.

Grosso modo, em sua versão básica, aprendizado semi-supervisionado funciona da seguinte maneira: seja  $L$  o conjunto de instâncias rotuladas de um conjunto de dados, e  $U$  o conjunto de instâncias não rotuladas. Um classificador é inicialmente treinado utilizando instâncias de  $L$  e então é utilizado para rotular instâncias do conjunto  $U$ , criando um novo grupo de instâncias rotuladas,  $L'$ . A seguir, o classificador é treinado novamente utilizando os dados tanto de  $L$  como de  $L'$ .

Dependendo da porção de instâncias não rotuladas (além de outros fatores associados aos métodos de classificação em si), este procedimento pode gerar classificadores de baixa qualidade (ZHOU; LI, 2005). Uma abordagem comum em ML para superar um desempenho baixo ou instável por parte dos métodos de aprendizado é utilizar uma *ensemble* de classificadores. Esta alternativa não é apenas robusta para este problema em específico, mas possui o potencial de melhorar os resultados quando os classificadores são precisos e diversos, ou seja, produzem erros não correlacionados (DIETTERICH, 2000). Esta diversidade entre os classificadores pode ser introduzida de diversas maneiras, incluindo manipulação no nível dos dados (instâncias ou atributos) e combinação de distintos métodos de aprendizado (KUNCHEVA, 2004).

Da mesma maneira, em aprendizado semi-supervisionado, alguns trabalhos mostraram que a utilização de múltiplos classificadores para classificar instâncias do conjunto  $U$  pode levar a uma melhora na acurácia dos rótulos atribuídos. Estas soluções se usam da opinião da maioria dos classificadores para determinar quais das instâncias não rotuladas em  $U$  serão adicionadas a  $L'$ . Especificamente, em (BLUM; MITCHELL, 1998) e (ZHOU; LI, 2005), novas instâncias são adicionadas ao conjunto  $L'$  apenas se a maioria entre dois e três classificadores, respectivamente, concordam no rótulo proposto.

Porém, o uso de mais de um classificador apresenta alguns desafios, como: (i) que

tipo de métodos de aprendizado constituem os classificadores; (ii) como agregar os resultados individuais dos classificadores de uma maneira que possamos incluir não apenas os rótulos nos quais houve concordância da maioria, mas, em vez disso, utilizar os resultados individuais dos classificadores em sua integridade, ou seja, seus rótulos propostos para cada instância; (iii) quais dentre os diversos algoritmos de aprendizado apresentam os melhores resultados sobre um conjunto de dados específico. O foco deste trabalho é sobre o segundo e terceiro itens.

Especificamente, o uso de sistemas de votação que vão além de eleições por maioria simples é proposto. Para este fim utilizam-se funções de escolha social (*Social Choice Functions* - SCF) que são baseadas em agregação de rankings. Para tanto, cada classificador cria um ranking de preferência sobre as instâncias em  $U$  baseado na confiança que o mesmo possui pelo rótulo proposto. Isto é possível graças a se utilizarem casos de classificação binária, onde só existem dois rótulos possíveis (positivo e negativo). Como este nível de confiança é conhecido para cada instância em  $U$ , estes rankings são utilizados como entrada para uma SCF, a qual então agrega os resultados e gera o conjunto  $L'$ . Uma diferença desta abordagem é que, como o ranking é completo, todas as instâncias não rotuladas do conjunto de dados recebem um rótulo. Isto permite a inclusão de mais informação ao processo de ampliar o conjunto  $L$  original. De fato, todas as instâncias de  $U$  recebem um rótulo e são adicionadas a  $L'$ , o qual é então combinado com  $L$  para classificar instâncias a seguir (um conjunto de teste, por exemplo).

Outra diferença nesta abordagem se comparada a (BLUM; MITCHELL, 1998) e (ZHOU; LI, 2005) é que utilizam-se múltiplos classificadores heterogêneos, ou seja, com diferentes métodos de aprendizado em cada um dos múltiplos classificadores. Isto é importante, pois cada algoritmo de classificação possui um viés diferente e portanto gera classificações diferentes para os mesmos dados; estas classificações podem ser vistas como "pontos de vista" sobre os dados, como descrito por (RECAMONDE-MENDOZA; BAZZAN, 2016) e (ZHOU; LI, 2005). Como este comportamento pode ser visto como uma interação social entre o conjunto de classificadores, o algoritmo proposto é chamado *Social-Training*. Este tipo de comportamento permite uma avaliação clara e objetiva sobre quais algoritmos de aprendizado apresentam os melhores resultados em um conjunto de dados específico.

Resumindo, apesar do fato que diversas abordagens para aprendizado semi supervisionado existam, este trabalho é baseado em abordagens que utilizam a ideia geral de uma *ensemble* de classificadores. Estas ideias foram parcialmente exploradas por (ZHOU; GOLDMAN, 2004), (JIANG; ZENG; ZHANG, 2013), (HADY; SCHWENKER, 2010) e (ROY; GHOSH; GHOSH, 2014) (sistemas de votação com algum tipo de maioria ponderada), e por (RECAMONDE-MENDOZA; BAZZAN, 2016) e (MATSUBARA et al., 2008) (uso de agregação de rankings

para definição dos rótulos das instâncias).

Este trabalho assume que cada classificador tem acesso a todo o conjunto de dados ( $L$  mais  $U$ ). Uma nova estratégia para a tarefa de aprimorar  $L$  é proposta: ao invés de adicionar a  $L$  apenas as instâncias de  $U$  para as quais houve concordância entre os classificadores quanto ao rótulo, todas as instâncias de  $U$  são adicionadas a  $L$ . Como mencionado, isto é possível graças ao uso de funções de escolha social que são baseadas em agregação de rankings. No que tange aos conjuntos de dados, cenários nos quais estes possuem uma quantidade variável de instâncias não rotuladas (entre 20 e 80% a partir de um conjunto de treinamento composto por 75% das instâncias) são assumidos.

A organização do texto da dissertação se dá da seguinte maneira: no Capítulo 2 os conceitos gerais sobre SCFs, métodos de aprendizado semi supervisionado e ML coletivo são introduzidos. Bem como comparações sobre as SCFs utilizadas no trabalho. No Capítulo 3 os trabalhos relacionados são apresentados. Neste capítulo, também são discutidas as diferenças entre os trabalhos relacionados e esta abordagem. Já no Capítulo 4 a solução proposta é apresentada em detalhes, tendo seu algoritmo discutido e suas variáveis de entrada e saída caracterizadas. A seguir, no Capítulo 5, os experimentos realizados são apresentados. Este capítulo contém os resultados específicos para cada conjunto de dados e para cada classificador, bem como as comparações com outros trabalhos relacionados e os testes realizados para a prova de conceito da solução (Resultados adicionais sobre conjuntos de dados menos discutidos estão presentes no Anexo A). Apresentamos casos onde a abordagem proposta é superior às existentes e casos em que a mesma é inferior, além disso, discussões sobre quais cenários geram os melhores resultados são apresentados. Finalmente, as conclusões e trabalhos futuros são apresentados no Capítulo 6.

## 2 EMBASAMENTO TEÓRICO

Este capítulo apresenta alguns conceitos sobre tarefas de classificação em ML e funções de escolha social, os quais aparecem no decorrer da dissertação.

### 2.1 Tarefas de classificação em ML

Em ML, tarefas de classificação se referem ao problema de identificar o rótulo (ou classe) para uma instância não rotulada, após a realização de uma etapa de treinamento utilizando instâncias rotuladas. Uma abordagem comum é utilizar um algoritmo de aprendizado, como uma árvore de decisão, para criar um modelo de classificação baseado nos dados de treinamento. Neste trabalho, evita-se o uso do termo classificador para denotar um algoritmo de aprendizado; ao invés disso utiliza-se o termo classificador para o modelo de classificação que resulta do processo de treinamento. Optou-se por essa diferenciação para evidenciar questões relacionadas aos algoritmos de aprendizado em si e suas características, que não necessariamente se aplicam aos classificadores após a etapa de treinamento.

De qualquer maneira, a descrição recém-mencionada se refere basicamente ao paradigma conhecido como aprendizado supervisionado, pois o mesmo requer que todas instâncias de treinamento estejam devidamente rotuladas. Como mencionado anteriormente, este processo pode ser muito caro ou inviável. Por isso utilizou-se o paradigma de aprendizado semi-supervisionado, no qual conjuntos de dados estão apenas parcialmente rotulados. A ideia por trás deste tipo de aprendizado é treinar classificadores com os dados rotulados ( $L$ ), e então rotular as instâncias não rotuladas ( $U$ ) do conjunto de dados com estes classificadores e então refazer o treinamento dos mesmos utilizando as instâncias recém rotuladas ( $L'$ ) além do conjunto  $L$ . Entretanto, como os rótulos de  $L'$  não foram definidos por um especialista humano, estes podem conter classificações errôneas. Portanto, há necessidade de múltiplas opiniões.

Diversas soluções existentes, como (BLUM; MITCHELL, 1998), (ZHOU; LI, 2005) e (ZHOU; GOLDMAN, 2004), tentaram mitigar este problema através de soluções que usam *ensembles* de classificadores para classificar os dados em  $U$ . Uma *ensemble* de classificadores, frequentemente chamada de sistema de múltiplos classificadores (JR., 2011), é um conceito que emprega múltiplos classificadores para solucionar uma tarefa de ML. De acordo com (KUNCHEVA, 2004), as estratégias para criar uma *ensemble* de classificadores vão desde utilizar diferentes subconjuntos de atributos até utilizar diferentes conjuntos de instâncias. Este trabalho, contudo, cria uma *ensemble* no nível dos classificadores, ou seja, utiliza diferentes algoritmos

de aprendizado a fim de gerar diversidade. No caso de aprendizado semi-supervisionado, *ensembles* são utilizadas visando reduzir os erros de classificação quando classificando instâncias de  $U$ . Isto é alcançado através do uso de algum sistema de votação para definir os rótulos. Como cada classificador é induzido por um algoritmo de aprendizado diferente, ou opera sobre uma partição diferente das instâncias, e, portanto, possui um viés diferente, estes podem produzir rótulos diferentes para a mesma instância. Na literatura, algumas abordagens propõem que os rótulos sejam adicionados a  $L'$  apenas caso a maioria dos classificadores concorde com dado rótulo.

Em *ensembles* de classificadores, homogeneidade se refere à variedade de algoritmos de aprendizado utilizados para treinar os classificadores. Por exemplo: se todos os classificadores em uma *ensemble* forem treinados usando o mesmo algoritmo de aprendizado, estes classificadores são considerados homogêneos. Caso contrário, eles são heterogêneos.

Outro conceito importante diz respeito ao tipo de partição de dados (ou se partições serão criadas em primeiro lugar). Existem duas estratégias possíveis de particionamento de dados: partições horizontais e verticais. Uma partição horizontal divide um conjunto de dados baseando-se em suas instâncias, ou seja, um classificador recebe apenas um subconjunto dos dados presentes no conjunto de dados. Dados particionados horizontalmente possuem todos os atributos que estavam presentes nas instâncias originais. Já em uma partição vertical, também chamada de "*view*", cada classificador tem acesso a todas as instâncias do conjunto de dados, mas apenas a um subconjunto de seus atributos. Além disso, sobreposição pode ocorrer nestes casos.

É importante ressaltar que, quando do uso de classificadores homogêneos, é necessária a utilização de alguma estratégia a fim de obter diversidade entre estes classificadores. Para tanto, pode-se utilizar algum tipo de particionamento de dados, estratégias como *bootstrapping* ou ainda manipulação nos parâmetros de configuração dos algoritmos. Adicionalmente, lembramos que em cenários de classificadores heterogêneos, particionamento de dados é opcional já que a diversidade entre os mesmos é obtida através dos diferentes algoritmos de aprendizado. Nossa solução utiliza esta última abordagem, de maneira que todos os classificadores possam ser treinados com a totalidade do conjunto  $L$ .

Além disso, em tarefas de classificação em ML, classificação binária é atribuída a cenários nos quais uma instância pode ser classificada em apenas dois grupos de rótulos, os quais são comumente definidos como positivo e negativo. Mesmo em casos onde existem mais de dois rótulos, em classificação binária os mesmos podem ser agregados nestes dois super-grupos, por exemplo: os rótulos  $A$  e  $B$  são definidos como pertencendo à classe positiva e o rótulo  $C$  à

classe negativa.

## 2.2 Funções de escolha social

Funções de escolha social (*Social Choice Functions* - SCF) são uma série de métodos de votação, variando de métodos que visam levar em conta a agregação de rankings (preferências dos eleitores), até métodos que simplesmente elegem o candidato mais votado.

As SCFs utilizadas neste trabalho endereçam problemas existentes em eleições com mais de dois candidatos, em que um critério de maioria simples não necessariamente produz um resultado que reflete a preferência da maioria dos eleitores. Por exemplo: em uma eleição com três candidatos, dois podem apoiar uma ideia e um terceiro ser contra a mesma. Se os dois candidatos que apoiam a mesma ideia obtivessem juntos 60% dos votos (30% cada) e o terceiro candidato 40% este último venceria a eleição. Entretanto, a ideia defendida pelos candidatos derrotados era a apoiada pela maioria dos eleitores, ou seja, a proposta que agrada a minoria termina sendo a vencedora. SCFs que trabalham com agregação de rankings de preferência endereçam principalmente problemas como o do cenário descrito.

Um exemplo de SCF baseada em agregação de rankings é a função *Borda count* (de agora em diante nos referimos a *Borda count* apenas como Borda). Em Borda, cada eleitor define um ranking de preferência entre os  $N$  possíveis candidatos em uma eleição onde o candidato melhor posicionado no ranking recebe  $N$  pontos, o segundo recebe  $N - 1$  pontos, e assim por diante. Ao final da votação, os pontos de todos os candidatos são somados e o vencedor é definido.

Outra SCF que utiliza uma abordagem similar é a função *Copeland* (COPELAND, 1951). Entretanto existe uma diferença na maneira em que os candidatos são comparados: em vez de atribuir pontos a cada candidato baseado em sua posição no ranking, todos os candidatos são comparados par a par, e sua posição final é baseada no candidato que é preferido sobre todos os demais na maioria das vezes.

A fim de ilustrar a aplicação destas funções, um exemplo de ranking de preferência pode ser visto na Tabela 2.1.

De acordo com a Tabela 2.1, os resultados dos candidatos seriam os seguintes para a função Borda:

- Candidato A: 12 pontos
- Candidato B: 11 pontos

Tabela 2.1 – Exemplo de Ranking de Preferência

	<b>Eleitor 1</b>	<b>Eleitor 2</b>	<b>Eleitor 3</b>	<b>Eleitor 4</b>	<b>Eleitor 5</b>
<b>3 pontos</b>	Candidato A	Candidato A	Candidato B	Candidato B	Candidato B
<b>2 pontos</b>	Candidato C	Candidato C	Candidato A	Candidato A	Candidato A
<b>1 pontos</b>	Candidato B	Candidato B	Candidato C	Candidato C	Candidato C

- Candidato C: 7 pontos

Neste caso o vencedor seria o candidato *A*, mesmo que o candidato que mais vezes foi escolhido em primeiro lugar tenha sido o candidato *B*. Isto acontece pois para alguns eleitores o candidato *B* era o último na lista de preferência, enquanto mesmo para os que classificaram o candidato *B* em primeiro lugar, o candidato *A* era a segunda opção. Desta maneira, a eleição acaba refletindo as preferências gerais sobre os candidatos e não apenas as preferências individuais de cada eleitor.

Já para a função Copeland, os resultados seriam os seguintes:

- $B \times A: 3 \times 2$
- $A \times C: 5 \times 0$
- $B \times C: 3 \times 2$

Neste caso, o vencedor seria o candidato *B*, pois ele vence o candidato *A* por  $3 \times 2$  e o candidato *C* também por  $3 \times 2$ , o que gera o seguinte resultado:  $B > A > C$ .

### 3 TRABALHOS RELACIONADOS

Como mencionado, este trabalho foca em abordagens que lidam com métodos de aprendizado semi-supervisionado para classificação, os quais utilizam múltiplas *views* ou mais de um classificador, ou ainda diversidade nos algoritmos de aprendizado. Este capítulo descreve abordagens ou algoritmos relacionados. Trabalhos estreitamente relacionados são discutidos com maior riqueza de detalhes, enquanto outros trabalhos são explicados apenas em alto nível. Um resumo dos trabalhos relacionados pode ser visualizado na Tabela 3.1, juntamente com suas principais características. O critério para a inclusão de um trabalho na tabela foi baseado em duas premissas: trabalhos que aplicam aprendizado semi-supervisionado e/ou trabalhos que utilizam algum tipo de método de votação para agregar múltiplas classificações, focando na primeira premissa.

A Tabela 3.1 apresenta uma comparação dos métodos estreitamente relacionados ao social-training (ST), como co-training (CT), tri-training (3T), democratic co-learning (DCL), além de outros métodos relacionados (RMB (RECAMONDE-MENDOZA; BAZZAN, 2016), NC (NG; CARDIE, 2003), inter-training (JIANG; ZENG; ZHANG, 2013), MPBM (MATSUBARA et al., 2008), HS (HADY; SCHWENKER, 2010), RGG (AZIMIFAR et al., 2015), (ROY; GHOSH; GHOSH, 2014)). Os acrônimos representam o nome do método (quando disponível) ou as iniciais dos nomes dos autores. Estes aparecem na primeira coluna desta tabela (i). A coluna (ii) se refere à questão de homogeneidade dos classificadores, conforme explicado na Seção 2.1

Os métodos de classificação referenciados na coluna (iii) são: C4.5, Naïve Bayes (NB), Neural Network (NN), JRip (uma implementação Java do algoritmo ripper (COHEN, 1995)), J48, K-NN (KNN), fuzzy K-NN (fKNN) (KELLER; GRAY; GIVENS, 1985), SVM, Logistic Regression (LR), Linear Discriminant Analysis (LDA) (HASTIE; TIBSHIRANI; FRIEDMAN, 2001), Quadratic Discriminant Analysis (QDA) (HASTIE; TIBSHIRANI; FRIEDMAN, 2001), Gradient Tree Boosting (GTB) (FRIEDMAN, 2000), Decision Lists (DL) (RIVEST, 1987).

A coluna (iv) mostra as SCFs (Seção 2.2) utilizadas por cada trabalho. É importante ressaltar que "ra" é um método de agregação de rankings específico proposto por (MATSUBARA et al., 2008) e que "fa" requer concordância entre todos os classificadores, como visto em (ROY; GHOSH; GHOSH, 2014).

Ao final, a coluna (v) apresenta a estratégia de particionamento de dados utilizada, conforme discutido na Seção 2.1.

Tabela 3.1 – Comparação de abordagens relacionadas de acordo com os seguintes critérios: (i): tipo de aprendizado (semi: semi-supervisionado; sup: supervisionado); (ii): classificador(s) homogêneos? (sim ou não); (iii) métodos de classificação reportados nos testes; (iv): SCF usada (B: Borda; C: Copeland; f: footrule; m: maioria simples; wm: maioria ponderada; ra: método de agregação de rankings proposto pelo trabalho MPBM; fa: concordância total; e: escolha de expert); (v): tipo de partição (h: horizontal; v: vertical; n: sem partição)

Abordagem	(i)	(ii)	(iii)	(iv)	(v)
CT	semi	s	NB	wm	v
3T	semi	s	C4.5, NB, NN	m	h
RMB	sup	n	JRip, J48 (C4.5), KNN, NB, SVM	B, C, f	v
DCL	semi	n	KNN, NB, C4.5	wm	n
NC	semi	n	NB, DL	wm	n
Inter-training	semi	n	SVM, NB, KNN	wm	h
MPBM	sup,semi	n	NB, C4.5	ra	v
HS	semi	s,n	C4.5, KNN	wm	v
RGG	semi	n	NN, fKNN	fa	h, v
ANP	semi	n	KNN, NB, SVM	e	v
<b>ST</b>	<b>semi</b>	<b>n</b>	<b>C4.5, KNN, NB, QDA, LDA, GTB, LR</b>	<b>B, C</b>	<b>n</b>

O primeiro trabalho discutido é considerado uma das soluções mais influentes na área de aprendizado semi-supervisionado. De fato, co-training (BLUM; MITCHELL, 1998) é frequentemente referenciado como motivação ou base para outros trabalhos.

Co-training (CT na Tabela 3.1) utiliza dois classificadores e divide o conjunto  $L$  de instâncias rotuladas em duas *views*, uma *view* por classificador. Uma deficiência desse algoritmo, que é citada com frequência, é quanto a sua hipótese de que  $L$  pode ser dividido em duas *views* redundantes, e que estas duas *views* são suficientes (ou seja, podem ser utilizadas independentemente uma da outra). Isto não é sempre possível, conforme mostrado em (ZHOU; LI, 2005). Além disso, o algoritmo co-training requer que ambos os classificadores concordem no rótulo que atribuíram a uma instância para que esta instância seja adicionada ao conjunto  $L'$  (instâncias rotuladas pelo algoritmo). Esta hipótese é descartada na abordagem proposta nesta dissertação.

Outro trabalho relacionado é o algoritmo tri-training (ZHOU; LI, 2005) (referenciado como 3T na Tabela 3.1), o qual tenta aprimorar a ideia do algoritmo co-training de maneira a obter uma classificação melhor das instâncias no conjunto de instâncias não rotuladas,  $U$ , através do uso de três classificadores com diferentes partições horizontais sobre  $L$ , em vez de *views*, como utilizado no co-training. Após realizar o treinamento com as instâncias de  $L$ , cada classificador rotula as instâncias de  $U$  e estas classificações são adicionadas a  $L'$  de acordo com uma votação por maioria simples. A abordagem proposta difere desta solução em alguns pontos: uma diferença inicial é que, enquanto o social-training utiliza diferentes algoritmos de aprendizado para cada classificador, o tri-training utiliza um único algoritmo de aprendizado para

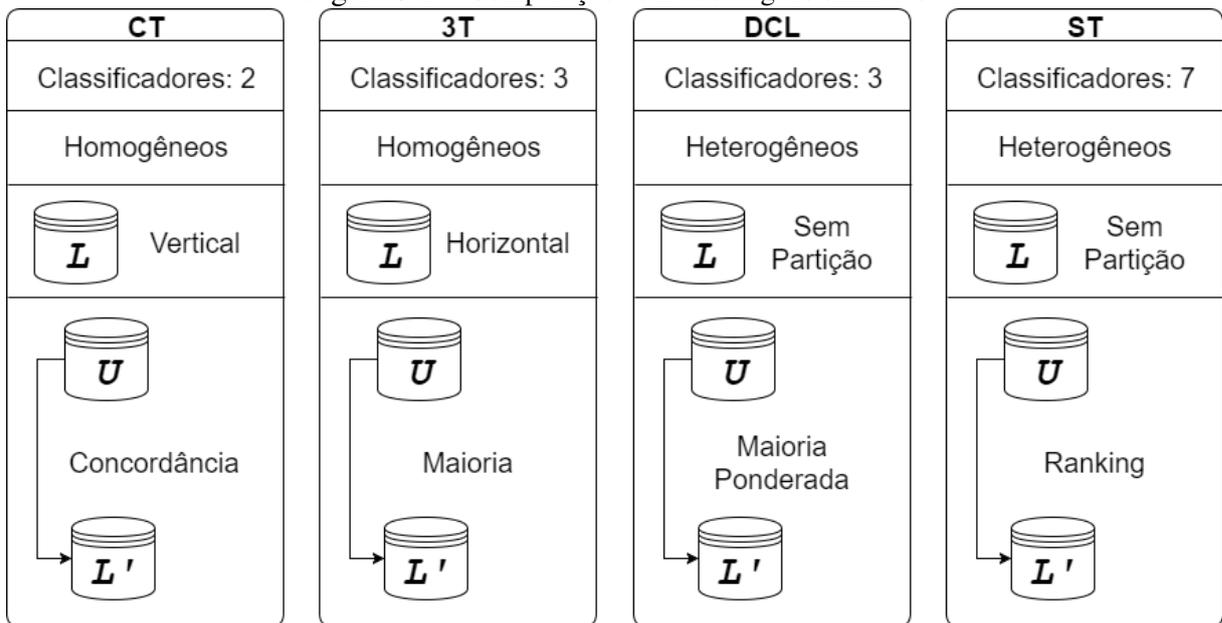
os três classificadores (embora tenha sido testado com diferentes algoritmos de classificação, como pode ser visto na Tabela 3.1, a cada teste os três classificadores são induzidos pelo mesmo algoritmo). Além disso, o algoritmo tri-training apenas adiciona uma instância que seus classificadores rotularam ao conjunto  $L'$  quando a maioria destes classificadores (dois deles, neste caso) concordam com o rótulo atribuído. A solução proposta, social-training, endereça esta questão através do uso de SCFs que trabalham com agregação de rankings, assim permitindo ao algoritmo o uso de todas as instâncias rotuladas pelos classificadores, e, portanto, realizar o treinamento de seus classificadores novamente, desta vez com mais informação. Outro ponto no qual o social-training difere das soluções co-training e tri-training é que o social-training não possui o caráter incremental destas soluções.

O algoritmo democratic co-learning (ZHOU; GOLDMAN, 2004) difere das soluções co-training e tri-training pois propõe o uso de múltiplos algoritmos de aprendizado em vez de estratégias de partição de dados. Em democratic co-learning (DCL na Tabela 3.1), todas as instâncias de  $L$  são utilizadas para treinar todos os classificadores; a diversidade é obtida através da diferença intrínseca entre estes algoritmos. Nos testes reportados, três algoritmos de aprendizado foram utilizados. A principal diferença entre as soluções democratic co-learning e social-training está na etapa de agregação dos resultados: em vez de utilizar SCFs com agregação de rankings como nossa solução, o trabalho democratic co-learning utiliza uma estratégia de maioria ponderada baseada na confiança média de cada rótulo atribuído pelos classificadores. A fim de ilustrar: utilizando três classificadores, se dois deles concordarem em um rótulo mas tiverem um baixo grau de confiança no mesmo, e um terceiro classificador divergir deste rótulo porém com um alto grau de confiança, a instância pode ser rotulada de acordo com a atribuição do terceiro classificador e adicionada ao conjunto  $L'$ . Além disso, o democratic co-learning utiliza um limiar mínimo de confiança para adicionar uma instância ao conjunto  $L'$ , enquanto o social-training adiciona todas as instâncias disponíveis.

A Figura 3.1 ilustra uma comparação entre estas soluções e o social-training (ST). Nesta figura, é possível analisar as diferentes estratégias utilizadas para a tarefa de aprendizado semi-supervisionado dentre os trabalhos mais similares ao social-training. Sendo estas: a quantidade de classificadores utilizada, a homogeneidade entre estes classificadores, as estratégias de partição do conjunto  $L$  e o método de agregação utilizado para selecionar instâncias de  $U$  para  $L'$ .

Um outro trabalho que também utiliza SCFs com agregação de rankings é (RECAMONDE-MENDOZA; BAZZAN, 2016) (referenciado como RMB daqui por diante). Em RMB, um *framework* de aprendizado multi-agente que trabalha com aprendizado supervisionado é proposto.

Figura 3.1 – Comparação com abordagens similares



Neste *framework* cinco agentes de ML utilizam diferentes algoritmos de aprendizado e representam diferentes partições verticais do mesmo conjunto de dados. O uso de SCFs, como Borda, Copeland e Footrule é proposto como uma maneira de agregar as classificações. A abordagem aqui proposta difere desta no tipo de tarefa de aprendizado e na maneira na qual as SCFs são utilizadas. Diferente do trabalho desta dissertação, RMB trabalha com aprendizado supervisionado, e portanto as SCFs são usadas para definir um rótulo final para dados do conjunto de teste apenas. Por outro lado, o algoritmo social-training utiliza as SCFs para agregar os rótulos das instâncias em  $U$  de maneira a poder utiliza-las em  $L'$ .

Outros trabalhos relacionados incluem: (NG; CARDIE, 2003), (JIANG; ZENG; ZHANG, 2013), (MATSUBARA et al., 2008), (HADY; SCHWENKER, 2010), (AZIMIFAR et al., 2015) e (ROY; GHOSH; GHOSH, 2014), os quais são brevemente apresentados a seguir.

O método em (NG; CARDIE, 2003) (referenciado como NC na Tabela 3.1) propôs uma mudança ao co-training utilizando dois algoritmos de aprendizado diferentes em vez de duas *views*. As vantagens desta abordagem são que a mesma não requer uma partição dos dados a fim de obter diversidade entre seus classificadores, já que estes são heterogêneos. Além disso, NC não depende de uma redundância entre as *views* como no caso do co-training.

Inter-training (JIANG; ZENG; ZHANG, 2013) utiliza uma estratégia que mistura maioria ponderada com maioria simples para definir os rótulos que serão adicionados a  $L'$ . Este método também utiliza estratégias nas quais os classificadores são re-treinados utilizando apenas o conjunto  $L'$  definido por outros classificadores, desconsiderando os rótulos gerados pelo próprio classificador.

O trabalho (MATSUBARA et al., 2008) (MPBM na Tabela 3.1) apresenta um método de imputação de dados para cenários em que um conjunto de dados possui instâncias com atributos faltantes. O objetivo do método de imputação de dados é tentar estimar o valor correto para um atributo faltante. Em MPBM, a classificação se dá de maneira similar ao co-training; a diferença se dá no objetivo desta classificação: determinar um atributo e não um rótulo. Além disso, este trabalho apresenta um método de agregação de rankings que é responsável por ordenar os resultados das classificações de acordo com uma pontuação definida pelos classificadores.

Em (HADY; SCHWENKER, 2010) (HS na Tabela 3.1) um novo método que combina aprendizado semi-supervisionado com aprendizado ativo é apresentado. Enquanto em aprendizado semi-supervisionado diversos trabalhos tentam utilizar apenas rótulos com o maior grau de confiança entre os rótulos definidos por seus classificadores, em aprendizado ativo o objetivo é determinar quais são os rótulos com os menores graus de confiança, para que então um especialista possa ajudar nos casos em que os classificadores estejam menos seguros quanto a um rótulo. O algoritmo apresentado em HS utiliza uma *ensemble* de classificadores que trabalha como um comitê para definir os rótulos agregados para os rótulos com menor grau de confiança. Este mesmo trabalho é apresentado com um número ainda maior de detalhes em (HADY, 2011).

O trabalho apresentado em (AZIMIFAR et al., 2015) (ANP na Tabela 3.1) consiste de um método de agregação de classificações baseado na opinião de um classificador que é definido como o *expert*. Este trabalho é focado para lidar com conjuntos de dados específicos para detecção de minas subaquáticas. Para tanto, os classificadores recebem diferentes partições verticais dos dados e são avaliados quanto aos seus desempenhos classificando instâncias das mesmas. O classificador que obtiver o melhor resultado para cada partição é definido como o *expert* sobre aquele conjunto de atributos. Desta maneira, em casos de discordância em classificações futuras, cabe ao *expert* definir qual rótulo será atribuído a uma dada instância.

Finalmente, (ROY; GHOSH; GHOSH, 2014) (referenciado como RGG na Tabela 3.1) apresentam um algoritmo que é calibrado especificamente para um conjunto de dados de imagens geográficas. Para tanto, cada algoritmo de aprendizado utilizado é configurado com os melhores parâmetros e partição de dados para este conjunto de dados em específico. Após isso, cada classificação para instâncias de  $U$  apenas é adicionada a  $L'$  se todos os classificadores concordarem com o rótulo atribuído.

Como pôde ser visto neste capítulo, a solução social-training possui algumas características que a diferenciam dos trabalhos existentes na literatura. Estas diferenças se dão em questões como o uso de todas as instâncias existentes em  $U$  para a criação do conjunto  $L'$

(utilizando, assim, mais informação para treinamento dos classificadores) ou na quantidade de classificadores utilizados (sete, sendo a média na Tabela 3.1 em torno de dois e meio). Outro ponto no qual a solução proposta nesta dissertação se diferencia é a especificidade da solução: enquanto outras soluções funcionam em apenas alguns conjuntos de dados específicos ou requerem que a mesma configuração seja sempre mantida, o algoritmo social-training é totalmente configurável de acordo com o conjunto de dados a ser utilizado, e é possível realizar ajustes específicos visando melhorar os resultados.

## 4 SOLUÇÃO PROPOSTA

Neste capítulo a solução proposta nesta dissertação é apresentada em detalhes. Na Seção 4.1 o algoritmo e seus parâmetros de entrada e saída são introduzidos e explicados. A seguir, na Seção 4.2 as métricas que são extraídas a partir da execução do algoritmo são apresentadas e detalhadas. A Seção 4.3 apresenta as tecnologias utilizadas para a criação do protótipo utilizado nos testes. Ao final, a Seção 4.4 discute os demais pontos apresentados ao longo do capítulo.

### 4.1 Algoritmo

A solução proposta nesta dissertação, *social-training*, consiste de um algoritmo que endereça aprendizado semi-supervisionado em casos de classificação binária. O objetivo principal deste algoritmo é atribuir rótulos para as instâncias não rotuladas em um conjunto de dados ( $U$ ), dado um treinamento anterior realizado com suas instâncias rotuladas ( $L$ ). Esta classificação das instâncias em  $U$  é realizada através da combinação de rótulos atribuídos por uma *ensemble* de classificadores.

Inicialmente, cada classificador rotula as instâncias em  $U$  e então as ordena de acordo com a probabilidade das mesmas pertencerem à classe positiva, criando um ranking conforme proposto por (RECAMONDE-MENDOZA; BAZZAN, 2016). Estes rankings são então utilizados como dados de entrada para uma SCF que trabalha com agregação de rankings e que define os rótulos vencedores para cada instância em  $U$ , assim gerando um novo conjunto de instâncias rotuladas pelo algoritmo,  $L'$ . Nós lembramos que o método de agregação de rankings varia de SCF para SCF, gerando, portanto, diferentes rankings (Seção 2.2).

Após isso, o algoritmo *social-training* refaz o treinamento de seus classificadores com um novo conjunto de dados que contém tanto as instâncias originalmente rotuladas ( $L$ ), além de instâncias que receberam um rótulo apenas após a etapa de agregação dos rankings ( $L'$ ). No âmbito de aprendizado semi-supervisionado, esta abordagem se aproxima de auto-aprendizado, conforme descrito por (ZHU, 2006).

A seguir, detalhes em alto nível sobre o funcionamento do algoritmo são apresentados. A Tabela 4.1 detalha os parâmetros de entrada e saída do algoritmo. Já o Algoritmo 1 exibe um pseudo código que apresenta o funcionamento interno do algoritmo, bem como conjuntos temporários que são criados e as interações entre os múltiplos componentes.

Tabela 4.1 – Parâmetros de Entrada e Saída Social-Training

Tipo	Notação	Descrição
Entrada	$L$	Um conjunto de dados contendo apenas instâncias rotuladas
Entrada	$U$	Um conjunto de dados, oriundo da mesma distribuição que compõe $L$ , contendo apenas instâncias não rotuladas
Entrada	$C$	Uma lista contendo os algoritmos de aprendizado que serão utilizados para criar os classificadores; os valores possíveis são qualquer combinação contendo pelo menos dois entre: C4.5, KNN, NB, QDA, LDA, GTB, LR
Entrada	$s$	Uma função de escolha social, a qual será utilizada para agregar os resultados dos rankings gerados pelos múltiplos classificadores. Valores possíveis são Borda ou Copeland
Entrada	$m$	O valor de uma porcentagem, a qual será utilizada como critério de corte para definir os rótulos para as instâncias do ranking agregado. As $m\%$ instâncias mais bem colocadas no ranking recebem o rótulo da classe positiva
Saída	$L'$	Uma atualização do conjunto $U$ , contendo suas instâncias rotuladas pelo uso do algoritmo
Saída	$C'$	Os classificadores criados utilizando a lista de algoritmos de aprendizado em $C$ , agora treinados usando instâncias de $L$ e $L'$

Na linha 1,  $TreinaClassificadores(Classificadores, Instâncias)$  treina todos os classificadores dado um conjunto de instâncias rotuladas, neste caso,  $L$ . Todos os algoritmos de aprendizado utilizam exatamente o mesmo conjunto de instâncias para treinar os classificadores. É importante lembrar que para *ensembles* contendo classificadores heterogêneos não é necessário existir particionamento dos dados, conforme explicado na Seção 2.1. A seguir, na linha 4 os classificadores atribuem rótulos para as instâncias de  $U$  de acordo com seu modelo. Este processo define, para cada instância, a probabilidade da mesma ser classificada com um dado rótulo, por exemplo: a probabilidade de pertencer à classe positiva. Após isso, cada classificador ordena estas instâncias de acordo com a probabilidade de cada instância pertencer à classe positiva, gerando um ranking de probabilidades (linha 5). Na linha 6, o ranking criado pelo classificador é adicionado a uma lista que contém os demais rankings a serem agregados pela SCF. Utilizando estes rankings, a SCF então consolida os resultados em um ranking final na linha 9. As instâncias são rotuladas utilizando o seguinte critério: as  $m\%$  instâncias melhores

---

**Algoritmo 1:** Pseudo Código Social-Training
 

---

**Input:**  $L$ : o conjunto de instâncias rotuladas  
 $U$ : o conjunto de instâncias não rotuladas  
 $C$ : a lista de algoritmos de aprendizado  
 $s$ : a função de escolha social a ser utilizada  
 $m$ : porcentagem de instâncias a serem rotuladas como pertencendo à classe positiva

**Resultado:** Todos os classificadores em  $C$  treinados utilizando tanto as instâncias originalmente rotuladas ( $L$ ) bem como as instâncias rotuladas pelo próprio algoritmo ( $L'$ )

```

1 TreinaClassificadores( $C, L$ )
  // treina os classificadores utilizando as instâncias
  rotuladas
2  $R \leftarrow \emptyset$ 
  // lista de rankings
3 para cada  $c$  in  $C$  faça
4   |  $c$  atribui rótulos para as instâncias em  $U$ 
5   |  $c$  gera o ranking  $r$  de acordo com a probabilidade de cada instância pertencer à
   | classe positiva
6   |  $R \leftarrow R \cup r$ 
7 fim
8  $s$  é aplicada sobre  $R$  e gera  $R'$ 
9 as primeiras  $m\%$  instâncias de  $R'$  são rotuladas como pertencendo à classe positiva
  // as demais como negativas
10 instâncias de  $R'$  são adicionadas a  $L'$ 
11 TreinaClassificadores( $C, L \cup L'$ )
  // classificadores são re-treinados utilizando as
  instâncias originalmente rotuladas e as rotuladas
  pelo algoritmo

```

---

posicionadas no ranking são classificadas como positivas. As instâncias restantes são classificadas como pertencendo à classe negativa. Finalmente, *TreinaClassificadores(Classificadores, Instâncias)* é invocado novamente na linha 11 e os classificadores são treinados com instâncias tanto de  $L$  como de  $L'$ , gerando  $C'$ . Os classificadores em  $C'$  estão então prontos para classificar instâncias de um possível conjunto de teste.

Deve-se lembrar que, devido ao uso de um ranking completo de instâncias e ao uso de SCFs, um ponto que diferencia social-training de outras soluções existentes é que rótulos são atribuídos a todas as instâncias não rotuladas disponíveis (a totalidade de  $U$ ), em vez de apenas um subconjunto de  $U$  dado algum critério de seleção (conforme visto na Figura 3.1).

Além disso, é importante ressaltar que o algoritmo social-training pode ser executado com conjuntos arbitrários de algoritmos de aprendizado (e.g., não apenas dois ou três, como é o caso nos métodos co-training e tri-training), bem como com diferentes SCFs. Isto permite

a avaliação de quais métodos de classificação são mais bem combinados (e utilizando quais SCFs) para um dado conjunto de dados.

## 4.2 Medidas de avaliação

Esta seção apresenta as medidas de avaliação que são obtidas a partir da execução do algoritmo, bem como suas fórmulas e aplicação. O objetivo destas medidas é permitir a avaliação do desempenho do algoritmo desenvolvido, além de permitir comparações com abordagens similares conforme será visto a seguir no Capítulo 5.

A Tabela 4.2 apresenta as medidas utilizadas para a avaliação deste algoritmo e suas finalidades. A seguir é possível verificar as equações das mesmas.

Tabela 4.2 – Medidas de Avaliação Obtidas

<b>Medida</b>	<b>Finalidade</b>
Taxa de Erro Médio ( <i>Average Error Rate - AER</i> )	Determina quantas instâncias foram classificadas erroneamente em um processo de classificação
Acurácia	Determina quantas instâncias foram classificadas corretamente em um processo de classificação
Precisão	Define a fração de instâncias corretamente classificadas como pertencendo à classe positiva
Revocação ( <i>Recall</i> )	Define a fração de instâncias pertencentes à classe positiva que foram corretamente classificadas pelo algoritmo
Medida-F	Média harmônica entre precisão e revocação, determina o desempenho da classificação para casos pertencentes à classe positiva

É importante ressaltar que a medida-F pode ser utilizada para calcular o desempenho de um classificador, entretanto esta medida não leva em conta instâncias classificadas corretamente como pertencendo a classe negativa, o que pode levar a resultados enviesados. As equações 4.1 a 4.5 apresentam estas medidas e utilizam as seguintes notações:

- $D$ : conjunto de dados do qual a medida foi extraída
- $e$ : número de instâncias classificadas de maneira incorreta

- $c$ : número de instâncias classificadas corretamente
- $vp$ : número de instâncias classificadas corretamente como pertencendo à classe positiva
- $fp$ : número de instâncias classificadas erroneamente como pertencendo à classe positiva
- $fn$ : número de instâncias classificadas erroneamente como pertencendo à classe negativa

$$AER = \frac{e}{|D|} \quad (4.1)$$

$$Acurácia = \frac{c}{|D|} \quad (4.2)$$

$$Precisão = \frac{vp}{vp + fp} \quad (4.3)$$

$$Revocação = \frac{vp}{vp + fn} \quad (4.4)$$

$$Medida - F = 2 \cdot \frac{Precisão \cdot Revocação}{Precisão + Revocação} \quad (4.5)$$

### 4.3 Tecnologias

Esta seção descreve as tecnologias utilizadas para o desenvolvimento do protótipo da solução proposta. A Tabela 4.3 apresenta estas tecnologias.

Tabela 4.3 – Tecnologias Utilizadas para o Desenvolvimento do Protótipo

<b>Objetivo</b>	<b>Tecnologia Utilizada</b>
Linguagem de Programação	<i>Python 3.4.3</i> (FOUNDATION, 2015)
Algoritmos de aprendizado	<i>Scikit-Learn</i> (PEDREGOSA et al., 2011)
Processamento dos conjuntos de dados	<i>Numpy</i> (WALT; COLBERT; VAROQUAUX, 2011) e <i>Pandas</i> (MCKINNEY, 2010)
Algoritmos de <i>Social Choice Functions</i>	<i>BallotBox</i> (MCGREGGOR, 2011)

Conforme pode ser visto na Tabela 4.3, utilizou-se a linguagem de programação *Python* em sua versão 3.4.3 para a criação do algoritmo. Optou-se pelo uso desta linguagem devido ao suporte de bibliotecas de ML existentes para a mesma, bem como à existência de bibliotecas com implementações das principais SCFs. Um outro motivo para a escolha foi o conhecimento profissional prévio no uso da mesma. Dado isso, para os algoritmos de aprendizado, foi utilizada a biblioteca *Scikit-Learn*, a qual possui uma documentação rica contendo diversos exemplos sobre cada função disponibilizada. Além disto, utilizaram-se as bibliotecas *Numpy* e *Pandas* para manipulação e processamento dos conjuntos de dados. Isto permitiu que conjuntos de dados que não possuíam um formato compatível com a biblioteca *Scikit-Learn* pudessem ser utilizados. Finalmente, para o desenvolvimento das SCFs, utilizou-se a biblioteca *Ballot-Box* devido à sua implementação existente de uma série de métodos de votação. O código desenvolvido durante esta dissertação é aberto e encontra na seguinte endereço do *GitHub*: <https://github.com/matheus-alves/social-training> .

Além disto, um fluxograma de uma execução deste algoritmo e a interação entre as múltiplas tecnologias utilizadas pode ser visto na Figura 4.1. Inicialmente, a solução principal *Python* recebe como entrada o conjunto de dados a ser processado, o qual foi carregado utilizando as bibliotecas *Numpy* e *Pandas*, e o divide nos múltiplos subconjuntos (Algoritmo 1 - Input). Daí repassa estas informações juntamente com a lista de classificadores ( $C$ ) criados utilizando a biblioteca *Scikit-Learn* para o processo principal do algoritmo social-training. Após o processo de criação dos rankings sobre o conjunto  $U$ , os resultados são então agregados utilizando a SCF  $s$  criada utilizando a biblioteca *BallotBox*, criando, assim, o conjunto  $L'$ . Finalmente, a classificação sobre os dados de teste é realizada. Este processo descrito na Figura 4.1 é realizado dez vezes, cada vez utilizando diferentes subconjuntos oriundos do conjunto de dados original, visando à aplicação da técnica de validação cruzada.

#### 4.4 Considerações Finais

Neste capítulo foram apresentadas as decisões envolvidas na criação do algoritmo proposto. O pseudo-código da solução (Algoritmo 1) demonstra os pontos em que a solução se diferencia das demais, seja referente à questão da criação dos rankings ou à utilização dos algoritmos de aprendizado. Além disto, as métricas que são extraídas e serão utilizadas para comparação com os resultados de outras soluções são descritas e questões técnicas de baixo



## 5 EXPERIMENTOS

Este capítulo apresenta os experimentos realizados a fim de avaliar a abordagem proposta. Na Seção 5.1, os parâmetros de configuração utilizados para a instanciação do algoritmo são apresentados em detalhes. Já a Seção 5.2 apresenta a organização utilizada para a apresentação dos resultados individuais de cada algoritmo de aprendizado. A seguir, nas Seções 5.3 e 5.4, comparações com os resultados dos métodos tri-training (ZHOU; LI, 2005) e democratic co-learning (ZHOU; GOLDMAN, 2004), respectivamente, são apresentadas. Estas seções apresentam também alguns dos resultados individuais de cada algoritmo de aprendizado. Entretanto, nem todos os resultados são apresentados nestas seções; os demais resultados podem ser encontrados no Apêndice A. Além destas comparações, a fim de validar o conceito do uso de múltiplos classificadores, testes utilizando diferentes quantidades de algoritmos de aprendizado foram realizados. A Seção 5.5 apresenta os resultados destes testes. Ao final, a Seção 5.6 apresenta as conclusões deste capítulo.

### 5.1 Configuração de Instanciação

Esta seção apresenta os parâmetros de configuração utilizados para a realização dos testes. Os detalhes destes parâmetros podem ser visualizados na Seção 4.1.

Para comparação, os testes foram realizados utilizando a mesma divisão do conjunto de dados proposta pelo algoritmo tri-training. Por exemplo, dado um conjunto de dados  $D$ , 25% dos dados são utilizados como dados de teste ( $T$ ). Os 75% restantes de  $D$  são utilizados como dados de treinamento (conjunto  $Tr$ ). Como de praxe em aprendizado semi-supervisionado, o conjunto  $Tr$  é dividido em dois grupos: rotulados ( $L$ ) e não rotulados ( $U$ ), em que algumas instâncias têm seus rótulos removidos a fim de simular cenários de conjuntos de dados com rótulos faltantes. A proporção entre  $L$  e  $U$  variou da seguinte maneira:  $U$  recebeu 20%, 40%, 60% ou 80% de  $Tr$ . É importante ressaltar que, quando realizando estas partições de  $Tr$ , a proporção de instâncias das classes positiva e negativa foi mantida como a existente no conjunto de dados original  $D$ .

Além disto, o algoritmo social-training foi instanciado utilizando como valor para o parâmetro  $C$  sete algoritmos de aprendizado, sendo estes:

- Linear Discriminant Analysis (LDA)
- Quadratic Discriminant Analysis (QDA)

- Gradient Tree Boosting (GTB)
- Logistic Regression (LR)
- K-NN (KNN)
- Naïve Bayes (NB)
- Árvore de Decisão (C4.5)

Estes algoritmos foram escolhidos, pois os mesmos também são utilizados na literatura relacionada (BLUM; MITCHELL, 1998), (ZHOU; GOLDMAN, 2004), (ZHOU; LI, 2005), (RECAMONDE-MENDOZA; BAZZAN, 2016), (NG; CARDIE, 2003), (JIANG; ZENG; ZHANG, 2013), (MATSUBARA et al., 2008), (HADY; SCHWENKER, 2010), (AZIMIFAR et al., 2015). Além disso, o parâmetro de saída de todos estes algoritmos é a probabilidade de uma instância pertencer a uma certa classe (rótulo), o que é utilizado como entrada para os rankings de instâncias (Algoritmo 1 - linha 4). Em todos os casos, os algoritmos foram instanciados utilizando sua configuração padrão.

Outra configuração necessária foi a escolha de SCFs ( $s$ ) que trabalhem com agregação de rankings. Para tanto foram utilizadas para os testes as funções: Borda e Copeland. Estas SCFs foram escolhidas, pois o seu parâmetro de entrada é um ranking de preferência e os resultados destas funções é um ranking completo de posições e não apenas o candidato vencedor.

Já para a porcentagem de rótulos da classe positiva ( $m$ ), mesmo que  $m$  possa ser definido arbitrariamente pelo usuário, nos testes que são apresentados nesta dissertação  $m$  foi instanciado utilizando como valor a probabilidade *a priori* de uma instância pertencer à classe positiva. Isto permitiu que se tivesse uma metodologia uniforme para os testes, ou seja, não dependente de um valor arbitrário para  $m$ .

Finalmente, o *workflow* presente na Figura 4.1 (Seção 4.3) foi repetido 10 vezes nos experimentos a fim de realizar validação cruzada do conjunto de dados.

## 5.2 Organização dos Resultados Individuais dos Algoritmos de Aprendizado

Esta seção apresenta a organização utilizada para a apresentação dos resultados individuais de cada algoritmo de aprendizado para cada conjunto de dados, a fim de embasar futuras discussões sobre os resultados. Os resultados são separados de acordo com a taxa de dados não rotulados utilizada (80%, 60%, 40% e 20% respectivamente), com uma tabela para cada uma destas taxas. Além disto, existe uma tabela para cada conjunto de dados. Dentro desta organização, cada tabela apresenta os seguintes dados:

- Acurácia obtida classificando  $U$  após treinamento com  $L$  (Algoritmo 1 - linha 4)
- Acurácia, Medida-F, Precisão e Revocação obtidas classificando  $T$  após treinamento com  $L$  e  $L'$

Quanto ao último item, os dados são apresentados seguindo a convenção apresentada na Tabela 5.1.

Tabela 5.1 – Organização das medidas de avaliação

<b>Acurácia (A)</b>
<b>Medida-F (F)</b>
<b>Precisão (P)</b>
<b>Revocação (R)</b>

### 5.3 Comparação com Tri-Training

Os resultados do social-training foram comparados com os do algoritmo tri-training (ZHOU; LI, 2005) para podermos avaliar o social-training contra algoritmos que apenas adicionam algumas instâncias selecionadas de  $U$  para  $L'$  e, em alguns casos, os resultados chegam a superar os do tri-training.

A fim de realizar esta comparação, social-training foi testado utilizando alguns dos conjuntos de dados utilizados pelo trabalho tri-training. Estes conjuntos de dados foram retirados do repositório UCI e consistem de casos de classificação binária. A Tabela 5.2 apresenta as características destes conjuntos de dados. É importante ressaltar que alguns dos conjuntos de dados apresentados no trabalho tri-training não foram encontrados ou possuíam um formato incompatível com as bibliotecas *Numpy*, *Scikit-Learn* e *Pandas*. Os conjuntos de dados incompatíveis foram: *australian*, *colic*, *hypothyroid* e *sick*. A razão da incompatibilidade do conjunto de dados *hypothyroid* é devido ao mesmo possuir 3 classes, e ser necessária uma modelagem adicional para transformar o mesmo em um caso de classificação binária. Já para os demais, os mesmos possuem atributos que são compostos de tipos de dados incompatíveis com a biblioteca *Scikit-Learn*.

Adicionalmente, é necessário apontar que a solução social-training aplica validação cruzada sobre seus resultados, enquanto, conforme declarado pelos autores ((ZHOU; LI, 2005) Seção 2), isso não é necessário para o tri-training. A solução apresentada nesta dissertação propõe o uso de validação cruzada pois este trabalho não utiliza particionamento de dados sobre o

Tabela 5.2 – Características dos conjuntos de dados utilizados pelo tri-training

Conjunto de dados	Número de atributos	Número de instâncias
bupa	6	345
diabetes	8	768
german	20	1000
ionosphere	34	351
kr-vs-kp	36	3196
tic-tac-toe	9	958
vote	16	435
wdbc	30	569

conjunto  $L$ , o que poderia levar a resultados enviesados. Já a solução tri-training utiliza partições aleatórias sobre os dados e, segundo os autores, validação cruzada não foi utilizada por ser um procedimento custoso computacionalmente.

Os resultados dos experimentos, dadas as variações de proporção de dados não rotulados ( $U$ ), e a comparação com o tri-training podem ser visualizados na Tabela 5.3. A métrica utilizada para comparação é a mesma utilizada em (ZHOU; LI, 2005), especificamente a taxa de erro médio (Equação. 4.1 - Seção 4.2, onde  $e$  é o número de instâncias classificadas de maneira incorreta). São apresentados a média dos resultados dos 10 *fold*s de validação cruzada e o desvio padrão.

Na Tabela 5.3 existe uma coluna específica para a SCF Borda e uma para a SCF Copeland. Para os resultados do tri-training, é apresentada a média entre os três algoritmos de aprendizado utilizados pelo trabalho (e o desvio padrão desta média), já que o tri-training utiliza classificadores homogêneos e apresenta resultados separados para cada um destes. Já para os resultados do social-training, os mesmos apresentam a média dos sete classificadores (e o desvio padrão de 10 *fold*s de validação cruzada). A melhor taxa de erro médio para cada linha é apresentada em **negrito**. Além disso, os valores de desvio padrão podem ser vistos entre parênteses.

Tabela 5.3: Comparação com Tri-Training. (AER  $\pm$  desvio padrão)

Conjunto de dados	Taxa de dados não rotulados	Tri-Training	Social-Training	
			Borda	Copeland
bupa	20%	0.398 ( $\pm 0.067$ )	<b>0.365</b> ( $\pm 0.029$ )	0.371 ( $\pm 0.040$ )
bupa	40%	0.386 ( $\pm 0.066$ )	<b>0.365</b> ( $\pm 0.022$ )	0.408 ( $\pm 0.037$ )
bupa	60%	<b>0.394</b> ( $\pm 0.052$ )	0.416 ( $\pm 0.062$ )	0.431 ( $\pm 0.023$ )
bupa	80%	<b>0.407</b> ( $\pm 0.072$ )	0.450 ( $\pm 0.103$ )	0.448 ( $\pm 0.038$ )

diabetes	20%	0.264 ( $\pm 0.030$ )	<b>0.252</b> ( $\pm 0.021$ )	0.268 ( $\pm 0.015$ )
diabetes	40%	0.269 ( $\pm 0.023$ )	<b>0.253</b> ( $\pm 0.026$ )	0.287 ( $\pm 0.020$ )
diabetes	60%	0.293 ( $\pm 0.021$ )	<b>0.255</b> ( $\pm 0.027$ )	0.313 ( $\pm 0.020$ )
diabetes	80%	0.290 ( $\pm 0.025$ )	<b>0.260</b> ( $\pm 0.030$ )	0.355 ( $\pm 0.024$ )
german	20%	0.288 ( $\pm 0.028$ )	0.133 ( $\pm 0.036$ )	<b>0.109</b> ( $\pm 0.008$ )
german	40%	0.286 ( $\pm 0.041$ )	0.220 ( $\pm 0.047$ )	<b>0.170</b> ( $\pm 0.044$ )
german	60%	0.304 ( $\pm 0.030$ )	0.328 ( $\pm 0.022$ )	<b>0.215</b> ( $\pm 0.042$ )
german	80%	0.312 ( $\pm 0.034$ )	0.471 ( $\pm 0.016$ )	<b>0.302</b> ( $\pm 0.042$ )
ionosphere	20%	<b>0.137</b> ( $\pm 0.018$ )	0.139 ( $\pm 0.028$ )	0.167 ( $\pm 0.024$ )
ionosphere	40%	<b>0.137</b> ( $\pm 0.036$ )	0.138 ( $\pm 0.024$ )	0.206 ( $\pm 0.055$ )
ionosphere	60%	0.179 ( $\pm 0.067$ )	<b>0.158</b> ( $\pm 0.027$ )	0.254 ( $\pm 0.069$ )
ionosphere	80%	<b>0.170</b> ( $\pm 0.026$ )	0.186 ( $\pm 0.039$ )	0.320 ( $\pm 0.079$ )
kr-vs-kp	20%	<b>0.053</b> ( $\pm 0.070$ )	0.270 ( $\pm 0.069$ )	0.162 ( $\pm 0.039$ )
kr-vs-kp	40%	<b>0.058</b> ( $\pm 0.076$ )	0.365 ( $\pm 0.090$ )	0.236 ( $\pm 0.095$ )
kr-vs-kp	60%	<b>0.057</b> ( $\pm 0.044$ )	0.494 ( $\pm 0.100$ )	0.310 ( $\pm 0.164$ )
kr-vs-kp	80%	<b>0.070</b> ( $\pm 0.062$ )	0.581 ( $\pm 0.114$ )	0.414 ( $\pm 0.156$ )
tic-tac-toe	20%	0.152 ( $\pm 0.133$ )	0.299 ( $\pm 0.132$ )	<b>0.138</b> ( $\pm 0.129$ )
tic-tac-toe	40%	<b>0.149</b> ( $\pm 0.126$ )	0.368 ( $\pm 0.171$ )	0.367 ( $\pm 0.187$ )
tic-tac-toe	60%	<b>0.171</b> ( $\pm 0.143$ )	0.364 ( $\pm 0.165$ )	0.368 ( $\pm 0.163$ )
tic-tac-toe	80%	<b>0.235</b> ( $\pm 0.144$ )	0.334 ( $\pm 0.153$ )	0.416 ( $\pm 0.205$ )
vote	20%	<b>0.062</b> ( $\pm 0.030$ )	<b>0.062</b> ( $\pm 0.019$ )	0.144 ( $\pm 0.025$ )
vote	40%	<b>0.068</b> ( $\pm 0.018$ )	0.086 ( $\pm 0.027$ )	0.200 ( $\pm 0.040$ )
vote	60%	<b>0.071</b> ( $\pm 0.034$ )	0.100 ( $\pm 0.033$ )	0.284 ( $\pm 0.056$ )
vote	80%	<b>0.083</b> ( $\pm 0.038$ )	0.140 ( $\pm 0.060$ )	0.397 ( $\pm 0.056$ )
wdbc	20%	0.056 ( $\pm 0.010$ )	<b>0.053</b> ( $\pm 0.015$ )	0.086 ( $\pm 0.032$ )
wdbc	40%	0.063 ( $\pm 0.020$ )	<b>0.053</b> ( $\pm 0.009$ )	0.127 ( $\pm 0.029$ )
wdbc	60%	<b>0.056</b> ( $\pm 0.028$ )	<b>0.056</b> ( $\pm 0.014$ )	0.185 ( $\pm 0.034$ )
wdbc	80%	0.063 ( $\pm 0.027$ )	<b>0.060</b> ( $\pm 0.013$ )	0.294 ( $\pm 0.050$ )

Os resultados mostrados na Tabela 5.3 demonstram que a SCF Borda é menos sensível as mudanças de quantidade de dados não rotulados, quando comparada com a função Copeland. O motivo para isto é o sistema de agregação de rankings utilizado nesta função (Seção 2.2). Mesmo se uma instância estiver ordenada de forma bastante errônea em um ranking específico,

ela acaba sendo uma das melhores posicionadas se estiver bem posicionada na maioria dos rankings.

Já a função Copeland é mais sensível a grandes variações nos rankings de instâncias. O motivo para tanto é que Copeland realiza uma comparação par a par com todas as posições do ranking. Portanto, se uma mesma instância tiver grandes divergências em sua posição entre diversos rankings, o seu posicionamento final é mais suscetível a este tipo de variação.

Além disso, é importante ressaltar que nestes testes é possível identificar que, em alguns conjuntos de dados como *diabetes*, *german* e *wdbc* (Tabelas 5.4 a 5.15), o social-training supera o tri-training em todas as taxas de dados não rotulados, ou seja, o uso de todas as instâncias não rotuladas como fonte de conhecimento diminui o erro médio.

Tabela 5.4: Resultados individuais *diabetes* 80%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.69 ( $\pm 0.02$ )	(A) 0.72 ( $\pm 0.04$ )	(A) 0.56 ( $\pm 0.04$ )
		(F) 0.60 ( $\pm 0.06$ )	(F) 0.39 ( $\pm 0.04$ )
		(P) 0.59 ( $\pm 0.06$ )	(P) 0.38 ( $\pm 0.04$ )
		(R) 0.62 ( $\pm 0.06$ )	(R) 0.40 ( $\pm 0.04$ )
GTB	0.73 ( $\pm 0.02$ )	(A) 0.74 ( $\pm 0.04$ )	(A) 0.64 ( $\pm 0.03$ )
		(F) 0.63 ( $\pm 0.06$ )	(F) 0.31 ( $\pm 0.07$ )
		(P) 0.63 ( $\pm 0.06$ )	(P) 0.46 ( $\pm 0.08$ )
		(R) 0.64 ( $\pm 0.07$ )	(R) 0.24 ( $\pm 0.07$ )
K-NN	0.70 ( $\pm 0.02$ )	(A) 0.73 ( $\pm 0.03$ )	(A) 0.62 ( $\pm 0.02$ )
		(F) 0.61 ( $\pm 0.06$ )	(F) 0.34 ( $\pm 0.03$ )
		(P) 0.62 ( $\pm 0.05$ )	(P) 0.43 ( $\pm 0.04$ )
		(R) 0.60 ( $\pm 0.08$ )	(R) 0.28 ( $\pm 0.03$ )
LDA	0.75 ( $\pm 0.01$ )	(A) 0.75 ( $\pm 0.03$ )	(A) 0.67 ( $\pm 0.03$ )
		(F) 0.63 ( $\pm 0.04$ )	(F) 0.13 ( $\pm 0.12$ )
		(P) 0.66 ( $\pm 0.05$ )	(P) 0.70 ( $\pm 0.34$ )
		(R) 0.61 ( $\pm 0.04$ )	(R) 0.08 ( $\pm 0.07$ )
LR	0.73 ( $\pm 0.01$ )	(A) 0.74 ( $\pm 0.03$ )	(A) 0.66 ( $\pm 0.02$ )
		(F) 0.62 ( $\pm 0.04$ )	(F) 0.08 ( $\pm 0.08$ )
		(P) 0.65 ( $\pm 0.05$ )	(P) 0.64 ( $\pm 0.34$ )
		(R) 0.59 ( $\pm 0.04$ )	(R) 0.05 ( $\pm 0.05$ )
NB	0.74 ( $\pm 0.01$ )	(A) 0.74 ( $\pm 0.03$ )	(A) 0.69 ( $\pm 0.05$ )
		(F) 0.62 ( $\pm 0.05$ )	(F) 0.36 ( $\pm 0.14$ )
		(P) 0.63 ( $\pm 0.05$ )	(P) 0.62 ( $\pm 0.15$ )
		(R) 0.62 ( $\pm 0.06$ )	(R) 0.26 ( $\pm 0.11$ )
QDA	0.72 ( $\pm 0.01$ )	(A) 0.74 ( $\pm 0.03$ )	(A) 0.67 ( $\pm 0.04$ )
		(F) 0.62 ( $\pm 0.05$ )	(F) 0.35 ( $\pm 0.09$ )
		(P) 0.63 ( $\pm 0.04$ )	(P) 0.57 ( $\pm 0.11$ )
		(R) 0.62 ( $\pm 0.06$ )	(R) 0.26 ( $\pm 0.08$ )

Tabela 5.5: Resultados individuais *diabetes* 60%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.68 ( $\pm 0.02$ )	(A) 0.71 ( $\pm 0.03$ )	(A) 0.59 ( $\pm 0.03$ )
		(F) 0.58 ( $\pm 0.06$ )	(F) 0.43 ( $\pm 0.04$ )
		(P) 0.58 ( $\pm 0.05$ )	(P) 0.42 ( $\pm 0.04$ )
		(R) 0.57 ( $\pm 0.09$ )	(R) 0.45 ( $\pm 0.05$ )
GTB	0.74 ( $\pm 0.02$ )	(A) 0.75 ( $\pm 0.03$ )	(A) 0.68 ( $\pm 0.02$ )
		(F) 0.63 ( $\pm 0.05$ )	(F) 0.43 ( $\pm 0.05$ )
		(P) 0.66 ( $\pm 0.05$ )	(P) 0.57 ( $\pm 0.05$ )
		(R) 0.61 ( $\pm 0.05$ )	(R) 0.35 ( $\pm 0.06$ )
K-NN	0.72 ( $\pm 0.03$ )	(A) 0.74 ( $\pm 0.03$ )	(A) 0.66 ( $\pm 0.03$ )
		(F) 0.61 ( $\pm 0.05$ )	(F) 0.44 ( $\pm 0.04$ )
		(P) 0.63 ( $\pm 0.04$ )	(P) 0.52 ( $\pm 0.06$ )
		(R) 0.59 ( $\pm 0.08$ )	(R) 0.38 ( $\pm 0.04$ )
LDA	0.76 ( $\pm 0.02$ )	(A) 0.76 ( $\pm 0.03$ )	(A) 0.72 ( $\pm 0.04$ )
		(F) 0.63 ( $\pm 0.05$ )	(F) 0.39 ( $\pm 0.14$ )
		(P) 0.68 ( $\pm 0.06$ )	(P) 0.77 ( $\pm 0.09$ )
		(R) 0.59 ( $\pm 0.05$ )	(R) 0.27 ( $\pm 0.11$ )
LR	0.75 ( $\pm 0.02$ )	(A) 0.76 ( $\pm 0.03$ )	(A) 0.71 ( $\pm 0.03$ )
		(F) 0.63 ( $\pm 0.04$ )	(F) 0.32 ( $\pm 0.13$ )
		(P) 0.68 ( $\pm 0.06$ )	(P) 0.81 ( $\pm 0.14$ )
		(R) 0.59 ( $\pm 0.03$ )	(R) 0.21 ( $\pm 0.09$ )
NB	0.75 ( $\pm 0.02$ )	(A) 0.76 ( $\pm 0.03$ )	(A) 0.73 ( $\pm 0.03$ )
		(F) 0.65 ( $\pm 0.04$ )	(F) 0.52 ( $\pm 0.07$ )
		(P) 0.66 ( $\pm 0.05$ )	(P) 0.68 ( $\pm 0.07$ )
		(R) 0.64 ( $\pm 0.04$ )	(R) 0.43 ( $\pm 0.08$ )
QDA	0.74 ( $\pm 0.01$ )	(A) 0.75 ( $\pm 0.02$ )	(A) 0.70 ( $\pm 0.03$ )
		(F) 0.62 ( $\pm 0.03$ )	(F) 0.45 ( $\pm 0.06$ )
		(P) 0.65 ( $\pm 0.04$ )	(P) 0.64 ( $\pm 0.09$ )
		(R) 0.60 ( $\pm 0.03$ )	(R) 0.35 ( $\pm 0.05$ )

Tabela 5.6: Resultados individuais *diabetes* 40%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.69 ( $\pm 0.04$ )	(A) 0.71 ( $\pm 0.02$ )	(A) 0.64 ( $\pm 0.03$ )
		(F) 0.58 ( $\pm 0.05$ )	(F) 0.49 ( $\pm 0.04$ )
		(P) 0.58 ( $\pm 0.03$ )	(P) 0.49 ( $\pm 0.04$ )
		(R) 0.58 ( $\pm 0.07$ )	(R) 0.48 ( $\pm 0.05$ )
GTB	0.75 ( $\pm 0.03$ )	(A) 0.77 ( $\pm 0.03$ )	(A) 0.72 ( $\pm 0.03$ )
		(F) 0.65 ( $\pm 0.05$ )	(F) 0.52 ( $\pm 0.06$ )
		(P) 0.69 ( $\pm 0.05$ )	(P) 0.64 ( $\pm 0.05$ )
		(R) 0.62 ( $\pm 0.05$ )	(R) 0.45 ( $\pm 0.07$ )
K-NN	0.73 ( $\pm 0.03$ )	(A) 0.73 ( $\pm 0.03$ )	(A) 0.68 ( $\pm 0.03$ )
		(F) 0.59 ( $\pm 0.05$ )	(F) 0.47 ( $\pm 0.05$ )
		(P) 0.62 ( $\pm 0.05$ )	(P) 0.55 ( $\pm 0.06$ )
		(R) 0.57 ( $\pm 0.06$ )	(R) 0.42 ( $\pm 0.05$ )
LDA	0.77 ( $\pm 0.02$ )	(A) 0.76 ( $\pm 0.03$ )	(A) 0.75 ( $\pm 0.03$ )
		(F) 0.64 ( $\pm 0.05$ )	(F) 0.54 ( $\pm 0.08$ )
		(P) 0.69 ( $\pm 0.06$ )	(P) 0.77 ( $\pm 0.07$ )
		(R) 0.60 ( $\pm 0.04$ )	(R) 0.42 ( $\pm 0.07$ )
LR	0.76 ( $\pm 0.02$ )	(A) 0.76 ( $\pm 0.03$ )	(A) 0.73 ( $\pm 0.02$ )
		(F) 0.62 ( $\pm 0.04$ )	(F) 0.48 ( $\pm 0.07$ )
		(P) 0.69 ( $\pm 0.05$ )	(P) 0.76 ( $\pm 0.06$ )
		(R) 0.56 ( $\pm 0.03$ )	(R) 0.35 ( $\pm 0.07$ )
NB	0.75 ( $\pm 0.02$ )	(A) 0.76 ( $\pm 0.03$ )	(A) 0.74 ( $\pm 0.03$ )
		(F) 0.64 ( $\pm 0.04$ )	(F) 0.57 ( $\pm 0.06$ )
		(P) 0.66 ( $\pm 0.05$ )	(P) 0.67 ( $\pm 0.06$ )
		(R) 0.63 ( $\pm 0.04$ )	(R) 0.50 ( $\pm 0.08$ )
QDA	0.74 ( $\pm 0.01$ )	(A) 0.75 ( $\pm 0.02$ )	(A) 0.72 ( $\pm 0.02$ )
		(F) 0.62 ( $\pm 0.02$ )	(F) 0.51 ( $\pm 0.06$ )
		(P) 0.65 ( $\pm 0.03$ )	(P) 0.64 ( $\pm 0.04$ )
		(R) 0.60 ( $\pm 0.02$ )	(R) 0.43 ( $\pm 0.07$ )

Tabela 5.7: Resultados individuais *diabetes* 20%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.69 ( $\pm 0.05$ )	(A) 0.73 ( $\pm 0.03$ )	(A) 0.67 ( $\pm 0.03$ )
		(F) 0.60 ( $\pm 0.05$ )	(F) 0.53 ( $\pm 0.06$ )
		(P) 0.61 ( $\pm 0.05$ )	(P) 0.53 ( $\pm 0.05$ )
		(R) 0.59 ( $\pm 0.06$ )	(R) 0.54 ( $\pm 0.09$ )
GTB	0.76 ( $\pm 0.03$ )	(A) 0.76 ( $\pm 0.03$ )	(A) 0.74 ( $\pm 0.02$ )
		(F) 0.63 ( $\pm 0.04$ )	(F) 0.58 ( $\pm 0.04$ )
		(P) 0.67 ( $\pm 0.04$ )	(P) 0.67 ( $\pm 0.04$ )
		(R) 0.60 ( $\pm 0.05$ )	(R) 0.53 ( $\pm 0.06$ )
K-NN	0.72 ( $\pm 0.04$ )	(A) 0.73 ( $\pm 0.03$ )	(A) 0.70 ( $\pm 0.03$ )
		(F) 0.59 ( $\pm 0.05$ )	(F) 0.53 ( $\pm 0.04$ )
		(P) 0.63 ( $\pm 0.05$ )	(P) 0.59 ( $\pm 0.06$ )
		(R) 0.56 ( $\pm 0.06$ )	(R) 0.49 ( $\pm 0.04$ )
LDA	0.76 ( $\pm 0.02$ )	(A) 0.77 ( $\pm 0.03$ )	(A) 0.77 ( $\pm 0.02$ )
		(F) 0.63 ( $\pm 0.03$ )	(F) 0.61 ( $\pm 0.02$ )
		(P) 0.70 ( $\pm 0.05$ )	(P) 0.75 ( $\pm 0.06$ )
		(R) 0.58 ( $\pm 0.03$ )	(R) 0.51 ( $\pm 0.01$ )
LR	0.77 ( $\pm 0.02$ )	(A) 0.76 ( $\pm 0.02$ )	(A) 0.75 ( $\pm 0.01$ )
		(F) 0.62 ( $\pm 0.03$ )	(F) 0.56 ( $\pm 0.02$ )
		(P) 0.70 ( $\pm 0.05$ )	(P) 0.74 ( $\pm 0.05$ )
		(R) 0.56 ( $\pm 0.03$ )	(R) 0.46 ( $\pm 0.02$ )
NB	0.76 ( $\pm 0.02$ )	(A) 0.75 ( $\pm 0.03$ )	(A) 0.76 ( $\pm 0.02$ )
		(F) 0.64 ( $\pm 0.03$ )	(F) 0.63 ( $\pm 0.03$ )
		(P) 0.66 ( $\pm 0.05$ )	(P) 0.67 ( $\pm 0.04$ )
		(R) 0.61 ( $\pm 0.03$ )	(R) 0.59 ( $\pm 0.03$ )
QDA	0.75 ( $\pm 0.02$ )	(A) 0.75 ( $\pm 0.02$ )	(A) 0.74 ( $\pm 0.02$ )
		(F) 0.62 ( $\pm 0.02$ )	(F) 0.58 ( $\pm 0.04$ )
		(P) 0.66 ( $\pm 0.03$ )	(P) 0.65 ( $\pm 0.04$ )
		(R) 0.58 ( $\pm 0.03$ )	(R) 0.51 ( $\pm 0.05$ )

Tabela 5.8: Resultados individuais *german* 80%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	1.00 ( $\pm 0.00$ )	(A) 0.52 ( $\pm 0.03$ )	(A) 0.64 ( $\pm 0.03$ )
		(F) 0.65 ( $\pm 0.03$ )	(F) 0.74 ( $\pm 0.02$ )
		(P) 0.66 ( $\pm 0.02$ )	(P) 0.76 ( $\pm 0.02$ )
		(R) 0.64 ( $\pm 0.04$ )	(R) 0.73 ( $\pm 0.03$ )
GTB	1.00 ( $\pm 0.00$ )	(A) 0.61 ( $\pm 0.02$ )	(A) 0.76 ( $\pm 0.04$ )
		(F) 0.75 ( $\pm 0.02$ )	(F) 0.85 ( $\pm 0.02$ )
		(P) 0.67 ( $\pm 0.01$ )	(P) 0.76 ( $\pm 0.03$ )
		(R) 0.85 ( $\pm 0.03$ )	(R) 0.96 ( $\pm 0.02$ )
K-NN	0.66 ( $\pm 0.02$ )	(A) 0.63 ( $\pm 0.03$ )	(A) 0.64 ( $\pm 0.03$ )
		(F) 0.76 ( $\pm 0.02$ )	(F) 0.77 ( $\pm 0.02$ )
		(P) 0.70 ( $\pm 0.01$ )	(P) 0.70 ( $\pm 0.01$ )
		(R) 0.84 ( $\pm 0.04$ )	(R) 0.85 ( $\pm 0.03$ )
LDA	0.70 ( $\pm 0.02$ )	(A) 0.61 ( $\pm 0.03$ )	(A) 0.79 ( $\pm 0.03$ )
		(F) 0.76 ( $\pm 0.03$ )	(F) 0.87 ( $\pm 0.02$ )
		(P) 0.67 ( $\pm 0.01$ )	(P) 0.77 ( $\pm 0.02$ )
		(R) 0.86 ( $\pm 0.05$ )	(R) 0.99 ( $\pm 0.01$ )
LR	0.99 ( $\pm 0.01$ )	(A) 0.63 ( $\pm 0.03$ )	(A) 0.77 ( $\pm 0.02$ )
		(F) 0.77 ( $\pm 0.03$ )	(F) 0.86 ( $\pm 0.01$ )
		(P) 0.68 ( $\pm 0.01$ )	(P) 0.76 ( $\pm 0.02$ )
		(R) 0.89 ( $\pm 0.05$ )	(R) 0.99 ( $\pm 0.01$ )
NB	0.99 ( $\pm 0.01$ )	(A) 0.33 ( $\pm 0.03$ )	(A) 0.64 ( $\pm 0.11$ )
		(F) 0.41 ( $\pm 0.12$ )	(F) 0.69 ( $\pm 0.20$ )
		(P) 0.52 ( $\pm 0.06$ )	(P) 0.80 ( $\pm 0.05$ )
		(R) 0.35 ( $\pm 0.12$ )	(R) 0.67 ( $\pm 0.22$ )
QDA	0.99 ( $\pm 0.01$ )	(A) 0.37 ( $\pm 0.06$ )	(A) 0.64 ( $\pm 0.11$ )
		(F) 0.41 ( $\pm 0.16$ )	(F) 0.71 ( $\pm 0.21$ )
		(P) 0.55 ( $\pm 0.10$ )	(P) 0.75 ( $\pm 0.05$ )
		(R) 0.35 ( $\pm 0.16$ )	(R) 0.76 ( $\pm 0.25$ )

Tabela 5.9: Resultados individuais *german* 60%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	1.00 ( $\pm 0.00$ )	(A) 0.63 ( $\pm 0.03$ )	(A) 0.72 ( $\pm 0.03$ )
		(F) 0.74 ( $\pm 0.02$ )	(F) 0.80 ( $\pm 0.02$ )
		(P) 0.74 ( $\pm 0.02$ )	(P) 0.81 ( $\pm 0.03$ )
		(R) 0.73 ( $\pm 0.04$ )	(R) 0.79 ( $\pm 0.04$ )
GTB	1.00 ( $\pm 0.00$ )	(A) 0.74 ( $\pm 0.02$ )	(A) 0.88 ( $\pm 0.02$ )
		(F) 0.84 ( $\pm 0.01$ )	(F) 0.92 ( $\pm 0.01$ )
		(P) 0.75 ( $\pm 0.01$ )	(P) 0.87 ( $\pm 0.02$ )
		(R) 0.95 ( $\pm 0.02$ )	(R) 0.97 ( $\pm 0.01$ )
K-NN	0.66 ( $\pm 0.02$ )	(A) 0.63 ( $\pm 0.02$ )	(A) 0.64 ( $\pm 0.03$ )
		(F) 0.76 ( $\pm 0.01$ )	(F) 0.76 ( $\pm 0.02$ )
		(P) 0.70 ( $\pm 0.01$ )	(P) 0.70 ( $\pm 0.02$ )
		(R) 0.84 ( $\pm 0.03$ )	(R) 0.84 ( $\pm 0.03$ )
LDA	0.74 ( $\pm 0.02$ )	(A) 0.76 ( $\pm 0.02$ )	(A) 0.93 ( $\pm 0.03$ )
		(F) 0.85 ( $\pm 0.01$ )	(F) 0.95 ( $\pm 0.02$ )
		(P) 0.76 ( $\pm 0.01$ )	(P) 0.92 ( $\pm 0.03$ )
		(R) 0.97 ( $\pm 0.02$ )	(R) 0.99 ( $\pm 0.01$ )
LR	1.00 ( $\pm 0.00$ )	(A) 0.76 ( $\pm 0.01$ )	(A) 0.90 ( $\pm 0.03$ )
		(F) 0.85 ( $\pm 0.01$ )	(F) 0.94 ( $\pm 0.02$ )
		(P) 0.75 ( $\pm 0.01$ )	(P) 0.88 ( $\pm 0.03$ )
		(R) 0.98 ( $\pm 0.01$ )	(R) 1.00 ( $\pm 0.00$ )
NB	1.00 ( $\pm 0.00$ )	(A) 0.57 ( $\pm 0.07$ )	(A) 0.72 ( $\pm 0.13$ )
		(F) 0.66 ( $\pm 0.10$ )	(F) 0.75 ( $\pm 0.22$ )
		(P) 0.73 ( $\pm 0.03$ )	(P) 0.88 ( $\pm 0.05$ )
		(R) 0.62 ( $\pm 0.16$ )	(R) 0.71 ( $\pm 0.23$ )
QDA	1.00 ( $\pm 0.00$ )	(A) 0.60 ( $\pm 0.09$ )	(A) 0.70 ( $\pm 0.13$ )
		(F) 0.70 ( $\pm 0.11$ )	(F) 0.75 ( $\pm 0.24$ )
		(P) 0.71 ( $\pm 0.02$ )	(P) 0.79 ( $\pm 0.07$ )
		(R) 0.72 ( $\pm 0.20$ )	(R) 0.82 ( $\pm 0.27$ )

Tabela 5.10: Resultados individuais *german* 40%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	1.00 ( $\pm 0.00$ )	(A) 0.72 ( $\pm 0.04$ )	(A) 0.82 ( $\pm 0.03$ )
		(F) 0.80 ( $\pm 0.03$ )	(F) 0.87 ( $\pm 0.02$ )
		(P) 0.81 ( $\pm 0.02$ )	(P) 0.88 ( $\pm 0.03$ )
		(R) 0.79 ( $\pm 0.04$ )	(R) 0.87 ( $\pm 0.02$ )
GTB	1.00 ( $\pm 0.000$ )	(A) 0.90 ( $\pm 0.01$ )	(A) 0.96 ( $\pm 0.02$ )
		(F) 0.93 ( $\pm 0.01$ )	(F) 0.97 ( $\pm 0.01$ )
		(P) 0.89 ( $\pm 0.01$ )	(P) 0.95 ( $\pm 0.02$ )
		(R) 0.98 ( $\pm 0.01$ )	(R) 0.99 ( $\pm 0.00$ )
K-NN	0.65 ( $\pm 0.02$ )	(A) 0.64 ( $\pm 0.03$ )	(A) 0.65 ( $\pm 0.03$ )
		(F) 0.77 ( $\pm 0.02$ )	(F) 0.77 ( $\pm 0.02$ )
		(P) 0.70 ( $\pm 0.02$ )	(P) 0.71 ( $\pm 0.02$ )
		(R) 0.85 ( $\pm 0.04$ )	(R) 0.84 ( $\pm 0.04$ )
LDA	0.74 ( $\pm 0.02$ )	(A) 0.96 ( $\pm 0.02$ )	(A) 0.99 ( $\pm 0.01$ )
		(F) 0.97 ( $\pm 0.01$ )	(F) 1.00 ( $\pm 0.00$ )
		(P) 0.95 ( $\pm 0.02$ )	(P) 0.99 ( $\pm 0.01$ )
		(R) 1.00 ( $\pm 0.01$ )	(R) 1.00 ( $\pm 0.00$ )
LR	1.00 ( $\pm 0.00$ )	(A) 0.92 ( $\pm 0.01$ )	(A) 0.97 ( $\pm 0.02$ )
		(F) 0.95 ( $\pm 0.01$ )	(F) 0.98 ( $\pm 0.01$ )
		(P) 0.90 ( $\pm 0.02$ )	(P) 0.96 ( $\pm 0.02$ )
		(R) 1.00 ( $\pm 0.00$ )	(R) 1.00 ( $\pm 0.00$ )
NB	1.00 ( $\pm 0.00$ )	(A) 0.67 ( $\pm 0.16$ )	(A) 0.72 ( $\pm 0.15$ )
		(F) 0.69 ( $\pm 0.25$ )	(F) 0.74 ( $\pm 0.21$ )
		(P) 0.84 ( $\pm 0.04$ )	(P) 0.90 ( $\pm 0.03$ )
		(R) 0.64 ( $\pm 0.27$ )	(R) 0.68 ( $\pm 0.24$ )
QDA	1.00 ( $\pm 0.00$ )	(A) 0.65 ( $\pm 0.17$ )	(A) 0.69 ( $\pm 0.16$ )
		(F) 0.67 ( $\pm 0.29$ )	(F) 0.72 ( $\pm 0.26$ )
		(P) 0.74 ( $\pm 0.08$ )	(P) 0.82 ( $\pm 0.05$ )
		(R) 0.72 ( $\pm 0.34$ )	(R) 0.75 ( $\pm 0.32$ )

Tabela 5.11: Resultados individuais *german* 20%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	1.00 ( $\pm 0.00$ )	(A) 0.87 ( $\pm 0.02$ )	(A) 0.89 ( $\pm 0.01$ )
		(F) 0.90 ( $\pm 0.01$ )	(F) 0.92 ( $\pm 0.01$ )
		(P) 0.91 ( $\pm 0.01$ )	(P) 0.93 ( $\pm 0.02$ )
		(R) 0.90 ( $\pm 0.02$ )	(R) 0.91 ( $\pm 0.02$ )
GTB	1.00 ( $\pm 0.00$ )	(A) 0.98 ( $\pm 0.01$ )	(A) 0.98 ( $\pm 0.01$ )
		(F) 0.98 ( $\pm 0.01$ )	(F) 0.99 ( $\pm 0.01$ )
		(P) 0.97 ( $\pm 0.01$ )	(P) 0.98 ( $\pm 0.01$ )
		(R) 1.00 ( $\pm 0.01$ )	(R) 1.00 ( $\pm 0.00$ )
K-NN	0.68 ( $\pm 0.03$ )	(A) 0.65 ( $\pm 0.02$ )	(A) 0.65 ( $\pm 0.02$ )
		(F) 0.77 ( $\pm 0.02$ )	(F) 0.77 ( $\pm 0.02$ )
		(P) 0.71 ( $\pm 0.01$ )	(P) 0.71 ( $\pm 0.01$ )
		(R) 0.84 ( $\pm 0.04$ )	(R) 0.83 ( $\pm 0.03$ )
LDA	0.75 ( $\pm 0.03$ )	(A) 1.00 ( $\pm 0.00$ )	(A) 1.00 ( $\pm 0.00$ )
		(F) 1.00 ( $\pm 0.00$ )	(F) 1.00 ( $\pm 0.00$ )
		(P) 1.00 ( $\pm 0.00$ )	(P) 1.00 ( $\pm 0.00$ )
		(R) 1.00 ( $\pm 0.00$ )	(R) 1.00 ( $\pm 0.00$ )
LR	1.00 ( $\pm 0.00$ )	(A) 0.99 ( $\pm 0.01$ )	(A) 1.00 ( $\pm 0.00$ )
		(F) 0.99 ( $\pm 0.00$ )	(F) 1.00 ( $\pm 0.00$ )
		(P) 0.99 ( $\pm 0.01$ )	(P) 0.99 ( $\pm 0.01$ )
		(R) 1.00 ( $\pm 0.00$ )	(R) 1.00 ( $\pm 0.00$ )
NB	1.00 ( $\pm 0.00$ )	(A) 0.82 ( $\pm 0.10$ )	(A) 0.87 ( $\pm 0.02$ )
		(F) 0.85 ( $\pm 0.12$ )	(F) 0.90 ( $\pm 0.02$ )
		(P) 0.95 ( $\pm 0.01$ )	(P) 0.96 ( $\pm 0.01$ )
		(R) 0.78 ( $\pm 0.15$ )	(R) 0.85 ( $\pm 0.03$ )
QDA	1.00 ( $\pm 0.00$ )	(A) 0.77 ( $\pm 0.14$ )	(A) 0.86 ( $\pm 0.04$ )
		(F) 0.81 ( $\pm 0.22$ )	(F) 0.90 ( $\pm 0.03$ )
		(P) 0.82 ( $\pm 0.04$ )	(P) 0.87 ( $\pm 0.05$ )
		(R) 0.86 ( $\pm 0.26$ )	(R) 0.95 ( $\pm 0.02$ )

Tabela 5.12: Resultados individuais *wdbc* 80%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.89 ( $\pm 0.02$ )	(A) 0.92 ( $\pm 0.02$ )	(A) 0.59 ( $\pm 0.06$ )
		(F) 0.94 ( $\pm 0.01$ )	(F) 0.67 ( $\pm 0.05$ )
		(P) 0.95 ( $\pm 0.02$ )	(P) 0.68 ( $\pm 0.05$ )
		(R) 0.94 ( $\pm 0.03$ )	(R) 0.66 ( $\pm 0.06$ )
GTB	0.89 ( $\pm 0.03$ )	(A) 0.95 ( $\pm 0.03$ )	(A) 0.68 ( $\pm 0.06$ )
		(F) 0.96 ( $\pm 0.02$ )	(F) 0.77 ( $\pm 0.04$ )
		(P) 0.95 ( $\pm 0.02$ )	(P) 0.70 ( $\pm 0.04$ )
		(R) 0.96 ( $\pm 0.03$ )	(R) 0.84 ( $\pm 0.05$ )
K-NN	0.91 ( $\pm 0.02$ )	(A) 0.93 ( $\pm 0.02$ )	(A) 0.68 ( $\pm 0.06$ )
		(F) 0.94 ( $\pm 0.02$ )	(F) 0.76 ( $\pm 0.05$ )
		(P) 0.93 ( $\pm 0.02$ )	(P) 0.72 ( $\pm 0.04$ )
		(R) 0.95 ( $\pm 0.03$ )	(R) 0.81 ( $\pm 0.05$ )
LDA	0.92 ( $\pm 0.02$ )	(A) 0.96 ( $\pm 0.01$ )	(A) 0.72 ( $\pm 0.06$ )
		(F) 0.97 ( $\pm 0.01$ )	(F) 0.81 ( $\pm 0.04$ )
		(P) 0.94 ( $\pm 0.02$ )	(P) 0.72 ( $\pm 0.04$ )
		(R) 0.99 ( $\pm 0.01$ )	(R) 0.92 ( $\pm 0.05$ )
LR	0.92 ( $\pm 0.01$ )	(A) 0.95 ( $\pm 0.02$ )	(A) 0.72 ( $\pm 0.05$ )
		(F) 0.96 ( $\pm 0.01$ )	(F) 0.81 ( $\pm 0.03$ )
		(P) 0.95 ( $\pm 0.02$ )	(P) 0.70 ( $\pm 0.03$ )
		(R) 0.97 ( $\pm 0.02$ )	(R) 0.97 ( $\pm 0.02$ )
NB	0.93 ( $\pm 0.01$ )	(A) 0.94 ( $\pm 0.03$ )	(A) 0.86 ( $\pm 0.04$ )
		(F) 0.95 ( $\pm 0.02$ )	(F) 0.90 ( $\pm 0.03$ )
		(P) 0.95 ( $\pm 0.02$ )	(P) 0.85 ( $\pm 0.03$ )
		(R) 0.95 ( $\pm 0.03$ )	(R) 0.95 ( $\pm 0.04$ )
QDA	0.85 ( $\pm 0.04$ )	(A) 0.95 ( $\pm 0.01$ )	(A) 0.71 ( $\pm 0.10$ )
		(F) 0.96 ( $\pm 0.01$ )	(F) 0.78 ( $\pm 0.09$ )
		(P) 0.97 ( $\pm 0.02$ )	(P) 0.74 ( $\pm 0.06$ )
		(R) 0.96 ( $\pm 0.02$ )	(R) 0.85 ( $\pm 0.14$ )

Tabela 5.13: Resultados individuais *wdbc* 60%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.90 ( $\pm 0.03$ )	(A) 0.93 ( $\pm 0.02$ )	(A) 0.68 ( $\pm 0.07$ )
		(F) 0.94 ( $\pm 0.01$ )	(F) 0.74 ( $\pm 0.05$ )
		(P) 0.96 ( $\pm 0.02$ )	(P) 0.76 ( $\pm 0.07$ )
		(R) 0.93 ( $\pm 0.03$ )	(R) 0.72 ( $\pm 0.05$ )
GTB	0.94 ( $\pm 0.03$ )	(A) 0.95 ( $\pm 0.03$ )	(A) 0.80 ( $\pm 0.07$ )
		(F) 0.96 ( $\pm 0.02$ )	(F) 0.85 ( $\pm 0.05$ )
		(P) 0.96 ( $\pm 0.02$ )	(P) 0.83 ( $\pm 0.07$ )
		(R) 0.95 ( $\pm 0.04$ )	(R) 0.88 ( $\pm 0.04$ )
K-NN	0.92 ( $\pm 0.02$ )	(A) 0.93 ( $\pm 0.02$ )	(A) 0.79 ( $\pm 0.05$ )
		(F) 0.95 ( $\pm 0.02$ )	(F) 0.84 ( $\pm 0.04$ )
		(P) 0.94 ( $\pm 0.02$ )	(P) 0.80 ( $\pm 0.03$ )
		(R) 0.96 ( $\pm 0.02$ )	(R) 0.88 ( $\pm 0.05$ )
LDA	0.95 ( $\pm 0.01$ )	(A) 0.96 ( $\pm 0.01$ )	(A) 0.84 ( $\pm 0.04$ )
		(F) 0.97 ( $\pm 0.01$ )	(F) 0.88 ( $\pm 0.02$ )
		(P) 0.94 ( $\pm 0.02$ )	(P) 0.82 ( $\pm 0.04$ )
		(R) 0.99 ( $\pm 0.01$ )	(R) 0.96 ( $\pm 0.02$ )
LR	0.94 ( $\pm 0.01$ )	(A) 0.96 ( $\pm 0.01$ )	(A) 0.84 ( $\pm 0.03$ )
		(F) 0.96 ( $\pm 0.01$ )	(F) 0.88 ( $\pm 0.02$ )
		(P) 0.96 ( $\pm 0.02$ )	(P) 0.81 ( $\pm 0.03$ )
		(R) 0.97 ( $\pm 0.01$ )	(R) 0.98 ( $\pm 0.02$ )
NB	0.93 ( $\pm 0.02$ )	(A) 0.94 ( $\pm 0.03$ )	(A) 0.91 ( $\pm 0.04$ )
		(F) 0.95 ( $\pm 0.02$ )	(F) 0.93 ( $\pm 0.03$ )
		(P) 0.95 ( $\pm 0.02$ )	(P) 0.90 ( $\pm 0.03$ )
		(R) 0.95 ( $\pm 0.03$ )	(R) 0.97 ( $\pm 0.03$ )
QDA	0.95 ( $\pm 0.1$ )	(A) 0.96 ( $\pm 0.01$ )	(A) 0.84 ( $\pm 0.04$ )
		(F) 0.97 ( $\pm 0.01$ )	(F) 0.88 ( $\pm 0.03$ )
		(P) 0.98 ( $\pm 0.01$ )	(P) 0.83 ( $\pm 0.04$ )
		(R) 0.96 ( $\pm 0.01$ )	(R) 0.94 ( $\pm 0.04$ )

Tabela 5.14: Resultados individuais *wdbc* 40%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.92 ( $\pm 0.01$ )	(A) 0.92 ( $\pm 0.02$ )	(A) 0.74 ( $\pm 0.05$ )
		(F) 0.94 ( $\pm 0.02$ )	(F) 0.79 ( $\pm 0.04$ )
		(P) 0.96 ( $\pm 0.01$ )	(P) 0.80 ( $\pm 0.04$ )
		(R) 0.92 ( $\pm 0.04$ )	(R) 0.79 ( $\pm 0.05$ )
GTB	0.94 ( $\pm 0.02$ )	(A) 0.95 ( $\pm 0.02$ )	(A) 0.87 ( $\pm 0.03$ )
		(F) 0.96 ( $\pm 0.01$ )	(F) 0.90 ( $\pm 0.02$ )
		(P) 0.96 ( $\pm 0.02$ )	(P) 0.88 ( $\pm 0.04$ )
		(R) 0.96 ( $\pm 0.02$ )	(R) 0.92 ( $\pm 0.02$ )
K-NN	0.92 ( $\pm 0.02$ )	(A) 0.93 ( $\pm 0.02$ )	(A) 0.85 ( $\pm 0.05$ )
		(F) 0.95 ( $\pm 0.01$ )	(F) 0.88 ( $\pm 0.04$ )
		(P) 0.94 ( $\pm 0.03$ )	(P) 0.85 ( $\pm 0.04$ )
		(R) 0.96 ( $\pm 0.02$ )	(R) 0.92 ( $\pm 0.04$ )
LDA	0.96 ( $\pm 0.01$ )	(A) 0.96 ( $\pm 0.02$ )	(A) 0.91 ( $\pm 0.04$ )
		(F) 0.97 ( $\pm 0.01$ )	(F) 0.93 ( $\pm 0.02$ )
		(P) 0.95 ( $\pm 0.02$ )	(P) 0.89 ( $\pm 0.04$ )
		(R) 0.99 ( $\pm 0.01$ )	(R) 0.98 ( $\pm 0.01$ )
LR	0.95 ( $\pm 0.02$ )	(A) 0.96 ( $\pm 0.01$ )	(A) 0.90 ( $\pm 0.13$ )
		(F) 0.96 ( $\pm 0.01$ )	(F) 0.92 ( $\pm 0.02$ )
		(P) 0.96 ( $\pm 0.02$ )	(P) 0.87 ( $\pm 0.04$ )
		(R) 0.97 ( $\pm 0.02$ )	(R) 0.99 ( $\pm 0.01$ )
NB	0.93 ( $\pm 0.03$ )	(A) 0.94 ( $\pm 0.03$ )	(A) 0.93 ( $\pm 0.03$ )
		(F) 0.95 ( $\pm 0.02$ )	(F) 0.94 ( $\pm 0.02$ )
		(P) 0.95 ( $\pm 0.02$ )	(P) 0.92 ( $\pm 0.03$ )
		(R) 0.95 ( $\pm 0.03$ )	(R) 0.97 ( $\pm 0.03$ )
QDA	0.96 ( $\pm 0.01$ )	(A) 0.96 ( $\pm 0.01$ )	(A) 0.90 ( $\pm 0.04$ )
		(F) 0.97 ( $\pm 0.01$ )	(F) 0.92 ( $\pm 0.03$ )
		(P) 0.98 ( $\pm 0.01$ )	(P) 0.89 ( $\pm 0.03$ )
		(R) 0.96 ( $\pm 0.02$ )	(R) 0.95 ( $\pm 0.03$ )

Tabela 5.15: Resultados individuais *wdbc* 20%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.91 ( $\pm 0.03$ )	(A) 0.92 ( $\pm 0.02$ )	(A) 0.84 ( $\pm 0.06$ )
		(F) 0.93 ( $\pm 0.02$ )	(F) 0.87 ( $\pm 0.05$ )
		(P) 0.96 ( $\pm 0.02$ )	(P) 0.89 ( $\pm 0.05$ )
		(R) 0.91 ( $\pm 0.04$ )	(R) 0.85 ( $\pm 0.05$ )
GTB	0.95 ( $\pm 0.02$ )	(A) 0.95 ( $\pm 0.03$ )	(A) 0.93 ( $\pm 0.03$ )
		(F) 0.96 ( $\pm 0.02$ )	(F) 0.95 ( $\pm 0.02$ )
		(P) 0.97 ( $\pm 0.02$ )	(P) 0.94 ( $\pm 0.03$ )
		(R) 0.96 ( $\pm 0.03$ )	(R) 0.96 ( $\pm 0.03$ )
K-NN	0.93 ( $\pm 0.03$ )	(A) 0.93 ( $\pm 0.02$ )	(A) 0.90 ( $\pm 0.05$ )
		(F) 0.95 ( $\pm 0.01$ )	(F) 0.92 ( $\pm 0.04$ )
		(P) 0.94 ( $\pm 0.03$ )	(P) 0.90 ( $\pm 0.05$ )
		(R) 0.96 ( $\pm 0.03$ )	(R) 0.95 ( $\pm 0.04$ )
LDA	0.96 ( $\pm 0.02$ )	(A) 0.96 ( $\pm 0.01$ )	(A) 0.93 ( $\pm 0.03$ )
		(F) 0.97 ( $\pm 0.01$ )	(F) 0.95 ( $\pm 0.02$ )
		(P) 0.95 ( $\pm 0.02$ )	(P) 0.92 ( $\pm 0.03$ )
		(R) 0.99 ( $\pm 0.01$ )	(R) 0.99 ( $\pm 0.01$ )
LR	0.95 ( $\pm 0.02$ )	(A) 0.95 ( $\pm 0.01$ )	(A) 0.93 ( $\pm 0.04$ )
		(F) 0.96 ( $\pm 0.01$ )	(F) 0.95 ( $\pm 0.03$ )
		(P) 0.96 ( $\pm 0.02$ )	(P) 0.91 ( $\pm 0.05$ )
		(R) 0.97 ( $\pm 0.02$ )	(R) 0.99 ( $\pm 0.01$ )
NB	0.93 ( $\pm 0.04$ )	(A) 0.94 ( $\pm 0.03$ )	(A) 0.93 ( $\pm 0.02$ )
		(F) 0.95 ( $\pm 0.02$ )	(F) 0.95 ( $\pm 0.02$ )
		(P) 0.95 ( $\pm 0.02$ )	(P) 0.93 ( $\pm 0.02$ )
		(R) 0.95 ( $\pm 0.03$ )	(R) 0.96 ( $\pm 0.02$ )
QDA	0.95 ( $\pm 0.02$ )	(A) 0.96 ( $\pm 0.02$ )	(A) 0.93 ( $\pm 0.03$ )
		(F) 0.97 ( $\pm 0.01$ )	(F) 0.94 ( $\pm 0.02$ )
		(P) 0.98 ( $\pm 0.01$ )	(P) 0.93 ( $\pm 0.02$ )
		(R) 0.96 ( $\pm 0.02$ )	(R) 0.95 ( $\pm 0.03$ )

Porém, no conjunto de dados *kr-vs-kp* (3196 instâncias; 52,2% positivas, 47,8% negativas) os resultados do social-training foram muito inferiores ao do tri-training (Tabelas 5.16 a

5.19). Uma análise inicial indica que utilizar agregação de rankings em casos onde a taxa entre instâncias positivas e negativas é similar, caso os algoritmos tenham muita variação de rankings isto pode acarretar em alto nível de ruído nos resultados. Ou seja, o grupo de instâncias que possuía uma confiança baixa sobre seu rótulo, e, portanto, estava mais próximo ao parâmetro  $m$ , acabou se mostrando problemático nesse conjunto de dados.

Tabela 5.16: Resultados individuais *kr-vs-kp* 80%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.89 ( $\pm 0.18$ )	(A) 0.42 ( $\pm 0.15$ )	(A) 0.60 ( $\pm 0.23$ )
		(F) 0.38 ( $\pm 0.18$ )	(F) 0.57 ( $\pm 0.23$ )
		(P) 0.43 ( $\pm 0.18$ )	(P) 0.67 ( $\pm 0.26$ )
		(R) 0.36 ( $\pm 0.20$ )	(R) 0.52 ( $\pm 0.25$ )
GTB	0.85 ( $\pm 0.18$ )	(A) 0.40 ( $\pm 0.18$ )	(A) 0.61 ( $\pm 0.23$ )
		(F) 0.34 ( $\pm 0.23$ )	(F) 0.58 ( $\pm 0.26$ )
		(P) 0.37 ( $\pm 0.20$ )	(P) 0.66 ( $\pm 0.26$ )
		(R) 0.34 ( $\pm 0.28$ )	(R) 0.56 ( $\pm 0.29$ )
K-NN	0.66 ( $\pm 0.09$ )	(A) 0.47 ( $\pm 0.08$ )	(A) 0.59 ( $\pm 0.11$ )
		(F) 0.49 ( $\pm 0.11$ )	(F) 0.59 ( $\pm 0.15$ )
		(P) 0.49 ( $\pm 0.07$ )	(P) 0.60 ( $\pm 0.11$ )
		(R) 0.50 ( $\pm 0.16$ )	(R) 0.60 ( $\pm 0.21$ )
LDA	0.59 ( $\pm 0.08$ )	(A) 0.31 ( $\pm 0.22$ )	(A) 0.61 ( $\pm 0.25$ )
		(F) 0.30 ( $\pm 0.27$ )	(F) 0.60 ( $\pm 0.26$ )
		(P) 0.29 ( $\pm 0.23$ )	(P) 0.64 ( $\pm 0.25$ )
		(R) 0.34 ( $\pm 0.34$ )	(R) 0.60 ( $\pm 0.29$ )
LR	0.82 ( $\pm 0.12$ )	(A) 0.33 ( $\pm 0.20$ )	(A) 0.61 ( $\pm 0.22$ )
		(F) 0.33 ( $\pm 0.25$ )	(F) 0.61 ( $\pm 0.22$ )
		(P) 0.31 ( $\pm 0.21$ )	(P) 0.65 ( $\pm 0.24$ )
		(R) 0.36 ( $\pm 0.32$ )	(R) 0.62 ( $\pm 0.27$ )
NB	0.72 ( $\pm 0.12$ )	(A) 0.50 ( $\pm 0.05$ )	(A) 0.53 ( $\pm 0.05$ )
		(F) 0.38 ( $\pm 0.27$ )	(F) 0.48 ( $\pm 0.26$ )
		(P) 0.50 ( $\pm 0.22$ )	(P) 0.49 ( $\pm 0.18$ )
		(R) 0.46 ( $\pm 0.43$ )	(R) 0.58 ( $\pm 0.41$ )
QDA	0.69 ( $\pm 0.12$ )	(A) 0.49 ( $\pm 0.05$ )	(A) 0.55 ( $\pm 0.11$ )
		(F) 0.40 ( $\pm 0.18$ )	(F) 0.52 ( $\pm 0.18$ )
		(P) 0.49 ( $\pm 0.09$ )	(P) 0.56 ( $\pm 0.09$ )
		(R) 0.38 ( $\pm 0.24$ )	(R) 0.53 ( $\pm 0.27$ )

Tabela 5.17: Resultados individuais *kr-vs-kp* 60%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	1.00 ( $\pm 0.00$ )	(A) 0.53 ( $\pm 0.19$ )	(A) 0.70 ( $\pm 0.25$ )
		(F) 0.44 ( $\pm 0.22$ )	(F) 0.69 ( $\pm 0.25$ )
		(P) 0.61 ( $\pm 0.27$ )	(P) 0.76 ( $\pm 0.27$ )
		(R) 0.36 ( $\pm 0.20$ )	(R) 0.65 ( $\pm 0.24$ )
GTB	1.00 ( $\pm 0.00$ )	(A) 0.51 ( $\pm 0.19$ )	(A) 0.73 ( $\pm 0.25$ )
		(F) 0.38 ( $\pm 0.25$ )	(F) 0.72 ( $\pm 0.25$ )
		(P) 0.54 ( $\pm 0.35$ )	(P) 0.78 ( $\pm 0.26$ )
		(R) 0.31 ( $\pm 0.21$ )	(R) 0.68 ( $\pm 0.25$ )
K-NN	0.70 ( $\pm 0.08$ )	(A) 0.54 ( $\pm 0.11$ )	(A) 0.67 ( $\pm 0.16$ )
		(F) 0.54 ( $\pm 0.12$ )	(F) 0.67 ( $\pm 0.18$ )
		(P) 0.57 ( $\pm 0.11$ )	(P) 0.69 ( $\pm 0.15$ )
		(R) 0.52 ( $\pm 0.14$ )	(R) 0.66 ( $\pm 0.21$ )
LDA	0.67 ( $\pm 0.10$ )	(A) 0.49 ( $\pm 0.16$ )	(A) 0.80 ( $\pm 0.24$ )
		(F) 0.42 ( $\pm 0.22$ )	(F) 0.80 ( $\pm 0.23$ )
		(P) 0.49 ( $\pm 0.24$ )	(P) 0.82 ( $\pm 0.24$ )
		(R) 0.39 ( $\pm 0.23$ )	(R) 0.80 ( $\pm 0.24$ )
LR	0.96 ( $\pm 0.04$ )	(A) 0.48 ( $\pm 0.14$ )	(A) 0.79 ( $\pm 0.21$ )
		(F) 0.42 ( $\pm 0.20$ )	(F) 0.80 ( $\pm 0.19$ )
		(P) 0.49 ( $\pm 0.20$ )	(P) 0.82 ( $\pm 0.21$ )
		(R) 0.39 ( $\pm 0.22$ )	(R) 0.78 ( $\pm 0.17$ )
NB	0.87 ( $\pm 0.15$ )	(A) 0.48 ( $\pm 0.07$ )	(A) 0.59 ( $\pm 0.11$ )
		(F) 0.43 ( $\pm 0.27$ )	(F) 0.62 ( $\pm 0.22$ )
		(P) 0.45 ( $\pm 0.18$ )	(P) 0.62 ( $\pm 0.12$ )
		(R) 0.55 ( $\pm 0.44$ )	(R) 0.76 ( $\pm 0.33$ )
QDA	0.83 ( $\pm 0.17$ )	(A) 0.51 ( $\pm 0.08$ )	(A) 0.55 ( $\pm 0.09$ )
		(F) 0.51 ( $\pm 0.13$ )	(F) 0.61 ( $\pm 0.15$ )
		(P) 0.53 ( $\pm 0.07$ )	(P) 0.55 ( $\pm 0.06$ )
		(R) 0.54 ( $\pm 0.22$ )	(R) 0.74 ( $\pm 0.24$ )

Tabela 5.18: Resultados individuais *kr-vs-kp* 40%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	1.00 ( $\pm 0.00$ )	(A) 0.71 ( $\pm 0.15$ )	(A) 0.82 ( $\pm 0.14$ )
		(F) 0.71 ( $\pm 0.13$ )	(F) 0.82 ( $\pm 0.14$ )
		(P) 0.78 ( $\pm 0.18$ )	(P) 0.86 ( $\pm 0.15$ )
		(R) 0.67 ( $\pm 0.14$ )	(R) 0.80 ( $\pm 0.17$ )
GTB	1.00 ( $\pm 0.00$ )	(A) 0.70 ( $\pm 0.15$ )	(A) 0.86 ( $\pm 0.13$ )
		(F) 0.72 ( $\pm 0.11$ )	(F) 0.87 ( $\pm 0.12$ )
		(P) 0.74 ( $\pm 0.16$ )	(P) 0.87 ( $\pm 0.14$ )
		(R) 0.71 ( $\pm 0.10$ )	(R) 0.88 ( $\pm 0.12$ )
K-NN	0.78 ( $\pm 0.09$ )	(A) 0.64 ( $\pm 0.10$ )	(A) 0.74 ( $\pm 0.10$ )
		(F) 0.68 ( $\pm 0.08$ )	(F) 0.76 ( $\pm 0.09$ )
		(P) 0.65 ( $\pm 0.10$ )	(P) 0.75 ( $\pm 0.11$ )
		(R) 0.71 ( $\pm 0.09$ )	(R) 0.77 ( $\pm 0.10$ )
LDA	0.75 ( $\pm 0.12$ )	(A) 0.69 ( $\pm 0.15$ )	(A) 0.88 ( $\pm 0.13$ )
		(F) 0.74 ( $\pm 0.13$ )	(F) 0.90 ( $\pm 0.10$ )
		(P) 0.69 ( $\pm 0.15$ )	(P) 0.88 ( $\pm 0.15$ )
		(R) 0.82 ( $\pm 0.16$ )	(R) 0.93 ( $\pm 0.08$ )
LR	1.00 ( $\pm 0.01$ )	(A) 0.67 ( $\pm 0.16$ )	(A) 0.86 ( $\pm 0.13$ )
		(F) 0.72 ( $\pm 0.13$ )	(F) 0.87 ( $\pm 0.11$ )
		(P) 0.68 ( $\pm 0.16$ )	(P) 0.86 ( $\pm 0.15$ )
		(R) 0.79 ( $\pm 0.16$ )	(R) 0.91 ( $\pm 0.10$ )
NB	0.99 ( $\pm 0.01$ )	(A) 0.49 ( $\pm 0.06$ )	(A) 0.54 ( $\pm 0.05$ )
		(F) 0.49 ( $\pm 0.25$ )	(F) 0.65 ( $\pm 0.08$ )
		(P) 0.52 ( $\pm 0.18$ )	(P) 0.54 ( $\pm 0.03$ )
		(R) 0.64 ( $\pm 0.42$ )	(R) 0.87 ( $\pm 0.20$ )
QDA	0.97 ( $\pm 0.05$ )	(A) 0.53 ( $\pm 0.07$ )	(A) 0.65 ( $\pm 0.10$ )
		(F) 0.54 ( $\pm 0.17$ )	(F) 0.69 ( $\pm 0.12$ )
		(P) 0.54 ( $\pm 0.08$ )	(P) 0.65 ( $\pm 0.12$ )
		(R) 0.62 ( $\pm 0.27$ )	(R) 0.80 ( $\pm 0.22$ )

Tabela 5.19: Resultados individuais *kr-vs-kp* 20%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	1.00 ( $\pm 0.00$ )	(A) 0.77 ( $\pm 0.08$ )	(A) 0.90 ( $\pm 0.05$ )
		(F) 0.77 ( $\pm 0.09$ )	(F) 0.91 ( $\pm 0.04$ )
		(P) 0.78 ( $\pm 0.08$ )	(P) 0.89 ( $\pm 0.07$ )
		(R) 0.80 ( $\pm 0.18$ )	(R) 0.93 ( $\pm 0.03$ )
GTB	1.00 ( $\pm 0.00$ )	(A) 0.80 ( $\pm 0.10$ )	(A) 0.95 ( $\pm 0.05$ )
		(F) 0.81 ( $\pm 0.10$ )	(F) 0.95 ( $\pm 0.04$ )
		(P) 0.80 ( $\pm 0.09$ )	(P) 0.95 ( $\pm 0.08$ )
		(R) 0.84 ( $\pm 0.15$ )	(R) 0.96 ( $\pm 0.04$ )
K-NN	0.84 ( $\pm 0.08$ )	(A) 0.74 ( $\pm 0.07$ )	(A) 0.81 ( $\pm 0.05$ )
		(F) 0.76 ( $\pm 0.07$ )	(F) 0.82 ( $\pm 0.04$ )
		(P) 0.74 ( $\pm 0.07$ )	(P) 0.81 ( $\pm 0.06$ )
		(R) 0.79 ( $\pm 0.10$ )	(R) 0.85 ( $\pm 0.05$ )
LDA	0.78 ( $\pm 0.11$ )	(A) 0.87 ( $\pm 0.10$ )	(A) 0.99 ( $\pm 0.03$ )
		(F) 0.87 ( $\pm 0.11$ )	(F) 0.99 ( $\pm 0.03$ )
		(P) 0.85 ( $\pm 0.09$ )	(P) 0.98 ( $\pm 0.05$ )
		(R) 0.91 ( $\pm 0.15$ )	(R) 1.00 ( $\pm 0.01$ )
LR	1.00 ( $\pm 0.00$ )	(A) 0.79 ( $\pm 0.15$ )	(A) 0.94 ( $\pm 0.05$ )
		(F) 0.80 ( $\pm 0.14$ )	(F) 0.94 ( $\pm 0.04$ )
		(P) 0.77 ( $\pm 0.14$ )	(P) 0.95 ( $\pm 0.07$ )
		(R) 0.85 ( $\pm 0.21$ )	(R) 0.95 ( $\pm 0.05$ )
NB	1.00 ( $\pm 0.01$ )	(A) 0.55 ( $\pm 0.09$ )	(A) 0.62 ( $\pm 0.11$ )
		(F) 0.59 ( $\pm 0.21$ )	(F) 0.72 ( $\pm 0.05$ )
		(P) 0.53 ( $\pm 0.14$ )	(P) 0.61 ( $\pm 0.12$ )
		(R) 0.75 ( $\pm 0.33$ )	(R) 0.91 ( $\pm 0.10$ )
QDA	1.00 ( $\pm 0.01$ )	(A) 0.59 ( $\pm 0.07$ )	(A) 0.67 ( $\pm 0.07$ )
		(F) 0.66 ( $\pm 0.08$ )	(F) 0.73 ( $\pm 0.05$ )
		(P) 0.59 ( $\pm 0.06$ )	(P) 0.63 ( $\pm 0.05$ )
		(R) 0.78 ( $\pm 0.17$ )	(R) 0.88 ( $\pm 0.09$ )

#### 5.4 Comparação com Democratic Co-Learning

A fim de obter uma comparação do social-training com um outro tipo de abordagem similar, testes utilizando alguns dos conjuntos de dados utilizados pelo trabalho democratic co-learning (DCL) (ZHOU; GOLDMAN, 2004) foram executados. Este trabalho foi escolhido por sua similaridade com a abordagem proposta nesta dissertação (conforme discutido no Capítulo 3) e por conter conjuntos de dados disponíveis no repositório UCI. A Tabela 5.20 apresenta as características dos mesmos. Entretanto, assim como na comparação com o tri-training, alguns dos conjuntos de dados apresentados no trabalho DCL não puderam ser utilizados. Os seguintes conjuntos de dados não foram encontrados com o nome informado no trabalho: *monk2*, *DNA*, *3-of-9* e *xd6*. Além disso, no caso de um dos conjuntos de dados utilizados nesse trabalho (*flare*), o mesmo se trata de um caso de classificação multiclasse, cenário que não é abordado na solução proposta nesta dissertação.

Tabela 5.20 – Características dos conjuntos de dados utilizados pelo DCL

Conjunto de dados	Número de atributos	Número de instâncias
adult	14	32561
cancer	9	286
vote	16	435

Um dos pontos que é necessário levantar sobre esta comparação é que o DCL não trabalha com uma taxa de dados não rotulados baseada em uma porcentagem e sim utiliza um tamanho arbitrário para o conjunto  $L$ . Este valor é apresentado, e o valor obtido através do uso de uma taxa por parte do social-training é apresentado igualmente. Neste caso, é importante ressaltar que, devido ao tamanho do conjunto de dados *adult*, foi necessário criar uma nova taxa de dados não rotulados especificamente para esta comparação, no caso a taxa de 95%. Mesmo assim o tamanho de  $L$  no caso do social-training é muito maior que no caso do DCL. Isto ocorre pois o DCL não respeita a taxa de distribuição entre rótulos do conjunto de dados original ao criar  $L$  e  $U$ . Outro problema que agravou isto, foi que este conjunto de dados em específico possui uma grande discrepância entre seus rótulos (99,9% são da classe positiva). Estes dados e a comparação dos resultados são apresentados na Tabela 5.21. Novamente, o melhor resultado para cada linha é apresentado em **negrito**.

Os resultados da Tabela 5.21 apontam um desempenho superior em dois dos três conjuntos de dados, sendo que o conjunto de dados no qual o desempenho foi inferior é o conjunto

Tabela 5.21 – Comparação com Democratic Co-Learning (DCL). (AER  $\pm$  desvio padrão)

Conjunto de dados	L		DCL	Social-Training	
	DCL	Social-Training		Borda	Copeland
adult	60	1223	0.216 ( $\pm 0.021$ )	<b>0.157</b> ( $\pm 0.076$ )	0.199 ( $\pm 0.066$ )
cancer	40	45	0.297 ( $\pm 0.026$ )	0.183 ( $\pm 0.055$ )	<b>0.149</b> ( $\pm 0.026$ )
vote	40	67	<b>0.056</b> ( $\pm 0.012$ )	0.140 ( $\pm 0.060$ )	0.397 ( $\pm 0.056$ )

*vote*, o qual também estava na comparação com o tri-training e no qual o social-training também teve um desempenho inferior nesse caso (Tabela 5.3).

A razão para o mau desempenho neste conjunto de dados em específico pode ser visto em detalhes nas Tabelas 5.22 a 5.25, onde são apresentados os resultados individuais de cada algoritmo. Se trata de um caso em que alguns algoritmos de aprendizado possuem uma acurácia muito inferior aos demais e geram ruído na etapa de agregação de rankings.

Tabela 5.22: Resultados individuais *vote* 80%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.98 ( $\pm 0.03$ )	(A) 0.84 ( $\pm 0.06$ )	(A) 0.57 ( $\pm 0.06$ )
		(F) 0.83 ( $\pm 0.07$ )	(F) 0.54 ( $\pm 0.08$ )
		(P) 0.97 ( $\pm 0.04$ )	(P) 0.64 ( $\pm 0.06$ )
		(R) 0.73 ( $\pm 0.10$ )	(R) 0.47 ( $\pm 0.10$ )
GTB	0.99 ( $\pm 0.01$ )	(A) 0.87 ( $\pm 0.06$ )	(A) 0.64 ( $\pm 0.09$ )
		(F) 0.86 ( $\pm 0.08$ )	(F) 0.60 ( $\pm 0.12$ )
		(P) 0.98 ( $\pm 0.02$ )	(P) 0.72 ( $\pm 0.09$ )
		(R) 0.76 ( $\pm 0.12$ )	(R) 0.53 ( $\pm 0.15$ )
K-NN	0.76 ( $\pm 0.04$ )	(A) 0.74 ( $\pm 0.07$ )	(A) 0.58 ( $\pm 0.06$ )
		(F) 0.73 ( $\pm 0.08$ )	(F) 0.56 ( $\pm 0.08$ )
		(P) 0.82 ( $\pm 0.06$ )	(P) 0.65 ( $\pm 0.06$ )
		(R) 0.66 ( $\pm 0.11$ )	(R) 0.50 ( $\pm 0.10$ )
LDA	0.80 ( $\pm 0.12$ )	(A) 0.92 ( $\pm 0.06$ )	(A) 0.66 ( $\pm 0.09$ )
		(F) 0.92 ( $\pm 0.07$ )	(F) 0.62 ( $\pm 0.11$ )
		(P) 0.97 ( $\pm 0.04$ )	(P) 0.78 ( $\pm 0.07$ )
		(R) 0.88 ( $\pm 0.09$ )	(R) 0.52 ( $\pm 0.13$ )
LR	0.93 ( $\pm 0.06$ )	(A) 0.86 ( $\pm 0.06$ )	(A) 0.65 ( $\pm 0.08$ )
		(F) 0.85 ( $\pm 0.08$ )	(F) 0.59 ( $\pm 0.12$ )
		(P) 0.98 ( $\pm 0.02$ )	(P) 0.78 ( $\pm 0.08$ )
		(R) 0.76 ( $\pm 0.13$ )	(R) 0.49 ( $\pm 0.13$ )
NB	0.88 ( $\pm 0.06$ )	(A) 0.88 ( $\pm 0.09$ )	(A) 0.56 ( $\pm 0.11$ )
		(F) 0.87 ( $\pm 0.11$ )	(F) 0.57 ( $\pm 0.16$ )
		(P) 0.97 ( $\pm 0.05$ )	(P) 0.60 ( $\pm 0.11$ )
		(R) 0.80 ( $\pm 0.15$ )	(R) 0.60 ( $\pm 0.24$ )
QDA	0.92 ( $\pm 0.04$ )	(A) 0.92 ( $\pm 0.10$ )	(A) 0.56 ( $\pm 0.03$ )
		(F) 0.92 ( $\pm 0.09$ )	(F) 0.66 ( $\pm 0.05$ )
		(P) 0.96 ( $\pm 0.10$ )	(P) 0.57 ( $\pm 0.02$ )
		(R) 0.89 ( $\pm 0.08$ )	(R) 0.80 ( $\pm 0.14$ )

Tabela 5.23: Resultados individuais *vote* 60%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.99 ( $\pm 0.02$ )	(A) 0.88 ( $\pm 0.05$ )	(A) 0.66 ( $\pm 0.03$ )
		(F) 0.88 ( $\pm 0.06$ )	(F) 0.64 ( $\pm 0.03$ )
		(P) 0.99 ( $\pm 0.02$ )	(P) 0.74 ( $\pm 0.07$ )
		(R) 0.79 ( $\pm 0.09$ )	(R) 0.57 ( $\pm 0.06$ )
GTB	0.99 ( $\pm 0.01$ )	(A) 0.91 ( $\pm 0.03$ )	(A) 0.81 ( $\pm 0.06$ )
		(F) 0.91 ( $\pm 0.03$ )	(F) 0.80 ( $\pm 0.08$ )
		(P) 0.99 ( $\pm 0.02$ )	(P) 0.88 ( $\pm 0.03$ )
		(R) 0.84 ( $\pm 0.07$ )	(R) 0.75 ( $\pm 0.12$ )
K-NN	0.76 ( $\pm 0.05$ )	(A) 0.75 ( $\pm 0.08$ )	(A) 0.69 ( $\pm 0.06$ )
		(F) 0.75 ( $\pm 0.10$ )	(F) 0.69 ( $\pm 0.07$ )
		(P) 0.81 ( $\pm 0.07$ )	(P) 0.75 ( $\pm 0.05$ )
		(R) 0.71 ( $\pm 0.14$ )	(R) 0.64 ( $\pm 0.09$ )
LDA	0.91 ( $\pm 0.10$ )	(A) 0.98 ( $\pm 0.01$ )	(A) 0.83 ( $\pm 0.08$ )
		(F) 0.98 ( $\pm 0.01$ )	(F) 0.83 ( $\pm 0.09$ )
		(P) 0.99 ( $\pm 0.02$ )	(P) 0.91 ( $\pm 0.05$ )
		(R) 0.98 ( $\pm 0.02$ )	(R) 0.77 ( $\pm 0.12$ )
LR	0.98 ( $\pm 0.01$ )	(A) 0.91 ( $\pm 0.04$ )	(A) 0.82 ( $\pm 0.08$ )
		(F) 0.91 ( $\pm 0.05$ )	(F) 0.81 ( $\pm 0.10$ )
		(P) 0.99 ( $\pm 0.02$ )	(P) 0.91 ( $\pm 0.04$ )
		(R) 0.85 ( $\pm 0.09$ )	(R) 0.73 ( $\pm 0.13$ )
NB	0.94 ( $\pm 0.05$ )	(A) 0.90 ( $\pm 0.07$ )	(A) 0.65 ( $\pm 0.13$ )
		(F) 0.90 ( $\pm 0.08$ )	(F) 0.67 ( $\pm 0.14$ )
		(P) 1.00 ( $\pm 0.01$ )	(P) 0.68 ( $\pm 0.11$ )
		(R) 0.83 ( $\pm 0.12$ )	(R) 0.69 ( $\pm 0.20$ )
QDA	0.95 ( $\pm 0.02$ )	(A) 0.96 ( $\pm 0.03$ )	(A) 0.56 ( $\pm 0.05$ )
		(F) 0.96 ( $\pm 0.03$ )	(F) 0.63 ( $\pm 0.11$ )
		(P) 0.99 ( $\pm 0.02$ )	(P) 0.58 ( $\pm 0.05$ )
		(R) 0.92 ( $\pm 0.04$ )	(R) 0.74 ( $\pm 0.21$ )

Tabela 5.24: Resultados individuais *vote* 40%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	1.00 ( $\pm 0.00$ )	(A) 0.90 ( $\pm 0.05$ )	(A) 0.78 ( $\pm 0.07$ )
		(F) 0.89 ( $\pm 0.05$ )	(F) 0.78 ( $\pm 0.07$ )
		(P) 1.00 ( $\pm 0.00$ )	(P) 0.86 ( $\pm 0.06$ )
		(R) 0.81 ( $\pm 0.09$ )	(R) 0.72 ( $\pm 0.10$ )
GTB	1.00 ( $\pm 0.00$ )	(A) 0.94 ( $\pm 0.04$ )	(A) 0.90 ( $\pm 0.05$ )
		(F) 0.94 ( $\pm 0.04$ )	(F) 0.90 ( $\pm 0.05$ )
		(P) 1.00 ( $\pm 0.01$ )	(P) 0.95 ( $\pm 0.03$ )
		(R) 0.89 ( $\pm 0.07$ )	(R) 0.86 ( $\pm 0.08$ )
K-NN	0.77 ( $\pm 0.05$ )	(A) 0.77 ( $\pm 0.07$ )	(A) 0.75 ( $\pm 0.05$ )
		(F) 0.77 ( $\pm 0.08$ )	(F) 0.77 ( $\pm 0.06$ )
		(P) 0.81 ( $\pm 0.05$ )	(P) 0.78 ( $\pm 0.05$ )
		(R) 0.74 ( $\pm 0.12$ )	(R) 0.76 ( $\pm 0.11$ )
LDA	0.96 ( $\pm 0.02$ )	(A) 1.00 ( $\pm 0.00$ )	(A) 0.95 ( $\pm 0.06$ )
		(F) 1.00 ( $\pm 0.00$ )	(F) 0.95 ( $\pm 0.06$ )
		(P) 1.00 ( $\pm 0.01$ )	(P) 0.97 ( $\pm 0.03$ )
		(R) 1.00 ( $\pm 0.01$ )	(R) 0.94 ( $\pm 0.09$ )
LR	0.98 ( $\pm 0.01$ )	(A) 0.94 ( $\pm 0.03$ )	(A) 0.92 ( $\pm 0.07$ )
		(F) 0.94 ( $\pm 0.04$ )	(F) 0.92 ( $\pm 0.08$ )
		(P) 0.98 ( $\pm 0.02$ )	(P) 0.97 ( $\pm 0.02$ )
		(R) 0.91 ( $\pm 0.07$ )	(R) 0.88 ( $\pm 0.12$ )
NB	0.95 ( $\pm 0.04$ )	(A) 0.92 ( $\pm 0.05$ )	(A) 0.69 ( $\pm 0.09$ )
		(F) 0.92 ( $\pm 0.05$ )	(F) 0.72 ( $\pm 0.08$ )
		(P) 1.00 ( $\pm 0.00$ )	(P) 0.74 ( $\pm 0.12$ )
		(R) 0.86 ( $\pm 0.09$ )	(R) 0.74 ( $\pm 0.15$ )
QDA	0.99 ( $\pm 0.01$ )	(A) 0.93 ( $\pm 0.05$ )	(A) 0.61 ( $\pm 0.05$ )
		(F) 0.93 ( $\pm 0.05$ )	(F) 0.65 ( $\pm 0.10$ )
		(P) 1.00 ( $\pm 0.00$ )	(P) 0.64 ( $\pm 0.07$ )
		(R) 0.88 ( $\pm 0.09$ )	(R) 0.71 ( $\pm 0.22$ )

Tabela 5.25: Resultados individuais *vote 20%*. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	1.00 ( $\pm 0.00$ )	(A) 0.95 ( $\pm 0.03$ )	(A) 0.90 ( $\pm 0.06$ )
		(F) 0.95 ( $\pm 0.03$ )	(F) 0.90 ( $\pm 0.05$ )
		(P) 1.00 ( $\pm 0.00$ )	(P) 0.93 ( $\pm 0.05$ )
		(R) 0.90 ( $\pm 0.06$ )	(R) 0.88 ( $\pm 0.07$ )
GTB	1.00 ( $\pm 0.00$ )	(A) 0.98 ( $\pm 0.02$ )	(A) 0.96 ( $\pm 0.02$ )
		(F) 0.98 ( $\pm 0.02$ )	(F) 0.96 ( $\pm 0.02$ )
		(P) 1.00 ( $\pm 0.00$ )	(P) 0.98 ( $\pm 0.02$ )
		(R) 0.96 ( $\pm 0.04$ )	(R) 0.95 ( $\pm 0.03$ )
K-NN	0.78 ( $\pm 0.07$ )	(A) 0.78 ( $\pm 0.06$ )	(A) 0.79 ( $\pm 0.05$ )
		(F) 0.79 ( $\pm 0.06$ )	(F) 0.81 ( $\pm 0.04$ )
		(P) 0.80 ( $\pm 0.05$ )	(P) 0.81 ( $\pm 0.06$ )
		(R) 0.79 ( $\pm 0.10$ )	(R) 0.82 ( $\pm 0.08$ )
LDA	0.97 ( $\pm 0.02$ )	(A) 1.00 ( $\pm 0.00$ )	(A) 1.00 ( $\pm 0.01$ )
		(F) 1.00 ( $\pm 0.00$ )	(F) 1.00 ( $\pm 0.01$ )
		(P) 1.00 ( $\pm 0.00$ )	(P) 1.00 ( $\pm 0.01$ )
		(R) 1.00 ( $\pm 0.00$ )	(R) 1.00 ( $\pm 0.01$ )
LR	0.99 ( $\pm 0.01$ )	(A) 0.98 ( $\pm 0.01$ )	(A) 0.98 ( $\pm 0.01$ )
		(F) 0.98 ( $\pm 0.01$ )	(F) 0.98 ( $\pm 0.01$ )
		(P) 0.98 ( $\pm 0.02$ )	(P) 0.98 ( $\pm 0.02$ )
		(R) 0.98 ( $\pm 0.03$ )	(R) 0.98 ( $\pm 0.02$ )
NB	0.96 ( $\pm 0.02$ )	(A) 0.93 ( $\pm 0.04$ )	(A) 0.76 ( $\pm 0.08$ )
		(F) 0.93 ( $\pm 0.05$ )	(F) 0.78 ( $\pm 0.05$ )
		(P) 1.00 ( $\pm 0.00$ )	(P) 0.80 ( $\pm 0.11$ )
		(R) 0.88 ( $\pm 0.08$ )	(R) 0.78 ( $\pm 0.10$ )
QDA	1.00 ( $\pm 0.01$ )	(A) 0.94 ( $\pm 0.03$ )	(A) 0.61 ( $\pm 0.08$ )
		(F) 0.95 ( $\pm 0.03$ )	(F) 0.66 ( $\pm 0.15$ )
		(P) 1.00 ( $\pm 0.00$ )	(P) 0.61 ( $\pm 0.06$ )
		(R) 0.90 ( $\pm 0.06$ )	(R) 0.78 ( $\pm 0.25$ )

Por exemplo, utilizando a função Copeland na taxa de dados não rotulados de 80%, o algoritmo K-NN teve uma acurácia de 76% quando classificando  $U$ , enquanto algoritmos como

C4.5 e GTB obtiveram uma acurácia de 99% (Tabela 5.22). Além disso, as métricas de precisão e *recall* na Tabela 5.25 demonstram que o algoritmo LDA obteve 100% de aproveitamento em ambas utilizando a função Borda, enquanto o algoritmo K-NN obteve apenas 80% e 79%, respectivamente.

Este tipo de situação, onde alguns rankings possuem resultados incorretos, é um problema conhecido de sistemas de agregação de rankings e mais detalhes sobre este tipo de cenário são apresentados na Seção 5.5.

## 5.5 Testes para prova de conceito

Para validar o uso de múltiplos algoritmos de aprendizado ao invés de apenas utilizar aquele com os melhores resultados, testes específicos do social-training foram realizados no início da pesquisa. Estes testes foram realizados utilizando apenas a proporção de dados não rotulados de 80%, por se tratar do cenário mais desafiador e com maior grau de interação entre os múltiplos rankings. Além disso, nessa taxa de dados os algoritmos são treinados com um número menor de dados rotulados ( $L$ ). Adicionalmente, isto permite avaliar o desempenho de cada classificador quanto à criação de rankings sobre  $U$ , já que é necessário criar maiores rankings de instâncias. O objetivo deste teste foi verificar se o uso de múltiplos algoritmos de aprendizado poderia produzir resultados melhores, quando comparado com os algoritmos de aprendizado com os melhores resultados dentre estes testes. O critério utilizado para definir quais algoritmos de aprendizado possuem o melhor desempenho foi a acurácia apresentada por cada classificador quando treinado utilizando apenas os dados de  $L$  (Apêndice A). Desta maneira podemos verificar se a *ensemble* de classificadores, através de seus rankings, contribuiu de forma positiva para a criação de  $L'$ , ou, inversamente, se o desempenho dos melhores classificadores estava sendo comprometida pelos piores. Dado seu desempenho superior nos testes iniciais, para este teste apenas a SCF Borda foi utilizada.

Os resultados para estes testes são apresentados na Tabela 5.26. São apresentados os resultados para todos os sete algoritmos de aprendizado (LDA, QDA, GTB, LR, KNN, NB, C4.5), para um grupo contendo os 3 melhores algoritmos, e para o melhor algoritmo (varia a cada caso). Os conjuntos de dados utilizados foram um subconjunto dos utilizados pelo tri-training. A métrica utilizada é a taxa de erro médio (AER) (Equação 4.1). Novamente, os melhores resultados para cada linha são definidos em **negrito**.

Tabela 5.26 – Comparações entre grupos de classificadores (AER (Algoritmos de aprendizado))

Conjunto de dados	Todos sete algoritmos	Três algoritmos com os melhores desempenhos	Melhor desempenho
bupa	0.450	<b>0.408</b> (KNN, LR, GTB)	0.444 (KNN)
diabetes	<b>0.260</b>	0.261 (LDA, NB, LR)	0.268 (LDA)
ionosphere	<b>0.186</b>	0.193 (GTB, C4.5, LR)	0.208 (GTB)
wdbc	0.060	<b>0.052</b> (NB, LDA, LR)	0.074 (NB)

Estes testes apresentam resultados interessantes para casos com múltiplos classificadores. Nos experimentos em que os classificadores possuem uma grande discrepância em seu desempenho quando atribuindo rótulos para instâncias em  $U$ , i.e., um classificador possui uma acurácia muito maior que outros da *ensemble*, o desempenho da SCF Borda é comprometida. Por exemplo, no conjunto de dados *wdbc*, o algoritmo de aprendizado QDA possui uma acurácia de 85% contra uma acurácia de 93% apresentado pelo algoritmo NB, quando classificando as instâncias de  $U$  (Tabela 5.12). Em casos como este, o uso de uma *ensemble* contendo apenas os três melhores algoritmos de aprendizado para aquele conjunto de dados apresentou resultados superiores ao do uso de todos os sete algoritmos. Isto se dá pois os rankings gerados pelos algoritmos com as piores acurácias podem ser vistos como maus votos para o processo de agregação de rankings. Este problema de maus votos é um desafio conhecido de soluções de agregação de rankings, conforme discutido por (ADALI; HILL; MAGDON-ISMAIL, 2006).

Entretanto, em casos em que a diferença de acurácia nas classificações individuais foi pequena, a taxa de erro médio geral foi inferior após a aplicação da SCF Borda. Para os conjuntos de dados *diabetes* e *ionosphere*, por exemplo, a diferença de acurácia entre o pior e o melhor algoritmo de aprendizado, quando gerando o ranking sobre  $U$ , foi de no máximo seis pontos percentuais (Tabelas 5.4 e 5.27).

Tabela 5.27: Resultados individuais *ionosphere* 80%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.80 ( $\pm 0.06$ )	(A) 0.78 ( $\pm 0.05$ )	(A) 0.62 ( $\pm 0.05$ )
		(F) 0.83 ( $\pm 0.04$ )	(F) 0.71 ( $\pm 0.04$ )
		(P) 0.82 ( $\pm 0.05$ )	(P) 0.70 ( $\pm 0.04$ )
		(R) 0.85 ( $\pm 0.07$ )	(R) 0.71 ( $\pm 0.05$ )
GTB	0.81 ( $\pm 0.07$ )	(A) 0.83 ( $\pm 0.07$ )	(A) 0.67 ( $\pm 0.09$ )
		(F) 0.87 ( $\pm 0.06$ )	(F) 0.76 ( $\pm 0.06$ )
		(P) 0.86 ( $\pm 0.05$ )	(P) 0.72 ( $\pm 0.07$ )
		(R) 0.88 ( $\pm 0.09$ )	(R) 0.80 ( $\pm 0.07$ )
K-NN	0.75 ( $\pm 0.06$ )	(A) 0.80 ( $\pm 0.05$ )	(A) 0.65 ( $\pm 0.07$ )
		(F) 0.85 ( $\pm 0.04$ )	(F) 0.76 ( $\pm 0.05$ )
		(P) 0.81 ( $\pm 0.03$ )	(P) 0.70 ( $\pm 0.07$ )
		(R) 0.91 ( $\pm 0.08$ )	(R) 0.84 ( $\pm 0.09$ )
LDA	0.76 ( $\pm 0.06$ )	(A) 0.80 ( $\pm 0.04$ )	(A) 0.74 ( $\pm 0.06$ )
		(F) 0.86 ( $\pm 0.03$ )	(F) 0.82 ( $\pm 0.04$ )
		(P) 0.80 ( $\pm 0.04$ )	(P) 0.73 ( $\pm 0.04$ )
		(R) 0.92 ( $\pm 0.06$ )	(R) 0.94 ( $\pm 0.06$ )
LR	0.80 ( $\pm 0.04$ )	(A) 0.79 ( $\pm 0.04$ )	(A) 0.73 ( $\pm 0.05$ )
		(F) 0.84 ( $\pm 0.03$ )	(F) 0.82 ( $\pm 0.03$ )
		(P) 0.80 ( $\pm 0.04$ )	(P) 0.72 ( $\pm 0.03$ )
		(R) 0.90 ( $\pm 0.07$ )	(R) 0.95 ( $\pm 0.04$ )
NB	0.79 ( $\pm 0.06$ )	(A) 0.80 ( $\pm 0.07$ )	(A) 0.69 ( $\pm 0.20$ )
		(F) 0.83 ( $\pm 0.07$ )	(F) 0.75 ( $\pm 0.17$ )
		(P) 0.89 ( $\pm 0.05$ )	(P) 0.74 ( $\pm 0.14$ )
		(R) 0.79 ( $\pm 0.11$ )	(R) 0.76 ( $\pm 0.21$ )
QDA	0.72 ( $\pm 0.04$ )	(A) 0.83 ( $\pm 0.10$ )	(A) 0.63 ( $\pm 0.21$ )
		(F) 0.85 ( $\pm 0.11$ )	(F) 0.71 ( $\pm 0.18$ )
		(P) 0.94 ( $\pm 0.05$ )	(P) 0.70 ( $\pm 0.18$ )
		(R) 0.79 ( $\pm 0.17$ )	(R) 0.73 ( $\pm 0.23$ )

Para estes conjuntos de dados, remover os rankings gerados pelos algoritmos com pior

acurácia piorou a classificação final. Este resultado demonstra uma das vantagens da utilização de SCFs que trabalham com agregação de rankings como o agregador de resultados: colaboração entre os classificadores. Quando treinados com instâncias rotuladas por todo o grupo, os algoritmos que mostraram uma acurácia levemente menor sozinhos, foram capazes de contribuir positivamente para a criação do conjunto de instâncias rotuladas pelo algoritmo,  $L'$ . Além disso, estes testes mostram que utilizar múltiplos algoritmos de aprendizado e agregar os resultados das classificações dos mesmos apresenta melhores resultados do que utilizar apenas o algoritmo com a melhor acurácia, ou utilizar apenas um classificador, corroborando com resultados obtidos por (ZHOU; LI, 2005), (RECAMONDE-MENDOZA; BAZZAN, 2016), (BLUM; MITCHELL, 1998), entre outros (Capítulo 3).

## 5.6 Considerações Finais

Neste capítulo, foram apresentados os resultados obtidos através da realização de experimentos com o algoritmo desenvolvido. Além disto, foram determinadas as configurações utilizadas na instanciação do algoritmo, o que permite a repetição destes experimentos com novos conjuntos de dados no futuro.

Quanto aos resultados obtidos, os mesmos são satisfatórios em diversos casos, e mesmo em casos em que os resultados foram inferiores se comparados aos de outras soluções, foi possível obter lições sobre quais são as limitações da abordagem proposta.

Dentre estes resultados é importante destacar que o algoritmo social-training obteve resultados superiores ao tri-training em quinze de trinta e duas comparações (além de dois empates). Já na comparação com o democratic co-learning, a solução proposta nesta dissertação foi superior em dois de três casos.

Finalmente, neste capítulo existem resultados que apontam para as vantagens do uso de *ensembles* de classificadores, os quais são também mostrados em outros trabalhos na literatura.

## 6 CONCLUSÃO

Como apontado por diversos trabalhos ((ZHOU; LI, 2005), (ZHOU; GOLDMAN, 2004)), a "explosão de dados" cria desafios que serão cada vez mais relevantes no contexto de aprendizado de máquina. A respeito de dados não rotulados, o algoritmo social-training apresenta uma proposta diferente para trabalhar com conjuntos de dados que se enquadrem neste cenário. Como esta solução atribui rótulos para todas as instâncias não rotuladas de um conjunto de dados para a sua rotina de aprendizado semi-supervisionado, toda informação disponível é utilizada como uma fonte de conhecimento para os classificadores. Além disso, como social-training utiliza múltiplos algoritmos de classificação, o desempenho destes algoritmos sobre um dado conjunto de dados pode ser avaliado e melhorado através da agregação dos rankings dos resultados oferecida pelas SCFs.

Como foi visto no Capítulo 3, a solução apresentada nesta dissertação se diferencia das demais soluções na literatura em diversas esferas, seja por utilizar um número maior de algoritmos de aprendizado, diferentes métodos de votação e agregação de resultados ou estratégias de particionamento de dados. No âmbito de SCFs, as mesmas são utilizadas de uma nova maneira, em cenários de aprendizado semi-supervisionado, especificamente.

Referente ao algoritmo desenvolvido (Capítulo 4) o mesmo possui alta modularidade quanto aos seus componentes, o que permite que os resultados sejam expandidos para trabalhos futuros de maneira simples. É possível aplicar melhorias tanto na forma de novos algoritmos de aprendizado, quanto funções de escolha social sem ser necessário realizar mudanças na lógica do algoritmo, apenas alterando a sua forma de instanciação. Além disso, o ponto de entrada que permite a definição do conjunto de dados a ser utilizado é simples; entretanto, o processamento dos mesmos se mostrou problemático dependendo do formato utilizado.

Os experimentos apresentados no Capítulo 5 mostraram que, em diversos casos, utilizar todas as instâncias disponíveis em  $U$  com rótulos gerados pela agregação de rankings ( $L'$ ) reduz a taxa de erro médio dos classificadores em comparação com soluções que utilizam apenas rótulos que possuam um grande grau de confiança. Outro ponto importante que é mostrado é que utilizar múltiplos classificadores com bom desempenho para criar  $L'$  tem resultados superiores ao caso em que apenas o algoritmo com a melhor acurácia sobre o conjunto de dados é utilizado.

Os resultados dos experimentos demonstram que se rótulos corretos puderem ser atribuídos aos dados não rotulados, utilizar todas estas instâncias de treinamento produz melhores resultados do que utilizar apenas rótulos das instâncias que possuem um alto grau de confiança. As propriedades de agregação de rankings das SCFs Borda e Copeland foram a chave para obter

estes resultados. Estes testes também demonstraram que agregar resultados de múltiplos algoritmos de aprendizado através do uso de SCFs produz resultados superiores a utilizar um único algoritmo. Porém é necessário definir qual o limiar máximo de instâncias não rotuladas onde a agregação de rankings começa a perder sua eficiência, conforme mostrado em alguns testes (Seções 5.3 e 5.4). Além disso, em alguns casos seria benéfico remover alguns dos algoritmos de classificação que possuem um desempenho muito inferior aos demais conforme discutido na Seção 5.5. Isto poderia ser feito definindo algum grau máximo de discrepância na acurácia da classificação dentro da *ensemble*, por exemplo.

Trabalhos futuros são possíveis em diversas esferas, sendo as principais: mudanças na maneira em que as SCFs são utilizadas e estratégias de particionamento de dados. Quanto às SCFs, seria possível utilizar um sistema de votação quando classificando dados de teste, com um comportamento similar ao que é utilizado para definir os rótulos na solução (RECAMONDE-MENDOZA; BAZZAN, 2016). Isto é, além da agregação dos votos para definir os rótulos do conjunto  $L'$ , utilizar também agregação de rankings para definir o rótulo das instâncias do conjunto de teste.

Já no âmbito das partições de dados, seria possível a utilização de partições horizontais sobre os dados, visando simular cenários nos quais nem todas as instâncias estão disponíveis na mesma localização geograficamente. Além disso, cenários nos quais o conjunto de dados é muito grande e o tempo para o processamento completo por um simples processo de um algoritmo não é viável podem ser simulados. Ainda neste tipo de caso, outra situação possível é a utilização de partições verticais para simular casos em que as instâncias possuem atributos que podem conter restrições de privacidade e portanto precisam ser distribuídos. Outro possível trabalho futuro seria particionar as instâncias do conjunto  $U$  entre os múltiplos classificadores. Isto geraria o desafio da criação de rankings com dados faltantes, e, desta maneira, seria necessário compartilhar os rankings entre os múltiplos classificadores a fim de gerar um resultado agregado. Devido a isto, seria necessário definir maneiras de inferir um rótulo para as instâncias faltantes do ranking, pois de outra maneira não seria possível trabalhar com SCFs que utilizam agregação de rankings.

Além destes casos, o algoritmo poderia ser alterado para lidar com conjuntos de dados de classificação do tipo multi-classe. Para tanto, seriam necessárias algumas alterações ao algoritmo, uma vez que um único ranking de instâncias ordenado por probabilidade de pertencer a uma dada classe não pode ser criado para dados multi-classe, já que os mesmos possuem mais de duas classes. Uma possível solução para este cenário seria a criação de rankings individuais para cada instância, e posteriormente a agregação destes resultados. Um primeiro passo

nesta direção pode ser visto em (BOOS, 2017). Adicionalmente, seria possível testar a solução existente utilizando diferentes algoritmos de aprendizado e diferentes conjuntos de dados.

## REFERÊNCIAS

- ADALI, S.; HILL, B.; MAGDON-ISMAIL, M. The impact of ranker quality on rank aggregation algorithms: Information vs. robustness. In: **Proceedings of the 22Nd International Conference on Data Engineering Workshops**. Washington, DC, USA: IEEE Computer Society, 2006. (ICDEW '06), p. 37–. ISBN 0-7695-2571-7. Available from Internet: <<http://dx.doi.org/10.1109/ICDEW.2006.146>>.
- AZIMIFAR, M. et al. Semi-supervised local aggregation methodology. In: \_\_\_\_\_. **Computational Science and Its Applications – ICCSA 2015: 15th International Conference, Banff, AB, Canada, June 22-25, 2015, Proceedings, Part IV**. Cham: Springer International Publishing, 2015. p. 233–245. ISBN 978-3-319-21410-8. Available from Internet: <[http://dx.doi.org/10.1007/978-3-319-21410-8\\_18](http://dx.doi.org/10.1007/978-3-319-21410-8_18)>.
- BLUM, A.; MITCHELL, T. Combining labeled and unlabeled data with co-training. In: **Proceedings of the Eleventh Annual Conference on Computational Learning Theory**. New York, NY, USA: ACM, 1998. (COLT' 98), p. 92–100. ISBN 1-58113-057-0. Available from Internet: <<http://doi.acm.org/10.1145/279943.279962>>.
- BOOS, R. A. S. Trabalho de Conclusão - Ciência da Computação, **Funções de Escolha Social para Elaboração de Consenso em Aprendizado de Máquina Descentralizado: um Estudo em Problemas de Classificação Multiclasse**. 2017. (To appear).
- COHEN, W. W. Fast effective rule induction. In: **Twelfth International Conference on Machine Learning**. [S.l.]: Morgan Kaufmann, 1995. p. 115–123.
- COPELAND, A. A "reasonable" social welfare function. Dissertation (Master) — Seminar on Mathematics in Social Sciences, University of Michigan, 1951.
- DIETTERICH, T. G. Ensemble methods in machine learning. In: KITTLER, J.; ROLI, F. (Ed.). **Proceedings of the First International Workshop on Multiple Classifier Systems**. London, UK, UK: Springer-Verlag, 2000. p. 1–15.
- FOUNDATION, P. S. **Python Language Version 3.5.0**. 2015. Available from Internet: <<http://www.python.org>>. Accessed in: 2015-10-18.
- FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. **Annals of Statistics**, v. 29, p. 1189–1232, 2000.
- HADY, M. F. A. **Semi-supervised learning with committees: exploiting unlabeled data using ensemble learning algorithms**. Thesis (PhD), 2011. Available from Internet: <<https://doi.org/10.18725/OPARU-1750>>.
- HADY, M. F. A.; SCHWENKER, F. Combining committee-based semi-supervised learning and active learning. **Journal of Computer Science and Technology**, v. 25, n. 4, p. 681–698, 2010. ISSN 1860-4749. Available from Internet: <<http://dx.doi.org/10.1007/s11390-010-9357-6>>.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning**. New York, NY, USA: Springer New York Inc., 2001. (Springer Series in Statistics).
- JIANG, Z.; ZENG, J.; ZHANG, S. Inter-training: Exploiting unlabeled data in multi-classifier systems. **Knowledge-Based Systems**, p. 8 – 19, 2013. ISSN 0950-7051. Available from Internet: <<http://www.sciencedirect.com/science/article/pii/S0950705113000488>>.

- JR., M. P. P. Combining classifiers: From the creation of ensembles to the decision fusion. In: **SIBGRAPI Tutorials**. IEEE Computer Society, 2011. p. 1–10. ISBN 978-1-4577-1627-0. Available from Internet: <<http://dblp.uni-trier.de/db/conf/sibgrapi/sibgrapi2011t.html#Ponti11>>.
- KELLER, J. M.; GRAY, M. R.; GIVENS, J. A. A fuzzy k-nearest neighbor algorithm. **IEEE Transactions on Systems, Man, and Cybernetics**, SMC-15, n. 4, p. 580–585, July 1985. ISSN 0018-9472.
- KUNCHEVA, L. I. **Combining Pattern Classifiers: Methods and Algorithms**. [S.l.]: Wiley-Interscience, 2004. ISBN 0471210781.
- MATSUBARA, E. T. et al. Missing value imputation using a semi-supervised rank aggregation approach. In: \_\_\_\_\_. **Advances in Artificial Intelligence - SBIA 2008: 19th Brazilian Symposium on Artificial Intelligence Savador, Brazil, October 26-30, 2008. Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 217–226. ISBN 978-3-540-88190-2. Available from Internet: <[http://dx.doi.org/10.1007/978-3-540-88190-2\\_27](http://dx.doi.org/10.1007/978-3-540-88190-2_27)>.
- MCGREGGOR, D. **Ballot Box 0.1**. 2011. Available from Internet: <<http://pythonhosted.org/ballotbox/>>. Accessed in: 2015-10-18.
- MCKINNEY, W. Data structures for statistical computing in python. In: WALT, S. van der; MILLMAN, J. (Ed.). **Proceedings of the 9th Python in Science Conference**. [S.l.: s.n.], 2010. p. 51 – 56.
- NG, V.; CARDIE, C. Bootstrapping coreference classifiers with multiple machine learning algorithms. In: **Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing**. [S.l.: s.n.], 2003. p. 113–120.
- PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.
- RECAMONDE-MENDOZA, M.; BAZZAN, A. L. C. **Inf. Sci.**, v. 332, p. 56–71, 2016. Available from Internet: <<http://dblp.uni-trier.de/db/journals/isci/isci332.html#MendozaB16>>.
- RIVEST, R. L. Learning decision lists. **Mach. Learn.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 2, n. 3, p. 229–246, nov. 1987. ISSN 0885-6125. Available from Internet: <<http://dx.doi.org/10.1023/A:1022607331053>>.
- ROY, M.; GHOSH, S.; GHOSH, A. A novel approach for change detection of remotely sensed images using semi-supervised multiple classifier system. **Information Sciences**, v. 269, p. 35 – 47, 2014. ISSN 0020-0255. Available from Internet: <<http://www.sciencedirect.com/science/article/pii/S0020025514000735>>.
- WALT, S. van der; COLBERT, S. C.; VAROQUAUX, G. The numpy array: A structure for efficient numerical computation. **Computing in Science Engineering**, v. 13, n. 2, p. 22–30, March 2011. ISSN 1521-9615.
- ZHOU, Y.; GOLDMAN, S. Democratic co-learning. In: **Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on**. [S.l.: s.n.], 2004. p. 594–602. ISSN 1082-3409.

ZHOU, Z.-H.; LI, M. Tri-training: exploiting unlabeled data using three classifiers. **Knowledge and Data Engineering, IEEE Transactions on**, v. 17, n. 11, p. 1529–1541, Nov 2005. ISSN 1041-4347.

ZHU, X. Semi-supervised learning literature survey. 2006.

**APÊNDICE A — MÉTRICAS INDIVIDUAIS DOS CLASSIFICADORES**



### A.0.1 Taxa de dados não rotulados de 80%

Tabela A.1: Resultados individuais *bupa* 80%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.59 ( $\pm 0.03$ )	(A) 0.56 ( $\pm 0.13$ )	(A) 0.54 ( $\pm 0.06$ )
		(F) 0.50 ( $\pm 0.13$ )	(F) 0.29 ( $\pm 0.14$ )
		(P) 0.50 ( $\pm 0.14$ )	(P) 0.41 ( $\pm 0.17$ )
		(R) 0.52 ( $\pm 0.15$ )	(R) 0.24 ( $\pm 0.13$ )
GTB	0.60 ( $\pm 0.04$ )	(A) 0.59 ( $\pm 0.14$ )	(A) 0.55 ( $\pm 0.06$ )
		(F) 0.52 ( $\pm 0.13$ )	(F) 0.24 ( $\pm 0.15$ )
		(P) 0.55 ( $\pm 0.16$ )	(P) 0.35 ( $\pm 0.20$ )
		(R) 0.52 ( $\pm 0.15$ )	(R) 0.20 ( $\pm 0.13$ )
K-NN	0.63 ( $\pm 0.04$ )	(A) 0.63 ( $\pm 0.07$ )	(A) 0.55 ( $\pm 0.03$ )
		(F) 0.54 ( $\pm 0.07$ )	(F) 0.38 ( $\pm 0.08$ )
		(P) 0.60 ( $\pm 0.11$ )	(P) 0.46 ( $\pm 0.06$ )
		(R) 0.50 ( $\pm 0.08$ )	(R) 0.33 ( $\pm 0.09$ )
LDA	0.58 ( $\pm 0.06$ )	(A) 0.55 ( $\pm 0.13$ )	(A) 0.57 ( $\pm 0.06$ )
		(F) 0.49 ( $\pm 0.11$ )	(F) 0.23 ( $\pm 0.17$ )
		(P) 0.50 ( $\pm 0.12$ )	(P) 0.47 ( $\pm 0.29$ )
		(R) 0.51 ( $\pm 0.16$ )	(R) 0.18 ( $\pm 0.16$ )
LR	0.63 ( $\pm 0.04$ )	(A) 0.55 ( $\pm 0.13$ )	(A) 0.57 ( $\pm 0.05$ )
		(F) 0.48 ( $\pm 0.12$ )	(F) 0.22 ( $\pm 0.16$ )
		(P) 0.50 ( $\pm 0.14$ )	(P) 0.46 ( $\pm 0.29$ )
		(R) 0.49 ( $\pm 0.15$ )	(R) 0.17 ( $\pm 0.14$ )
NB	0.55 ( $\pm 0.05$ )	(A) 0.48 ( $\pm 0.08$ )	(A) 0.54 ( $\pm 0.07$ )
		(F) 0.46 ( $\pm 0.10$ )	(F) 0.26 ( $\pm 0.16$ )
		(P) 0.42 ( $\pm 0.06$ )	(P) 0.37 ( $\pm 0.21$ )
		(R) 0.55 ( $\pm 0.18$ )	(R) 0.22 ( $\pm 0.16$ )
QDA	0.54 ( $\pm 0.07$ )	(A) 0.52 ( $\pm 0.10$ )	(A) 0.55 ( $\pm 0.05$ )
		(F) 0.50 ( $\pm 0.09$ )	(F) 0.27 ( $\pm 0.13$ )
		(P) 0.46 ( $\pm 0.09$ )	(P) 0.40 ( $\pm 0.19$ )
		(R) 0.56 ( $\pm 0.13$ )	(R) 0.22 ( $\pm 0.13$ )

Tabela A.2: Resultados individuais *tic-tac-toe* 80%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.81 ( $\pm 0.29$ )	(A) 0.74 ( $\pm 0.20$ )	(A) 0.69 ( $\pm 0.22$ )
		(F) 0.76 ( $\pm 0.23$ )	(F) 0.67 ( $\pm 0.30$ )
		(P) 0.84 ( $\pm 0.15$ )	(P) 0.86 ( $\pm 0.27$ )
		(R) 0.74 ( $\pm 0.28$ )	(R) 0.59 ( $\pm 0.32$ )
GTB	0.96 ( $\pm 0.10$ )	(A) 0.72 ( $\pm 0.21$ )	(A) 0.58 ( $\pm 0.29$ )
		(F) 0.79 ( $\pm 0.18$ )	(F) 0.55 ( $\pm 0.35$ )
		(P) 0.74 ( $\pm 0.16$ )	(P) 0.62 ( $\pm 0.37$ )
		(R) 0.86 ( $\pm 0.22$ )	(R) 0.52 ( $\pm 0.35$ )
K-NN	0.74 ( $\pm 0.07$ )	(A) 0.63 ( $\pm 0.13$ )	(A) 0.66 ( $\pm 0.18$ )
		(F) 0.73 ( $\pm 0.11$ )	(F) 0.71 ( $\pm 0.18$ )
		(P) 0.68 ( $\pm 0.08$ )	(P) 0.77 ( $\pm 0.16$ )
		(R) 0.81 ( $\pm 0.17$ )	(R) 0.71 ( $\pm 0.23$ )
LDA	0.44 ( $\pm 0.15$ )	(A) 0.66 ( $\pm 0.26$ )	(A) 0.59 ( $\pm 0.29$ )
		(F) 0.74 ( $\pm 0.25$ )	(F) 0.64 ( $\pm 0.31$ )
		(P) 0.67 ( $\pm 0.24$ )	(P) 0.70 ( $\pm 0.30$ )
		(R) 0.85 ( $\pm 0.29$ )	(R) 0.69 ( $\pm 0.37$ )
LR	0.79 ( $\pm 0.13$ )	(A) 0.67 ( $\pm 0.26$ )	(A) 0.56 ( $\pm 0.28$ )
		(F) 0.75 ( $\pm 0.26$ )	(F) 0.61 ( $\pm 0.31$ )
		(P) 0.68 ( $\pm 0.24$ )	(P) 0.68 ( $\pm 0.30$ )
		(R) 0.86 ( $\pm 0.29$ )	(R) 0.66 ( $\pm 0.39$ )
NB	0.63 ( $\pm 0.20$ )	(A) 0.64 ( $\pm 0.24$ )	(A) 0.52 ( $\pm 0.27$ )
		(F) 0.73 ( $\pm 0.25$ )	(F) 0.54 ( $\pm 0.34$ )
		(P) 0.66 ( $\pm 0.22$ )	(P) 0.60 ( $\pm 0.32$ )
		(R) 0.82 ( $\pm 0.29$ )	(R) 0.57 ( $\pm 0.39$ )
QDA	0.43 ( $\pm 0.06$ )	(A) 0.61 ( $\pm 0.08$ )	(A) 0.49 ( $\pm 0.26$ )
		(F) 0.74 ( $\pm 0.09$ )	(F) 0.57 ( $\pm 0.29$ )
		(P) 0.65 ( $\pm 0.03$ )	(P) 0.54 ( $\pm 0.23$ )
		(R) 0.88 ( $\pm 0.19$ )	(R) 0.65 ( $\pm 0.38$ )



## A.0.2 Taxa de dados não rotulados de 60%

Tabela A.3: Resultados individuais *bupa* 60%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.61 ( $\pm 0.04$ )	(A) 0.56 ( $\pm 0.09$ )	(A) 0.54 ( $\pm 0.08$ )
		(F) 0.45 ( $\pm 0.11$ )	(F) 0.36 ( $\pm 0.14$ )
		(P) 0.48 ( $\pm 0.11$ )	(P) 0.44 ( $\pm 0.13$ )
		(R) 0.44 ( $\pm 0.15$ )	(R) 0.33 ( $\pm 0.15$ )
GTB	0.63 ( $\pm 0.03$ )	(A) 0.60 ( $\pm 0.10$ )	(A) 0.55 ( $\pm 0.07$ )
		(F) 0.48 ( $\pm 0.14$ )	(F) 0.30 ( $\pm 0.15$ )
		(P) 0.55 ( $\pm 0.15$ )	(P) 0.45 ( $\pm 0.14$ )
		(R) 0.44 ( $\pm 0.16$ )	(R) 0.25 ( $\pm 0.14$ )
K-NN	0.66 ( $\pm 0.03$ )	(A) 0.65 ( $\pm 0.05$ )	(A) 0.58 ( $\pm 0.05$ )
		(F) 0.56 ( $\pm 0.06$ )	(F) 0.44 ( $\pm 0.06$ )
		(P) 0.59 ( $\pm 0.08$ )	(P) 0.51 ( $\pm 0.08$ )
		(R) 0.53 ( $\pm 0.07$ )	(R) 0.39 ( $\pm 0.06$ )
LDA	0.66 ( $\pm 0.04$ )	(A) 0.60 ( $\pm 0.07$ )	(A) 0.59 ( $\pm 0.03$ )
		(F) 0.47 ( $\pm 0.08$ )	(F) 0.29 ( $\pm 0.10$ )
		(P) 0.55 ( $\pm 0.10$ )	(P) 0.58 ( $\pm 0.12$ )
		(R) 0.44 ( $\pm 0.13$ )	(R) 0.21 ( $\pm 0.08$ )
LR	0.68 ( $\pm 0.02$ )	(A) 0.62 ( $\pm 0.07$ )	(A) 0.60 ( $\pm 0.03$ )
		(F) 0.51 ( $\pm 0.09$ )	(F) 0.28 ( $\pm 0.11$ )
		(P) 0.57 ( $\pm 0.11$ )	(P) 0.61 ( $\pm 0.27$ )
		(R) 0.47 ( $\pm 0.12$ )	(R) 0.20 ( $\pm 0.09$ )
NB	0.56 ( $\pm 0.05$ )	(A) 0.51 ( $\pm 0.09$ )	(A) 0.54 ( $\pm 0.07$ )
		(F) 0.47 ( $\pm 0.12$ )	(F) 0.26 ( $\pm 0.16$ )
		(P) 0.44 ( $\pm 0.09$ )	(P) 0.39 ( $\pm 0.21$ )
		(R) 0.56 ( $\pm 0.24$ )	(R) 0.22 ( $\pm 0.16$ )
QDA	0.61 ( $\pm 0.03$ )	(A) 0.55 ( $\pm 0.06$ )	(A) 0.56 ( $\pm 0.03$ )
		(F) 0.49 ( $\pm 0.08$ )	(F) 0.30 ( $\pm 0.12$ )
		(P) 0.48 ( $\pm 0.07$ )	(P) 0.46 ( $\pm 0.08$ )
		(R) 0.53 ( $\pm 0.17$ )	(R) 0.24 ( $\pm 0.12$ )

Tabela A.4: Resultados individuais *ionosphere* 60%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.83 ( $\pm 0.06$ )	(A) 0.82 ( $\pm 0.03$ )	(A) 0.66 ( $\pm 0.10$ )
		(F) 0.86 ( $\pm 0.03$ )	(F) 0.72 ( $\pm 0.09$ )
		(P) 0.86 ( $\pm 0.04$ )	(P) 0.75 ( $\pm 0.08$ )
		(R) 0.87 ( $\pm 0.05$ )	(R) 0.71 ( $\pm 0.11$ )
GTB	0.86 ( $\pm 0.05$ )	(A) 0.87 ( $\pm 0.03$ )	(A) 0.75 ( $\pm 0.09$ )
		(F) 0.90 ( $\pm 0.03$ )	(F) 0.81 ( $\pm 0.07$ )
		(P) 0.88 ( $\pm 0.03$ )	(P) 0.79 ( $\pm 0.06$ )
		(R) 0.92 ( $\pm 0.05$ )	(R) 0.84 ( $\pm 0.08$ )
K-NN	0.79 ( $\pm 0.05$ )	(A) 0.82 ( $\pm 0.03$ )	(A) 0.67 ( $\pm 0.06$ )
		(F) 0.87 ( $\pm 0.02$ )	(F) 0.77 ( $\pm 0.04$ )
		(P) 0.81 ( $\pm 0.03$ )	(P) 0.70 ( $\pm 0.05$ )
		(R) 0.94 ( $\pm 0.04$ )	(R) 0.85 ( $\pm 0.07$ )
LDA	0.81 ( $\pm 0.04$ )	(A) 0.83 ( $\pm 0.04$ )	(A) 0.78 ( $\pm 0.06$ )
		(F) 0.87 ( $\pm 0.03$ )	(F) 0.85 ( $\pm 0.04$ )
		(P) 0.81 ( $\pm 0.04$ )	(P) 0.76 ( $\pm 0.05$ )
		(R) 0.95 ( $\pm 0.04$ )	(R) 0.96 ( $\pm 0.04$ )
LR	0.83 ( $\pm 0.04$ )	(A) 0.82 ( $\pm 0.03$ )	(A) 0.78 ( $\pm 0.05$ )
		(F) 0.87 ( $\pm 0.02$ )	(F) 0.85 ( $\pm 0.03$ )
		(P) 0.81 ( $\pm 0.03$ )	(P) 0.76 ( $\pm 0.05$ )
		(R) 0.94 ( $\pm 0.04$ )	(R) 0.97 ( $\pm 0.03$ )
NB	0.82 ( $\pm 0.04$ )	(A) 0.84 ( $\pm 0.05$ )	(A) 0.75 ( $\pm 0.12$ )
		(F) 0.87 ( $\pm 0.04$ )	(F) 0.81 ( $\pm 0.11$ )
		(P) 0.90 ( $\pm 0.04$ )	(P) 0.81 ( $\pm 0.10$ )
		(R) 0.86 ( $\pm 0.08$ )	(R) 0.81 ( $\pm 0.14$ )
QDA	0.78 ( $\pm 0.10$ )	(A) 0.87 ( $\pm 0.07$ )	(A) 0.80 ( $\pm 0.16$ )
		(F) 0.89 ( $\pm 0.06$ )	(F) 0.85 ( $\pm 0.11$ )
		(P) 0.94 ( $\pm 0.05$ )	(P) 0.86 ( $\pm 0.15$ )
		(R) 0.85 ( $\pm 0.10$ )	(R) 0.85 ( $\pm 0.10$ )

Tabela A.5: Resultados individuais *tic-tac-toe* 60%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	1.00 ( $\pm 0.00$ )	(A) 0.73 ( $\pm 0.19$ )	(A) 0.73 ( $\pm 0.15$ )
		(F) 0.74 ( $\pm 0.24$ )	(F) 0.74 ( $\pm 0.20$ )
		(P) 0.83 ( $\pm 0.14$ )	(P) 0.88 ( $\pm 0.09$ )
		(R) 0.75 ( $\pm 0.32$ )	(R) 0.70 ( $\pm 0.27$ )
GTB	1.00 ( $\pm 0.00$ )	(A) 0.74 ( $\pm 0.23$ )	(A) 0.68 ( $\pm 0.25$ )
		(F) 0.78 ( $\pm 0.24$ )	(F) 0.66 ( $\pm 0.33$ )
		(P) 0.75 ( $\pm 0.21$ )	(P) 0.73 ( $\pm 0.30$ )
		(R) 0.84 ( $\pm 0.28$ )	(R) 0.63 ( $\pm 0.36$ )
K-NN	0.77 ( $\pm 0.07$ )	(A) 0.64 ( $\pm 0.07$ )	(A) 0.65 ( $\pm 0.16$ )
		(F) 0.75 ( $\pm 0.07$ )	(F) 0.71 ( $\pm 0.17$ )
		(P) 0.69 ( $\pm 0.05$ )	(P) 0.77 ( $\pm 0.14$ )
		(R) 0.82 ( $\pm 0.13$ )	(R) 0.70 ( $\pm 0.23$ )
LDA	0.63 ( $\pm 0.25$ )	(A) 0.62 ( $\pm 0.03$ )	(A) 0.63 ( $\pm 0.27$ )
		(F) 0.70 ( $\pm 0.23$ )	(F) 0.66 ( $\pm 0.31$ )
		(P) 0.68 ( $\pm 0.23$ )	(P) 0.74 ( $\pm 0.28$ )
		(R) 0.78 ( $\pm 0.31$ )	(R) 0.70 ( $\pm 0.39$ )
LR	0.93 ( $\pm 0.10$ )	(A) 0.63 ( $\pm 0.23$ )	(A) 0.64 ( $\pm 0.21$ )
		(F) 0.70 ( $\pm 0.25$ )	(F) 0.68 ( $\pm 0.24$ )
		(P) 0.69 ( $\pm 0.22$ )	(P) 0.78 ( $\pm 0.18$ )
		(R) 0.77 ( $\pm 0.33$ )	(R) 0.71 ( $\pm 0.34$ )
NB	0.87 ( $\pm 0.22$ )	(A) 0.51 ( $\pm 0.25$ )	(A) 0.55 ( $\pm 0.25$ )
		(F) 0.55 ( $\pm 0.30$ )	(F) 0.56 ( $\pm 0.35$ )
		(P) 0.60 ( $\pm 0.30$ )	(P) 0.63 ( $\pm 0.32$ )
		(R) 0.59 ( $\pm 0.37$ )	(R) 0.60 ( $\pm 0.40$ )
QDA	0.64 ( $\pm 0.16$ )	(A) 0.58 ( $\pm 0.22$ )	(A) 0.54 ( $\pm 0.20$ )
		(F) 0.69 ( $\pm 0.24$ )	(F) 0.62 ( $\pm 0.030$ )
		(P) 0.62 ( $\pm 0.19$ )	(P) 0.53 ( $\pm 0.24$ )
		(R) 0.82 ( $\pm 0.32$ )	(R) 0.74 ( $\pm 0.38$ )



## A.0.3 Taxa de dados não rotulados de 40%

Tabela A.6: Resultados individuais *bupa* 40%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.65 ( $\pm 0.03$ )	(A) 0.64 ( $\pm 0.05$ )	(A) 0.59 ( $\pm 0.05$ )
		(F) 0.55 ( $\pm 0.09$ )	(F) 0.52 ( $\pm 0.05$ )
		(P) 0.58 ( $\pm 0.06$ )	(P) 0.53 ( $\pm 0.08$ )
		(R) 0.54 ( $\pm 0.13$ )	(R) 0.54 ( $\pm 0.12$ )
GTB	0.70 ( $\pm 0.04$ )	(A) 0.69 ( $\pm 0.06$ )	(A) 0.61 ( $\pm 0.08$ )
		(F) 0.61 ( $\pm 0.07$ )	(F) 0.54 ( $\pm 0.11$ )
		(P) 0.67 ( $\pm 0.11$ )	(P) 0.54 ( $\pm 0.10$ )
		(R) 0.56 ( $\pm 0.07$ )	(R) 0.55 ( $\pm 0.17$ )
K-NN	0.67 ( $\pm 0.02$ )	(A) 0.66 ( $\pm 0.05$ )	(A) 0.60 ( $\pm 0.04$ )
		(F) 0.56 ( $\pm 0.06$ )	(F) 0.48 ( $\pm 0.04$ )
		(P) 0.62 ( $\pm 0.07$ )	(P) 0.54 ( $\pm 0.06$ )
		(R) 0.52 ( $\pm 0.07$ )	(R) 0.44 ( $\pm 0.05$ )
LDA	0.68 ( $\pm 0.04$ )	(A) 0.66 ( $\pm 0.03$ )	(A) 0.62 ( $\pm 0.06$ )
		(F) 0.57 ( $\pm 0.05$ )	(F) 0.46 ( $\pm 0.16$ )
		(P) 0.61 ( $\pm 0.05$ )	(P) 0.61 ( $\pm 0.10$ )
		(R) 0.54 ( $\pm 0.07$ )	(R) 0.42 ( $\pm 0.21$ )
LR	0.69 ( $\pm 0.03$ )	(A) 0.68 ( $\pm 0.04$ )	(A) 0.61 ( $\pm 0.06$ )
		(F) 0.60 ( $\pm 0.05$ )	(F) 0.42 ( $\pm 0.19$ )
		(P) 0.64 ( $\pm 0.05$ )	(P) 0.60 ( $\pm 0.13$ )
		(R) 0.57 ( $\pm 0.05$ )	(R) 0.39 ( $\pm 0.23$ )
NB	0.57 ( $\pm 0.04$ )	(A) 0.53 ( $\pm 0.03$ )	(A) 0.55 ( $\pm 0.07$ )
		(F) 0.57 ( $\pm 0.03$ )	(F) 0.42 ( $\pm 0.19$ )
		(P) 0.47 ( $\pm 0.02$ )	(P) 0.43 ( $\pm 0.13$ )
		(R) 0.75 ( $\pm 0.08$ )	(R) 0.46 ( $\pm 0.25$ )
QDA	0.63 ( $\pm 0.04$ )	(A) 0.59 ( $\pm 0.04$ )	(A) 0.58 ( $\pm 0.06$ )
		(F) 0.59 ( $\pm 0.02$ )	(F) 0.48 ( $\pm 0.12$ )
		(P) 0.51 ( $\pm 0.04$ )	(P) 0.52 ( $\pm 0.07$ )
		(R) 0.71 ( $\pm 0.04$ )	(R) 0.50 ( $\pm 0.19$ )

Tabela A.7: Resultados individuais *ionosphere* 40%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.85 ( $\pm 0.06$ )	(A) 0.84 ( $\pm 0.02$ )	(A) 0.72 ( $\pm 0.11$ )
		(F) 0.88 ( $\pm 0.02$ )	(F) 0.77 ( $\pm 0.10$ )
		(P) 0.86 ( $\pm 0.04$ )	(P) 0.78 ( $\pm 0.08$ )
		(R) 0.91 ( $\pm 0.04$ )	(R) 0.78 ( $\pm 0.13$ )
GTB	0.89 ( $\pm 0.04$ )	(A) 0.90 ( $\pm 0.03$ )	(A) 0.80 ( $\pm 0.09$ )
		(F) 0.92 ( $\pm 0.02$ )	(F) 0.85 ( $\pm 0.08$ )
		(P) 0.89 ( $\pm 0.03$ )	(P) 0.83 ( $\pm 0.07$ )
		(R) 0.96 ( $\pm 0.02$ )	(R) 0.87 ( $\pm 0.11$ )
K-NN	0.81 ( $\pm 0.06$ )	(A) 0.84 ( $\pm 0.04$ )	(A) 0.75 ( $\pm 0.09$ )
		(F) 0.89 ( $\pm 0.02$ )	(F) 0.82 ( $\pm 0.07$ )
		(P) 0.82 ( $\pm 0.03$ )	(P) 0.76 ( $\pm 0.06$ )
		(R) 0.97 ( $\pm 0.02$ )	(R) 0.89 ( $\pm 0.10$ )
LDA	0.83 ( $\pm 0.03$ )	(A) 0.84 ( $\pm 0.04$ )	(A) 0.82 ( $\pm 0.04$ )
		(F) 0.89 ( $\pm 0.02$ )	(F) 0.87 ( $\pm 0.03$ )
		(P) 0.82 ( $\pm 0.04$ )	(P) 0.79 ( $\pm 0.05$ )
		(R) 0.97 ( $\pm 0.01$ )	(R) 0.98 ( $\pm 0.01$ )
LR	0.83 ( $\pm 0.05$ )	(A) 0.83 ( $\pm 0.04$ )	(A) 0.81 ( $\pm 0.04$ )
		(F) 0.88 ( $\pm 0.03$ )	(F) 0.87 ( $\pm 0.02$ )
		(P) 0.81 ( $\pm 0.04$ )	(P) 0.78 ( $\pm 0.04$ )
		(R) 0.96 ( $\pm 0.02$ )	(R) 0.99 ( $\pm 0.01$ )
NB	0.84 ( $\pm 0.03$ )	(A) 0.87 ( $\pm 0.05$ )	(A) 0.80 ( $\pm 0.07$ )
		(F) 0.90 ( $\pm 0.04$ )	(F) 0.84 ( $\pm 0.06$ )
		(P) 0.88 ( $\pm 0.04$ )	(P) 0.87 ( $\pm 0.07$ )
		(R) 0.92 ( $\pm 0.07$ )	(R) 0.82 ( $\pm 0.09$ )
QDA	0.88 ( $\pm 0.06$ )	(A) 0.90 ( $\pm 0.05$ )	(A) 0.84 ( $\pm 0.07$ )
		(F) 0.92 ( $\pm 0.04$ )	(F) 0.86 ( $\pm 0.06$ )
		(P) 0.95 ( $\pm 0.04$ )	(P) 0.93 ( $\pm 0.08$ )
		(R) 0.90 ( $\pm 0.05$ )	(R) 0.82 ( $\pm 0.10$ )

Tabela A.8: Resultados individuais *tic-tac-toe* 40%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	1.00 ( $\pm 0.00$ )	(A) 0.70 ( $\pm 0.25$ )	(A) 0.71 ( $\pm 0.29$ )
		(F) 0.71 ( $\pm 0.27$ )	(F) 0.67 ( $\pm 0.36$ )
		(P) 0.82 ( $\pm 0.19$ )	(P) 0.76 ( $\pm 0.36$ )
		(R) 0.66 ( $\pm 0.30$ )	(R) 0.63 ( $\pm 0.37$ )
GTB	1.00 ( $\pm 0.00$ )	(A) 0.64 ( $\pm 0.25$ )	(A) 0.52 ( $\pm 0.32$ )
		(F) 0.66 ( $\pm 0.25$ )	(F) 0.54 ( $\pm 0.36$ )
		(P) 0.76 ( $\pm 0.18$ )	(P) 0.53 ( $\pm 0.35$ )
		(R) 0.62 ( $\pm 0.30$ )	(R) 0.57 ( $\pm 0.39$ )
K-NN	0.85 ( $\pm 0.08$ )	(A) 0.66 ( $\pm 0.09$ )	(A) 0.68 ( $\pm 0.12$ )
		(F) 0.74 ( $\pm 0.10$ )	(F) 0.74 ( $\pm 0.14$ )
		(P) 0.72 ( $\pm 0.06$ )	(P) 0.77 ( $\pm 0.10$ )
		(R) 0.79 ( $\pm 0.19$ )	(R) 0.75 ( $\pm 0.21$ )
LDA	0.75 ( $\pm 0.31$ )	(A) 0.69 ( $\pm 0.23$ )	(A) 0.68 ( $\pm 0.29$ )
		(F) 0.71 ( $\pm 0.28$ )	(F) 0.68 ( $\pm 0.36$ )
		(P) 0.74 ( $\pm 0.22$ )	(P) 0.77 ( $\pm 0.28$ )
		(R) 0.75 ( $\pm 0.35$ )	(R) 0.72 ( $\pm 0.41$ )
LR	0.96 ( $\pm 0.10$ )	(A) 0.69 ( $\pm 0.20$ )	(A) 0.69 ( $\pm 0.25$ )
		(F) 0.72 ( $\pm 0.26$ )	(F) 0.68 ( $\pm 0.34$ )
		(P) 0.77 ( $\pm 0.18$ )	(P) 0.76 ( $\pm 0.28$ )
		(R) 0.75 ( $\pm 0.34$ )	(R) 0.73 ( $\pm 0.41$ )
NB	0.93 ( $\pm 0.19$ )	(A) 0.51 ( $\pm 0.28$ )	(A) 0.52 ( $\pm 0.32$ )
		(F) 0.54 ( $\pm 0.34$ )	(F) 0.54 ( $\pm 0.36$ )
		(P) 0.57 ( $\pm 0.33$ )	(P) 0.53 ( $\pm 0.35$ )
		(R) 0.55 ( $\pm 0.38$ )	(R) 0.57 ( $\pm 0.39$ )
QDA	0.71 ( $\pm 0.25$ )	(A) 0.54 ( $\pm 0.18$ )	(A) 0.57 ( $\pm 0.18$ )
		(F) 0.67 ( $\pm 0.20$ )	(F) 0.70 ( $\pm 0.20$ )
		(P) 0.59 ( $\pm 0.15$ )	(P) 0.60 ( $\pm 0.15$ )
		(R) 0.79 ( $\pm 0.28$ )	(R) 0.84 ( $\pm 0.27$ )



## A.0.4 Taxa de dados não rotulados de 20%

Tabela A.9: Resultados individuais *bupa* 20%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.63 ( $\pm 0.07$ )	(A) 0.60 ( $\pm 0.04$ )	(A) 0.60 ( $\pm 0.05$ )
		(F) 0.54 ( $\pm 0.05$ )	(F) 0.52 ( $\pm 0.07$ )
		(P) 0.52 ( $\pm 0.04$ )	(P) 0.54 ( $\pm 0.06$ )
		(R) 0.57 ( $\pm 0.08$ )	(R) 0.52 ( $\pm 0.09$ )
GTB	0.70 ( $\pm 0.05$ )	(A) 0.72 ( $\pm 0.05$ )	(A) 0.66 ( $\pm 0.06$ )
		(F) 0.64 ( $\pm 0.06$ )	(F) 0.56 ( $\pm 0.07$ )
		(P) 0.69 ( $\pm 0.07$ )	(P) 0.65 ( $\pm 0.11$ )
		(R) 0.60 ( $\pm 0.07$ )	(R) 0.50 ( $\pm 0.07$ )
K-NN	0.66 ( $\pm 0.04$ )	(A) 0.66 ( $\pm 0.04$ )	(A) 0.63 ( $\pm 0.03$ )
		(F) 0.55 ( $\pm 0.05$ )	(F) 0.51 ( $\pm 0.04$ )
		(P) 0.62 ( $\pm 0.06$ )	(P) 0.58 ( $\pm 0.05$ )
		(R) 0.50 ( $\pm 0.06$ )	(R) 0.46 ( $\pm 0.06$ )
LDA	0.68 ( $\pm 0.04$ )	(A) 0.66 ( $\pm 0.03$ )	(A) 0.65 ( $\pm 0.05$ )
		(F) 0.57 ( $\pm 0.04$ )	(F) 0.49 ( $\pm 0.11$ )
		(P) 0.62 ( $\pm 0.06$ )	(P) 0.63 ( $\pm 0.06$ )
		(R) 0.53 ( $\pm 0.05$ )	(R) 0.41 ( $\pm 0.14$ )
LR	0.70 ( $\pm 0.03$ )	(A) 0.68 ( $\pm 0.02$ )	(A) 0.66 ( $\pm 0.03$ )
		(F) 0.59 ( $\pm 0.02$ )	(F) 0.51 ( $\pm 0.08$ )
		(P) 0.64 ( $\pm 0.04$ )	(P) 0.67 ( $\pm 0.05$ )
		(R) 0.56 ( $\pm 0.04$ )	(R) 0.43 ( $\pm 0.11$ )
NB	0.52 ( $\pm 0.09$ )	(A) 0.53 ( $\pm 0.07$ )	(A) 0.58 ( $\pm 0.09$ )
		(F) 0.58 ( $\pm 0.04$ )	(F) 0.53 ( $\pm 0.11$ )
		(P) 0.47 ( $\pm 0.05$ )	(P) 0.52 ( $\pm 0.09$ )
		(R) 0.78 ( $\pm 0.06$ )	(R) 0.58 ( $\pm 0.18$ )
QDA	0.61 ( $\pm 0.08$ )	(A) 0.59 ( $\pm 0.05$ )	(A) 0.63 ( $\pm 0.07$ )
		(F) 0.60 ( $\pm 0.04$ )	(F) 0.57 ( $\pm 0.09$ )
		(P) 0.52 ( $\pm 0.05$ )	(P) 0.57 ( $\pm 0.08$ )
		(R) 0.72 ( $\pm 0.06$ )	(R) 0.60 ( $\pm 0.15$ )

Tabela A.10: Resultados individuais *ionosphere* 20%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	0.86 ( $\pm 0.07$ )	(A) 0.86 ( $\pm 0.02$ )	(A) 0.80 ( $\pm 0.06$ )
		(F) 0.90 ( $\pm 0.02$ )	(F) 0.85 ( $\pm 0.05$ )
		(P) 0.87 ( $\pm 0.03$ )	(P) 0.84 ( $\pm 0.05$ )
		(R) 0.92 ( $\pm 0.03$ )	(R) 0.86 ( $\pm 0.04$ )
GTB	0.90 ( $\pm 0.05$ )	(A) 0.90 ( $\pm 0.03$ )	(A) 0.88 ( $\pm 0.03$ )
		(F) 0.93 ( $\pm 0.02$ )	(F) 0.91 ( $\pm 0.02$ )
		(P) 0.89 ( $\pm 0.03$ )	(P) 0.87 ( $\pm 0.03$ )
		(R) 0.97 ( $\pm 0.02$ )	(R) 0.95 ( $\pm 0.02$ )
K-NN	0.79 ( $\pm 0.06$ )	(A) 0.84 ( $\pm 0.04$ )	(A) 0.82 ( $\pm 0.04$ )
		(F) 0.89 ( $\pm 0.02$ )	(F) 0.87 ( $\pm 0.03$ )
		(P) 0.81 ( $\pm 0.04$ )	(P) 0.79 ( $\pm 0.04$ )
		(R) 0.98 ( $\pm 0.01$ )	(R) 0.97 ( $\pm 0.03$ )
LDA	0.82 ( $\pm 0.06$ )	(A) 0.85 ( $\pm 0.04$ )	(A) 0.83 ( $\pm 0.04$ )
		(F) 0.89 ( $\pm 0.03$ )	(F) 0.88 ( $\pm 0.02$ )
		(P) 0.82 ( $\pm 0.04$ )	(P) 0.80 ( $\pm 0.04$ )
		(R) 0.97 ( $\pm 0.01$ )	(R) 0.98 ( $\pm 0.01$ )
LR	0.81 ( $\pm 0.07$ )	(A) 0.84 ( $\pm 0.04$ )	(A) 0.83 ( $\pm 0.03$ )
		(F) 0.88 ( $\pm 0.03$ )	(F) 0.88 ( $\pm 0.02$ )
		(P) 0.82 ( $\pm 0.04$ )	(P) 0.80 ( $\pm 0.04$ )
		(R) 0.96 ( $\pm 0.02$ )	(R) 0.97 ( $\pm 0.01$ )
NB	0.86 ( $\pm 0.05$ )	(A) 0.88 ( $\pm 0.05$ )	(A) 0.82 ( $\pm 0.08$ )
		(F) 0.91 ( $\pm 0.04$ )	(F) 0.85 ( $\pm 0.07$ )
		(P) 0.88 ( $\pm 0.05$ )	(P) 0.89 ( $\pm 0.06$ )
		(R) 0.94 ( $\pm 0.05$ )	(R) 0.82 ( $\pm 0.11$ )
QDA	0.90 ( $\pm 0.07$ )	(A) 0.88 ( $\pm 0.08$ )	(A) 0.86 ( $\pm 0.03$ )
		(F) 0.90 ( $\pm 0.08$ )	(F) 0.88 ( $\pm 0.03$ )
		(P) 0.95 ( $\pm 0.03$ )	(P) 0.96 ( $\pm 0.04$ )
		(R) 0.87 ( $\pm 0.12$ )	(R) 0.81 ( $\pm 0.06$ )

Tabela A.11: Resultados individuais *tic-tac-toe* 20%. ( $\pm$  desvio padrão)

Algoritmo de aprendizado	Acurácia sobre $U$	Resultados sobre $T$	
		Borda	Copeland
C4.5	1.00 ( $\pm 0.00$ )	(A) 0.73 ( $\pm 0.26$ )	(A) 0.86 ( $\pm 0.18$ )
		(F) 0.72 ( $\pm 0.28$ )	(F) 0.86 ( $\pm 0.19$ )
		(P) 0.87 ( $\pm 0.19$ )	(P) 0.93 ( $\pm 0.09$ )
		(R) 0.66 ( $\pm 0.32$ )	(R) 0.82 ( $\pm 0.24$ )
GTB	1.00 ( $\pm 0.00$ )	(A) 0.74 ( $\pm 0.27$ )	(A) 0.86 ( $\pm 0.20$ )
		(F) 0.74 ( $\pm 0.28$ )	(F) 0.86 ( $\pm 0.22$ )
		(P) 0.87 ( $\pm 0.19$ )	(P) 0.92 ( $\pm 0.12$ )
		(R) 0.69 ( $\pm 0.33$ )	(R) 0.84 ( $\pm 0.27$ )
K-NN	0.93 ( $\pm 0.05$ )	(A) 0.77 ( $\pm 0.09$ )	(A) 0.87 ( $\pm 0.11$ )
		(F) 0.83 ( $\pm 0.07$ )	(F) 0.89 ( $\pm 0.09$ )
		(P) 0.81 ( $\pm 0.08$ )	(P) 0.93 ( $\pm 0.08$ )
		(R) 0.86 ( $\pm 0.13$ )	(R) 0.87 ( $\pm 0.14$ )
LDA	0.94 ( $\pm 0.18$ )	(A) 0.89 ( $\pm 0.12$ )	(A) 1.00 ( $\pm 0.01$ )
		(F) 0.92 ( $\pm 0.09$ )	(F) 1.00 ( $\pm 0.01$ )
		(P) 0.90 ( $\pm 0.10$ )	(P) 1.00 ( $\pm 0.01$ )
		(R) 0.95 ( $\pm 0.11$ )	(R) 1.00 ( $\pm 0.01$ )
LR	1.00 ( $\pm 0.00$ )	(A) 0.77 ( $\pm 0.21$ )	(A) 0.92 ( $\pm 0.16$ )
		(F) 0.82 ( $\pm 0.20$ )	(F) 0.93 ( $\pm 0.13$ )
		(P) 0.80 ( $\pm 0.19$ )	(P) 0.95 ( $\pm 0.11$ )
		(R) 0.87 ( $\pm 0.26$ )	(R) 0.91 ( $\pm 0.16$ )
NB	1.00 ( $\pm 0.00$ )	(A) 0.52 ( $\pm 0.34$ )	(A) 0.84 ( $\pm 0.29$ )
		(F) 0.59 ( $\pm 0.34$ )	(F) 0.86 ( $\pm 0.27$ )
		(P) 0.59 ( $\pm 0.33$ )	(P) 0.86 ( $\pm 0.26$ )
		(R) 0.60 ( $\pm 0.37$ )	(R) 0.87 ( $\pm 0.28$ )
QDA	0.90 ( $\pm 0.20$ )	(A) 0.47 ( $\pm 0.21$ )	(A) 0.71 ( $\pm 0.23$ )
		(F) 0.58 ( $\pm 0.27$ )	(F) 0.78 ( $\pm 0.23$ )
		(P) 0.51 ( $\pm 0.23$ )	(P) 0.71 ( $\pm 0.20$ )
		(R) 0.68 ( $\pm 0.35$ )	(R) 0.87 ( $\pm 0.26$ )