



Universidade Federal de Pernambuco
Centro de Informática

Pós-graduação em Ciência da Computação

**Um Novo Algoritmo de Agrupamento
Semi-Supervisionado Baseado no *Fuzzy*
*C-Means***

Valmir Macário Filho

Dissertação de Mestrado

Recife
28 de Agosto de 2009

Universidade Federal de Pernambuco
Centro de Informática

Valmir Macário Filho

**Um Novo Algoritmo de Agrupamento Semi-Supervisionado
Baseado no *Fuzzy C-Means***

*Trabalho apresentado ao Programa de Pós-graduação em
Ciência da Computação do Centro de Informática da Uni-
versidade Federal de Pernambuco como requisito parcial
para obtenção do grau de Mestre em Ciência da Com-
putação.*

Orientador: *Prof. Francisco Assis Tenório De Carvalho*

Recife
28 de Agosto de 2009

Macário Filho, Valmir

Um novo algoritmo de agrupamento semi-supervisionado baseado no Fuzzy C-Means / Valmir Macário Filho. - Recife: O autor, 2009.

xvii, 89 páginas: il., fig., tab.

Dissertação (mestrado) - Universidade Federal de Pernambuco. CIN. Ciência da Computação, 2009.

Inclui bibliografia.

1. Inteligência artificial. 2. Sistemas difusos. I. Título.

006.3

CDD (22.ed.)

MEI-2010-022

*Dedico esse trabalho que fortifica meu senso de
pesquisador a minha mãe apoiadora da minha realização
como cientista.*

Agradecimentos

Gostaria de agradecer ao meu pai, Valmir Macário, que tornou possível eu estudar numa cidade como Recife, ofereceu todo o suporte para que eu tivesse uma boa educação e nunca deixou faltar nada na caminhada para o término desse curso.

Agradeço a minha Mãe que apesar de longe me forneceu apoio e amor para que conseguisse passar por todas as dificuldades.

Agradeço a Patrícia Maia pelo incentivo, compreensão e ajuda incondicional durante o período de realização deste trabalho.

Agradeço a meu orientador, Francisco Carvalho, que me orientou sempre que precisei, fornecendo dicas valiosas e necessárias para que esse trabalho fosse possível.

Agradeço a Manuela Souza que apareceu no momento final desse trabalho e me ajudou na medida do possível na confecção de figuras e gráficos presentes nesse trabalho.

Agradeço a Thaís Gaudêncio pelas correções valiosas em alguns capítulos desse trabalho.

Por fim agradeço a todos que me ajudaram num menor ou maior grau.

Muito obrigado a Deus, que guia minha vida!

Resumo

Nas aplicações tradicionais de aprendizagem de máquina, os classificadores utilizam apenas dados rotulados em seu treinamento. Os dados rotulados, por sua vez, são difíceis, caros, consomem tempo e requerem especialistas humanos para serem obtidos em algumas aplicações reais. Entretanto, dados não rotulados são abundantes e fáceis de serem obtidos mas há poucas abordagens que os utilizam no treinamento. Para contornar esse problema existe a aprendizagem semi-supervisionada.

A aprendizagem semi-supervisionada utiliza uma grande quantidade de dados não rotulados, juntamente com dados rotulados, com a finalidade de construir classificadores melhores. A abordagem semi-supervisionada obtém resultados melhores do que se utilizassem apenas poucos padrões rotulados em uma abordagem supervisionada ou se utilizassem apenas padrões não rotulados numa abordagem não supervisionada. O algoritmo semi-supervisionado pode ser uma extensão de um algoritmo não supervisionado. Um algoritmo desse tipo pode se basear em algoritmos de agrupamento não supervisionado, adicionando-se um termo em sua função objetivo que faz uso de informações rotuladas para guiar o processo de aprendizagem do algoritmo.

Este trabalho apresenta um estudo da aprendizagem semi-supervisionada e apresenta um novo algoritmo de agrupamento semi-supervisionado baseado no algoritmo *Fuzzy C-Means*. Também, apresenta uma validação cruzada para o contexto de algoritmos semi-supervisionados. Estudos experimentais são apresentados. Primeiro, o algoritmo semi-supervisionado proposto é avaliado com dados completamente rotulados, comparado com alguns classificadores totalmente supervisionados. Depois, o mesmo algoritmo semi-supervisionado é, então, avaliado e comparado com três algoritmos também de agrupamento semi-supervisionados que otimizam uma função objetivo no contexto da aprendizagem a partir de dados parcialmente rotulados. Além disso, o comportamento do algoritmo é discutido e os resultados examinados através da construção de intervalos de confiança.

Derivou deste trabalho, uma ferramenta contendo os algoritmos semi-supervisionados e o ambiente experimental para validação desses algoritmos foi desenvolvida. Desse modo, foi possível certificar que o novo algoritmo de agrupamento semi-supervisionado apresenta desempenho melhor, ou pelo menos do mesmo nível, que algoritmos já consolidados na literatura.

Palavras-chave: Aprendizagem Semi-Supervisionada, Agrupamento Semi-Supervisionado, Agrupamento Fuzzy, Função Objetivo, Classificação de Padrões, Validação Cruzada

Abstract

In traditional machine learning applications, one uses only labeled data to train the classifier. Labeled data are difficult, expensive, time consuming and require human experts to be obtained in some real applications. However, unlabeled data are abundant and easy to be obtained but there has been few approaches to use them in training. Semi-supervised learning address this problem.

The semi-supervised learning uses large amount of unlabeled data, together with the labeled data, to build better classifiers. The semi-supervised approach obtains better results than if using a few labeled patterns in a supervised approach or using only standard not supervised approach. The semi-supervised algorithm can be an extension of an unsupervised algorithm. Such algorithm can be based on unsupervised clustering algorithms, adding a term in its objective function that makes use of labeled information to guide the learning process of the algorithm.

This work presents a semi-supervised learning study and presents a new algorithm for semi-supervised clustering based on *Fuzzy C-Means* algorithm. Comprehensive experimental studies are presented. First, the induced classifier is evaluated on completely labeled data and validated by comparison against some fully supervised classifiers. This classifier is then evaluated and compared against three semi-supervised clustering algorithms in the context of learning from partly labeled data. In addition, the behavior of the algorithm is discussed and the results are investigated using confidence interval.

As a result, a tool containing the semi-supervised algorithms and the experimental environment for evaluate these algorithms was developed. Thus, it was possible certify that the performance of the new semi-supervised clustering algorithm is better, or at least, is in the same level, of consolidated algorithms.

Keywords: Semi-Supervised Learning, Semi-Supervised Clustering, Fuzzy Clustering, Objective Function, Pattern Classification, Cross-Validation

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Objetivos	2
1.3	Estrutura da dissertação	3
2	Aprendizagem Semi-Supervisionada	5
2.1	Classificação	5
2.2	Agrupamento	5
2.3	Aprendizagem Semi-Supervisionada	6
2.3.1	Classificação Semi-Supervisionada	7
2.3.2	Agrupamento Semi-Supervisionado	10
2.4	Questões na aprendizagem Semi-Supervisionada	11
2.4.1	Dados rotulados trazem benefício ou prejuízo à aprendizagem de máquina?	11
2.4.2	Humanos realizam aprendizagem Semi-Supervisionada?	12
2.5	Considerações Finais	13
3	Agrupamento Semi-Supervisionado	15
3.1	Agrupamento	15
3.1.1	Representação de Objetos do Mundo Real	16
3.1.2	Distâncias e Similaridades	17
3.1.3	Centróide, Partição e Classe e Grupo	17
3.1.4	Agrupamento <i>Hard</i> e Agrupamento <i>Fuzzy</i>	17
3.1.5	Processo de Agrupamento	18
3.2	Algoritmo de Agrupamento <i>Fuzzy C-Means</i>	21
3.3	Algoritmos de Agrupamento Semi-Supervisionado	22
3.3.1	<i>Algoritmo de Pedrycz</i>	23
3.3.2	Algoritmo de Bouchachia	24
3.3.3	Algoritmo Baseado em Sementes	27
3.3.4	Algoritmo de Agrupamento Semi-Supervisionado Proposto	28
3.3.5	Parâmetros Importantes nos Algoritmos de Agrupamento	29
3.3.5.1	Normalização da Base de Dados	30
3.3.5.2	Inicialização da Matriz de Protótipos	30
3.3.5.3	Inicialização da Matriz de Grau de Pertinência	30
3.3.5.4	Critério de Parada	31
3.3.6	Resumo dos Algoritmos de Agrupamento Semi-Supervisionados	32

4	Validação Experimental	35
4.1	Metodologia de Validação	35
4.1.1	Validação Cruzada	35
4.1.2	Taxa de Acerto	37
4.1.3	Índice Externo - Índice de Rand Corrigido	38
4.1.4	Intervalo de Confiança	39
4.2	Bases de Dados	41
4.2.1	Base de Dados Iris	41
4.2.2	Base de Dados Diabetes	41
4.2.3	Base de Dados Wine	41
4.2.4	Base de Dados Spam	41
4.2.5	Base de Dados Sintética	42
4.3	Experimentos	42
4.3.1	Experimento 1: Classificação	43
4.3.2	Agrupamento Semi-Supervisionado	45
4.3.2.1	Experimento 2: Tarefa de Classificação Semi-Supervisionada	47
4.3.2.2	Experimento 3: Tarefa de Agrupamento Semi-Supervisionada	47
5	Resultados e Discussão	49
5.1	Do Agrupamento a Classificação	49
5.1.1	Discussão	51
5.2	Resultados da Tarefa de Classificação Semi-Supervisionada	52
5.2.1	Resultados para a base de dados Iris	52
5.2.2	Resultados para a base de dados Diabetes	54
5.2.3	Resultados para a base de dados <i>Wine</i>	55
5.2.4	Resultados para a base de dados Sintética	56
5.2.5	Discussão	58
5.3	Resultados da Tarefa de Agrupamento Semi-Supervisionado	60
5.3.1	Resultados para a base de dados Iris	60
5.3.2	Resultados para a base de dados Diabetes	64
5.3.3	Resultados para a base de dados Wine	66
5.3.4	Resultados para a base de dados Sintética	70
5.3.5	Resultados para a base de dados <i>Spam</i>	74
5.3.6	Discussão	77
6	Conclusões e Trabalhos Futuros	81
6.1	Conclusão	81
6.2	Discussão	82
6.3	Trabalhos Futuros	82

Lista de Figuras

2.1	A aprendizagem semi-supervisionada é uma abordagem intermediária entre as aprendizagens não supervisionada e supervisionada.	7
2.2	Nesta base de dados, a aprendizagem não supervisionada encontra dois grupos (representados pelas elipses horizontal e vertical em volta dos exemplos). Os dados parcialmente rotulados mostram que apenas uma dessas representações está correta. Esta informação é utilizada na aprendizagem semi-supervisionada.	7
2.3	TSVM	10
3.1	Agrupamentos em duas dimensões [JD88]	16
3.2	Grupos <i>versus</i> classes	18
3.3	<i>Fuzzy Versus Hard</i>	19
3.4	Processo Geral de Rotulação por Algoritmos de Agrupamento	20
4.1	Intervalos de Confiança (95%)	40
4.2	Base de dados Sintética (Pontos em azul: Classe 1; Pontos em vermelho: Classe 2)	42
4.3	Usando o algoritmo de agrupamento como um classificador fuzzy	43
4.4	Processo de Rotulação por Agrupamento	47
5.1	Estudo Comparativo da Classificação: Semi-Supervisionado x MLP x Bayes x Part	50
5.2	Intervalos de Confiança Para o Comparativo do Algoritmo Semi-Supervisionado Proposto com Algoritmos Totalmente Supervisionados	51
5.3	Validação Cruzada de Algoritmos de Agrupamento Semi-Supervisionado: Base de Dados Iris	52
5.4	Intervalos de Confiança para a Base de Dados Iris Utilizando Validação Cruzada com Inicialização <i>Hard</i>	53
5.5	Validação Cruzada de Algoritmos De Agrupamento Semi-Supervisionado: Base de Dados Diabetes	54
5.6	Intervalos de Confiança para a Base de Dados Diabetes Utilizando Validação Cruzada com Inicialização <i>Hard</i>	55
5.7	Validação Cruzada de Algoritmos De Agrupamento Semi-Supervisionado: Base de Dados Wine	56
5.8	Intervalos de Confiança para a Base de Dados <i>Wine</i> Utilizando Validação Cruzada com Inicialização <i>Hard</i>	57

5.9	Validação Cruzada de Algoritmos De Agrupamento Semi-Supervisionado: Base de Dados Sintética	58
5.10	Intervalos de Confiança para a Base de Dados Sintética Utilizando Validação Cruzada com Inicialização <i>Hard</i>	59
5.11	Estudo Comparativo: Base de Dados Iris	61
5.12	Estudo Comparativo do Índice de Rand Corrigido Global: Base de Dados Iris	61
5.13	Estudo Comparativo do Índice de Rand Corrigido Rotulado: Base de Dados Iris	62
5.14	Estudo Comparativo do Índice de Rand Corrigido Não Rotulado: Base de Dados Iris	62
5.15	Intervalos de Confiança para a Base de Dados Iris na Tarefa de Agrupamento	63
5.16	Estudo Comparativo: Base de Dados Diabetes	64
5.17	Estudo Comparativo do Índice de Rand Corrigido Global: Base de Dados Diabetes	65
5.18	Estudo Comparativo do Índice de Rand Corrigido Rotulado: Base de Dados Diabetes	66
5.19	Estudo Comparativo do Índice de Rand Corrigido Não Rotulado: Base de Dados Diabetes	66
5.20	Intervalos de Confiança para a Base de Dados Diabetes na Tarefa de Agrupamento	67
5.21	Estudo Comparativo: Base de Dados Wine	68
5.22	Estudo Comparativo do Índice de Rand Corrigido Global: Base de Dados Wine	69
5.23	Estudo Comparativo do Índice de Rand Corrigido Rotulado: Base de Dados Wine	69
5.24	Estudo Comparativo do Índice de Rand Corrigido Não Rotulado: Base de Dados Wine	70
5.25	Intervalos de Confiança para a Base de Dados <i>Wine</i> na Tarefa de Agrupamento	71
5.26	Estudo Comparativo: Base de Dados Sintética	72
5.27	Estudo Comparativo do Índice de Rand Corrigido Global: Base de Dados Sintética	72
5.28	Estudo Comparativo do Índice de Rand Corrigido Rotulado: Base de Dados Sintética	73
5.29	Estudo Comparativo do Índice de Rand Corrigido Não Rotulado: Base de Dados Sintética	73
5.30	Intervalos de Confiança para a Base de Dados Sintética na Tarefa de Agrupamento	74
5.31	Estudo Comparativo: Base de Dados Spam	75
5.32	Estudo Comparativo do Índice de Rand Corrigido Global: Base de Dados Spam	76
5.33	Estudo Comparativo do Índice de Rand Corrigido Rotulado: Base de Dados Spam	76
5.34	Estudo Comparativo do Índice de Rand Corrigido Não Rotulado: Base de Dados Spam	77
5.35	Intervalos de Confiança para a Base de Dados <i>Spam</i> na Tarefa de Agrupamento	78

Lista de Tabelas

3.1	Exemplos no formato atributo-valor	16
4.1	Matriz de Confusão	37
4.2	Tabela de contingência para duas partições	38
4.3	Base de dados Sintética	42
4.4	Parâmetros dos experimentos para MLP	44

CAPÍTULO 1

Introdução

Na medida em que a tecnologia evolui e que mais pessoas têm acesso à mesma, altera-se continuamente a forma de interação, busca e disponibilização da informação. Assim, torna-se evidente a necessidade de construir ferramentas e gerar técnicas capazes de ajudar os seres humanos a extrair conhecimento de forma automática e inteligente dessa enorme quantidade de informação armazenada.

Uma das técnicas amplamente empregada nesta tarefa de extrair conhecimento é a aprendizagem de máquina [Mit97]. A aprendizagem de máquina constrói modelos computacionais que aprendem a extrair conhecimento a partir da análise desses dados. A aprendizagem de máquina pode ser caracterizada pela aprendizagem supervisionada e não supervisionada.

Na primeira, um conjunto conhecido, ou seja, rotulado, de dados é apresentado ao algoritmo. A aprendizagem é realizada a partir da apresentação desses padrões conhecidos. O objetivo é aprender uma função que mapeie o padrão x ao rótulo h a partir do conjunto de treinamento que contém pares (x_i, h_i) , onde o $h_i \in H$ são chamados de rótulos dos exemplos x_i . O objetivo é prever o rótulo h_i dos exemplos que não possuem seus rótulos conhecidos. Existem duas famílias de algoritmos supervisionados [CZS06]: algoritmos generativos que tentam modelar uma densidade condicional por algum procedimento de aprendizagem não supervisionado e os algoritmos discriminativos que determinam o valor da probabilidade de se encontrar x numa determinada classe e não em saber como a distribuição da classe foi gerada como no método anterior.

Na aprendizagem não supervisionada os dados apresentados ao algoritmo são desconhecidos, não rotulados. O objetivo é aprender a estrutura dos dados X apresentados. Esses algoritmos procuram aprender a distribuição dos dados de acordo com alguma medida de similaridade. Portanto, os padrões mais similares entre si são dispostos num mesmo grupo.

1.1 Motivação

A escolha das abordagens clássicas entre as aprendizagens supervisionadas e não supervisionadas se baseia principalmente no aspecto de quanta informação existe acerca dos dados. Escolhe-se a primeira se houver informações descritivas acerca dos dados, ou seja, rótulos que indiquem a classe de um padrão pertencente a base de dados. Assim, os dados rotulados, chamados de conjunto de treinamento, são utilizados para treinar o algoritmo. A aprendizagem não supervisionada é escolhida quando não há informações descritivas dos dados. Os dados são agrupados de acordo com algum critério. Existem diversas técnicas de agrupamento [JMF99]. Uma dessas técnicas realizam a otimização de uma função objetivo. O processo de

agrupamento é definido pela otimização dos parâmetros dessa função objetivo.

A aprendizagem não supervisionada sofre algumas limitações significativas. Diferente do que acontece no processo supervisionado, os resultados de um aprendizado não supervisionado são apenas os grupos ou partições, sem informações descritivas acerca dessas partições geradas. Porém, algumas vezes, os usuários não estão interessados apenas nessas partições, mas também em alguma explicação do que essas partições representam. Para superar essa limitação é necessária uma interpretação das partições encontradas. Essa interpretação não é uma tarefa fácil, pois é forçoso realizar uma tarefa inferencial complexa [San03].

Pode-se então pensar na alternativa de realizar uma marcação manual dos dados para utilização de algoritmos supervisionados. Entretanto, essa tarefa pode ser extremamente complexa, cara, demorada e requerer especialistas humanos em algumas aplicações reais. Podemos citar como exemplo a extração de processos em diários oficiais. Essa tarefa requer que cada linha de um diário oficial seja rotulada, sendo que a publicação de apenas um dia desse documento pode conter mais de 100000 linhas [MPC⁺08]. Também, há os casos de *Spam*, onde o classificador é obrigado a aprender os e-mails não desejados a partir de uma porcentagem pequena de e-mails marcados [CL06]. Ainda podemos citar aplicações na mineração de textos [NMTM00] ou ainda aplicações em processamento de imagens [BHBC96]. Desse modo, a utilização de algoritmos supervisionados é inviável, pois existem poucos dados rotulados disponíveis para o treinamento desse algoritmo. Tais fatores sugerem uma técnica que utilize exemplos não rotulados para aumentar a precisão de um classificador.

A aprendizagem semi-supervisionada [CZS06][ZL05b] é uma abordagem intermediária entre a aprendizagem supervisionada e a aprendizagem não-supervisionada. Na aprendizagem semi-supervisionada são utilizados exemplos rotulados e não rotulados para guiar a aprendizagem de um algoritmo com a finalidade de construir classificadores melhores. Devido as diversas aplicações que podem ser auxiliadas por esse tipo de abordagem, este trabalho possui como objeto de estudo a aprendizagem semi-supervisionada.

1.2 Objetivos

Como mencionado, rotular dados em algumas aplicações reais pode ser bastante caro. Entretanto, é aceitável pedir a um especialista para rotular um pequeno conjunto de exemplos. Nesse caso, pode-se considerar a utilização da aprendizagem semi-supervisionada. Há dois aspectos fundamentais na realização da aprendizagem semi-supervisionada. A primeira, é tentar estimar os rótulos dos dados não rotulados antes de iniciar um algoritmo totalmente supervisionado com esses dados. A segunda é utilizar uma combinação completa de dados rotulados e não rotulados para treinar um algoritmo parcialmente supervisionado [Bou07]. Neste trabalho é proposto um novo algoritmo de agrupamento semi-supervisionado que segue o segundo aspecto.

O algoritmo proposto neste trabalho é baseado no algoritmo de agrupamento *Fuzzy C-Means* [Bez81] com o acréscimo de um segundo termo à função objetivo que é responsável por guiar o processo de agrupamento utilizando informações dos rótulos das classes. Desse modo, o primeiro termo realiza o agrupamento independente dos rótulos, enquanto o segundo é guiado pelos rótulos. Assim, o algoritmo utiliza ambos os tipos de dados em seu treinamento.

Desse modo, as contribuições deste trabalho são:

1. **Estudo da aprendizagem semi-supervisionada:** Um estudo de aspectos relevantes da aprendizagem semi-supervisionada, como estado da arte e principais abordagens.
2. **Estudo de algoritmos de agrupamento semi-supervisionados:** Análise desses tipos de algoritmos.
3. **Implementação e experimentação de um novo algoritmo de agrupamento semi-supervisionado:** Foi proposto um novo algoritmo de agrupamento semi-supervisionado que realiza a otimização de uma função objetivo. O algoritmo é comparado com outros já consolidados na literatura. A validação cruzada utilizada em algoritmos não supervisionados é utilizada no contexto de algoritmos semi-supervisionados. Depois, são construídos intervalos de confiança para investigar os experimentos realizados.

1.3 Estrutura da dissertação

Esta dissertação está dividida em 6 capítulos no total. Além deste capítulo de introdução, essa dissertação possui 5 capítulos organizados como segue:

Capítulo 2 - Aprendizagem Semi-Supervisionada: Os fundamentos da aprendizagem semi supervisionada são apresentados neste capítulo.

Capítulo 3 - Agrupamento Semi-Supervisionado: Este capítulo apresenta o conceito de algoritmos de agrupamento não supervisionados e de algoritmos de agrupamento semi-supervisionados . Além disso descreve algoritmos de agrupamento semi-supervisionado consolidados na literatura e um novo algoritmo deste tipo proposto por este trabalho.

Capítulo 4 - Validação Experimental: Este capítulo descreve a metodologia dos experimentos. O ambiente, as técnicas utilizadas, bem como as configurações dos experimentos são explicadas. Além disso, são apresentadas todas as bases de dados utilizadas na validação desse trabalho.

Capítulo 5 - Resultados e Discussão: Os resultados dos experimentos de comparação do algoritmo proposto com outros já descritos na literatura são apresentados neste capítulo sendo os experimentos discutidos ao final de cada seção.

Capítulo 6 - Conclusões e Trabalhos Futuros: As conclusões deste trabalho, assim como nossas perspectivas de pesquisas futuras a partir das idéias aqui apresentadas estão presentes neste Capítulo.

Aprendizagem Semi-Supervisionada

Neste capítulo será descrita resumidamente a tarefa de classificação e de agrupamento realizadas pelas aprendizagens supervisionada e não supervisionada. Depois será explorado o conceito e o estado da arte dos algoritmos semi-supervisionados, desde o surgimento desse tipo de abordagem até os dias atuais.

2.1 Classificação

Classificação é uma tarefa, por definição, supervisionada. A supervisão é realizada através de um conjunto de dados de treinamento rotulados, ou seja, cada exemplo do conjunto de treinamento possui um rótulo que indica a classe que o exemplo pertence [Mit97]. O objetivo da classificação é aprender uma função através dos dados de treinamento que construa a melhor predição dos rótulos das classes de dados nunca vistos.

Existem dois tipos de algoritmos supervisionados: generativos e discriminativos. Algoritmos generativos tentam modelar uma densidade condicional ($p(x|y)$) por algum procedimento para descobrir como os dados são distribuídos. Formalmente, o modelo generativo assume que $p(x, y) = p(y)p(x|y)$. O parâmetro $p(x|y)$ é uma mistura de distribuição identificável [RV95], como por exemplo, uma mistura de modelos Gaussianos e o parâmetro $p(y)$ é uma probabilidade conhecida previamente. O resultado $p(x, y)$ é a probabilidade de um exemplo x pertencer a uma classe de rótulo y . Sabendo como a distribuição foi gerada, dados do conjunto de testes, ou dados não vistos pelo algoritmo, são rotulados através da mesma distribuição que gerou os dados de treinamento. Algoritmos discriminativos concentram-se em determinar uma função discriminante para as fronteiras das classes ou de uma probabilidade de encontrar o exemplo x numa determinada classe y ao invés de aprender como a distribuição da classe foi gerada, como acontece no método anterior. Algoritmos desse tipo possuem uma generalização maior sob certas suposições e geram classificadores clássicos como a rede neural SVM [Vap98].

2.2 Agrupamento

Agrupamento é uma tarefa não supervisionada que reúne um conjunto de dados num grupo onde esses dados são mais similares entre si do que entre dados reunidos em diferentes grupos [JD88]. Os dados apresentados aos algoritmos não supervisionados não possuem a informação dos rótulos da classe (categoria). Levam em conta apenas as descrições dos exemplos para realizar a tarefa de agrupamento. As descrições dos exemplos são vetores contendo características descritivas de cada exemplo. Nesta seção daremos apenas uma breve introdução sobre a

tarefa de agrupamento, no próximo capítulo abordaremos com mais detalhes essa tarefa.

Agrupamento também pode ser categorizado como generativo e discriminativo. O modelo generativo assume uma distribuição paramétrica de dados. Assim, é realizada a maximização da verossimilhança que possui como objetivo encontrar os parâmetros que maximizem a probabilidade de os dados terem sido gerados pelo modelo avaliado. Por exemplo, numa distribuição gaussiana, temos que escolher a média e a covariância que gerou a distribuição dos dados. Na formulação mais geral, o número de grupos k também é considerado um parâmetro desconhecido. O valor de k é escolhido tal que o modelo de agrupamento melhor se ajuste aos dados. No modelo de agrupamento discriminativo, o algoritmo tenta agrupar os dados de forma a maximizar a similaridade dentro de cada grupo e minimizar a similaridade entre os grupos, com base numa matriz de similaridade definida sobre os dados do conjunto de entrada. Neste paradigma, não é necessário considerar um modelo de geração de base de dados paramétricos. Em ambos, modelos generativos e discriminativo, os algoritmos de agrupamento são geralmente colocados como problemas de otimização e resolvidos por métodos iterativos como EM [DLR77] e o *Fuzzy C-Means* [Bez81].

2.3 Aprendizagem Semi-Supervisionada

Os algoritmos semi-supervisionados são uma abordagem intermediária entre os algoritmos de aprendizagem supervisionada e não supervisionada. O detalhe crucial de um algoritmo semi-supervisionado é que o mesmo seja capaz de aprender a partir de dados rotulados e não rotulados para realizar uma tarefa de classificação ou de agrupamento. Os algoritmos semi-supervisionados podem realizar os dois tipos de tarefa, pois ele pode aprender a partir dos dois tipos de dados (rotulados e não rotulados).

Para ilustrar a aprendizagem semi-supervisionada, a Figura 2.1 apresenta às diferenças entre os conjuntos de treinamento entre os diferentes paradigmas de aprendizagem. A Figura 2.1(a) apresenta os dados não rotulados. Esses dados não possuem informação acerca do rótulo, então não é possível saber a qual classe cada um dos exemplos pertence. A Figura 2.1(b) representa o conjunto de treinamento rotulado. O conjunto de treinamento é formado por duas classes, uma delas representada pelas estrelas vermelhas e a outra pelo sinal de positivo verde. Dessa forma, a aprendizagem supervisionada é guiada pelo conjunto de treinamento totalmente rotulado. Por último, temos a Figura 2.1(c) que representa a aprendizagem semi-supervisionada. Note que no conjunto de treinamento existe dados não rotulados, representados pelos triângulos e dados rotulados, representados pela estrela vermelha e pelo sinal positivo verde. Dessa forma, o treinamento semi-supervisionado é guiado pelos dois tipos de dados, sendo guiados pelas informações disponíveis da descrição dos dados não rotulados e também pelo rótulo dos dados rotulados.

Um exemplo de como a informação rotulada pode ser útil é apresentada na Figura 2.2. Nela há dois grupos representando duas classes. Se não houvesse informação nenhuma acerca de alguns exemplos do conjunto de dados os grupos seriam colocados em classes erradas, pois o algoritmo poderia agrupar os grupos verticalmente como mostrado na Figura 2.2(a). Mas na Figura 2.2(b) existe informação de rótulos de alguns exemplos da base de dados, desse modo o algoritmo agrupa corretamente os exemplos, colocando-os dentro da representação correta.

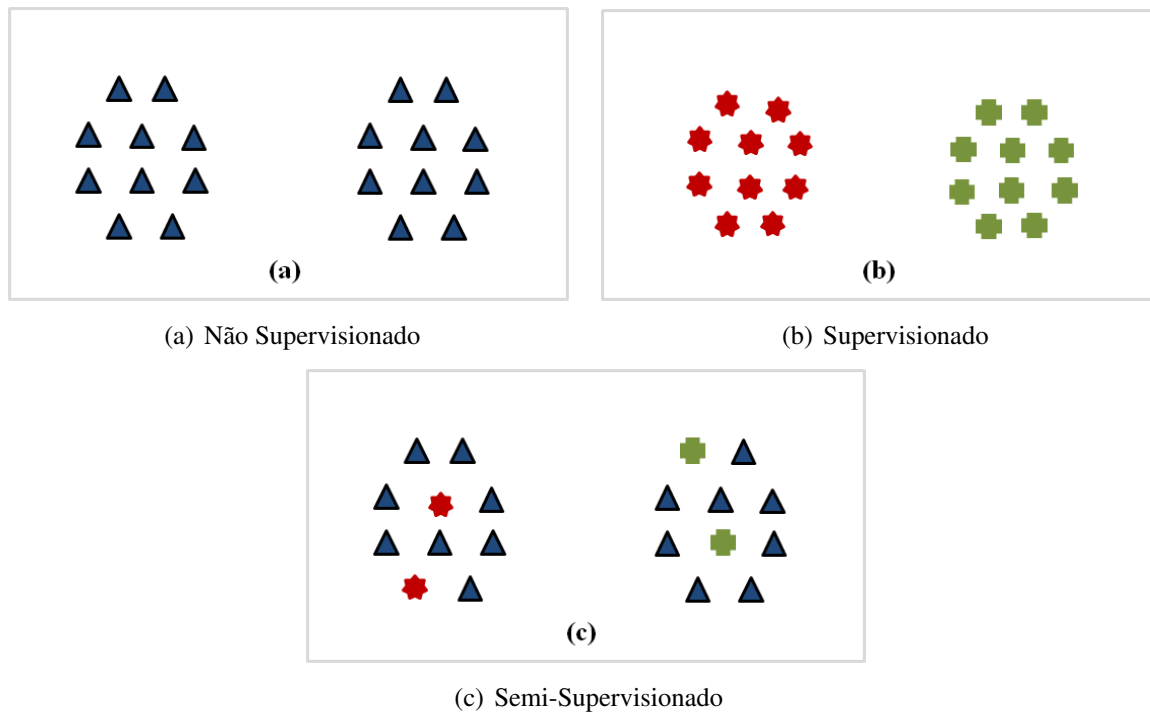


Figura 2.1 A aprendizagem semi-supervisionada é uma abordagem intermediária entre as aprendizagens não supervisionada e supervisionada.

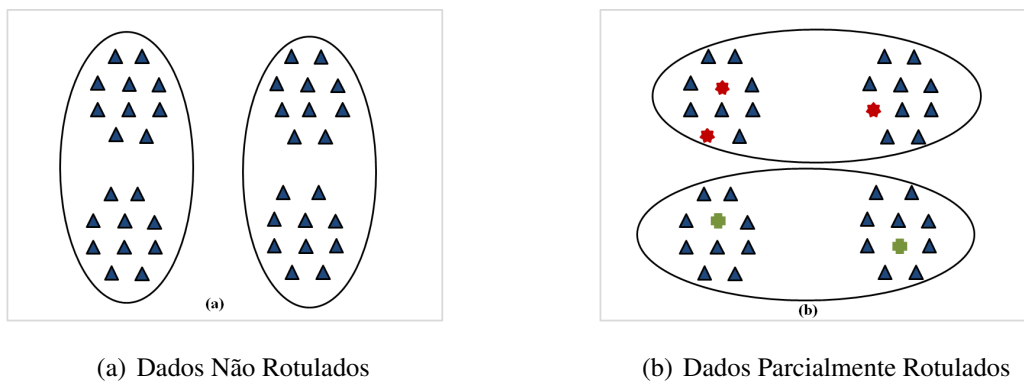


Figura 2.2 Nesta base de dados, a aprendizagem não supervisionada encontra dois grupos (representados pelas elipses horizontal e vertical em volta dos exemplos). Os dados parcialmente rotulados mostram que apenas uma dessas representações está correta. Esta informação é utilizada na aprendizagem semi-supervisionada.

2.3.1 Classificação Semi-Supervisionada

A classificação supervisionada possui um número fixo de classes ou categorias previamente conhecidas e ainda, um conjunto de treinamento totalmente rotulado para que o classificador

aprenda um modelo capaz de marcar dados nunca vistos antes pelo classificador. Na classificação semi-supervisionada dados não rotulados são acrescentados ao conjunto de treinamento para aumentar a eficiência do classificador.

As primeiras idéias que utilizaram dados não rotulados para melhorar o treinamento de algoritmos surgiram na comunidade estatística. As primeiras técnicas eram conhecidas por outros nomes: técnica de auto-aprendizagem, auto-rotulação e treinamento de decisão direta [Scu65][Fra67]. No auto treinamento, o algoritmo é treinado com um pequeno grupo de dados rotulados. O classificador treinado, classifica o montante restante de dados não rotulados. Os dados que conseguirem uma alta confiança na classificação são adicionados ao conjunto de treinamento. Desse modo, o algoritmo é treinado novamente com esse novo conjunto de treinamento de dados rotulados previamente e dados rotulados pelo algoritmo. Esse procedimento é repetido até que os dados sejam classificados em sua maioria com uma alta confiança. Assim, o classificador ensina a si mesmo, com suas próprias previsões. Esse procedimento é chamado de auto treinamento ou *bootstrapping* (não é o mesmo que o utilizado na estatística). O procedimento de auto treinamento tem sido aplicado principalmente em tarefas de processamento de linguagem natural. Yarowsky [Yar95] utiliza esse procedimento para desambiguação do sentido de palavras, por exemplo, decidir se o termo vegetais, seria um organismo vivo ou fabricado, num dado contexto. Riloff et al. [RWW03] utiliza para identificar substantivos subjetivos. Maeireizo et al. [MLH04] classifica diálogos como "emocional" ou "não-emocional" com um procedimento que envolve dois classificadores auto treinados. Também tem sido aplicado à análise e tradução automática. Rosenberg et al. [RH05] aplica esse procedimento para sistemas de detecção de imagens de objetos. O auto treinamento é bastante aplicado por ser o método semi-supervisionado mais simples. Esse método é aplicado a classificadores já existentes, apenas mudando a forma de treinamento desses classificadores. Apesar disso, este método apresenta desvantagens como o fato de erros ocorridos previamente serem reforçados pelo treinamento além de possuir pouco conhecimento sobre a sua convergência.

Após a década de 70 houve um desinteresse pelas técnicas semi-supervisionadas. O interesse voltou após a década de 90, quando problemas de linguagem natural e classificação de texto apareceram mais fortemente. Nessa mesma década, uma abordagem eficiente foi utilizada por Blum e Mitchel [BM98][Mit99]. O *co-training* assume que as características podem ser divididas em dois subconjuntos. Cada subconjunto dessas características é suficiente para treinar um bom classificador. Dado a classe, esses conjuntos são condicionalmente independentes. Assim, dois classificadores são treinados com os dados rotulados, utilizando os dois subconjuntos. Dessa forma, os classificadores usam os dados não rotulados e ensinam o outro classificador com alguns dos exemplos não rotulados que foram classificados com uma confiança alta. Continuando, cada classificador é re-treinado com dados adicionais resultantes de outro classificador. Desse modo o processo se repete até que todos os dados sejam rotulados com um valor alto de confiança. Na abordagem *co-training*, os dois classificadores (ou hipóteses) devem concordar com a grande maioria dos dados não rotulados tão bem quanto os dados rotulados. As sub-características tem que ser condicionalmente independentes para que um dos dados de um classificador com alta confiança sejam boas amostras para o outro classificador. O trabalho de Nigam e Ghani [NMTM00] realiza experiências empíricas extensivas para comparar formações de *co-training* de mistura de modelos generativos e o algoritmo

EM [DLR77]. O seu resultado mostra que o *co-training* conseguiu obter um bom desempenho quando a hipótese de independência condicional realmente existe. Além disso, é melhor rotular todos os dados não rotulados, ao invés de rotular apenas uma pequena parte. Os autores deram o nome desse paradigma co-EM.

Outra idéia relacionada à aprendizagem semi-supervisionada é o conceito de transdução criado por Vapnik [VC74]. Na transdução, ao invés de uma regra de decisão ser construída para modelar a distribuição dos dados, apenas os rótulos dos dados não rotulados são previstos (modelo discriminativo). Um exemplo de uma tarefa transdutiva é aprender à relevância da experiência do usuário na recuperação da informação. Nessa tarefa, o usuário marca alguns documentos retornados por um engenho de busca como relevantes ou não relevantes. Esses documentos servem para treinar um classificador de textos binário. Esse problema pode ser visto como um problema de classificação supervisionada mas possui duas características que podem ter um ponto de vista diferente. Em primeiro lugar, o algoritmo não tem que necessariamente aprender uma regra geral, só precisa prever a acurácia para um conjunto finito de exemplos de teste (os documentos no bando de dados). Em segundo lugar, o conjunto de teste pode ser conhecido previamente e observado durante o treinamento. A aprendizagem transdutiva pode explorar os dados não rotulados no conjunto de teste. Então, o algoritmo tenta prever o rótulo dos dados no conjunto de testes explorando esses conjuntos juntamente com o conjunto de treinamento, minimizando o erro dessas previsões.

O algoritmo de rede neural Máquina de Vetor de Suporte supervisionado [Vap98] é um algoritmo discriminativo que atua diretamente fazendo previsões sobre $p(y|x)$. Essa abordagem deixa de lado o parâmetro generativo $p(x)$, que é usualmente conseguido através dos dados não rotulados. O algoritmo semi-supervisionado Máquina de Vetor de Suporte Transdutivo (TSVM) [Joa99] é uma extensão do algoritmo padrão SVM com a característica de utilizar dados não rotulados em seu treinamento. No SVM padrão apenas, dados rotulados são utilizados, e o objetivo é encontrar uma margem linear máxima na fronteira das classes reproduzindo o espaço do núcleo de Hilbert. O algoritmo TSVM constrói a relação entre $p(x)$ e a decisão discriminativa de fronteira. Não discrimina o limite dessas fronteiras em regiões de alta densidade e possui como objetivo encontrar os rótulos dos dados não marcados de forma que a margem da fronteira linear seja máxima para ambos os dados. A fronteira de decisão possui a menor generalização de erro sobre os dados não marcados [Vap98] e guia a fronteira linear para longe de regiões densas. O TSVM pode ser visto como um SVM padrão com a adição de um termo que regulariza o termo original utilizando dados não rotulados. A Figura 2.3 apresenta um exemplo de separação de duas classes utilizando padrões rotulados (em vermelho) e não rotulados. As vantagens dessa abordagem é que esta pode ser utilizada em qualquer problema onde a rede SVM padrão é aplicada e a matemática dessa abordagem é amplamente estudada e prova que esse método pode convergir para bons resultados. Porém, as desvantagens estão justamente nos pontos mais estudados dessa abordagem. Sua otimização é difícil e o resultado encontrado pode ser um máximo local ruim.

Nesta seção foram apresentadas algumas abordagens mais conhecidas de classificação semi-supervisionada. Outros métodos de classificação semi-supervisionada podem ser encontrados no trabalho de Seeger [See00] e no estudo de Zhu [Zhu08].

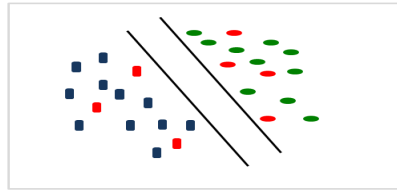


Figura 2.3 TSVM

2.3.2 Agrupamento Semi-Supervisionado

Este tópico se refere ao método estudado nesse trabalho. Os algoritmos de agrupamento semi-supervisionados adicionam um termo supervisionado a função objetivo não supervisionada do algoritmo *Fuzzy C-Means*. Assim, a função objetivo de um algoritmo de agrupamento semi-supervisionado é composta por pelo menos dois termos. O primeiro termo é o termo não supervisionado herdado do algoritmo de agrupamento não supervisionado *Fuzzy C-Means* e o segundo termo é um termo supervisionado que utiliza informações rotuladas para guiar o aprendizado do algoritmo.

O trabalho de Pedrycz [Ped85] foi um dos primeiros a adaptar a função objetivo do algoritmo *Fuzzy C-Means*, mostrando que os dados não rotulados utilizados em conjunto com dados rotulados melhoraram a qualidade dos grupos formados pelo algoritmo. Seguindo essa linha, Bouchachia e Pedrycz [BP06] adicionaram um termo com duas otimizações na função objetivo do algoritmo *Fuzzy C-Means*. Os autores testaram ainda esse algoritmo como sendo um classificador supervisionado, utilizando apenas dados rotulados no treinamento e obtiveram resultados melhores do que os algoritmos clássicos como a rede neural SVM [Vap98]. Mais tarde, Bouchachia [Bou07] adaptou outro algoritmo não supervisionado que utiliza em sua otimização apenas os dados não rotulados vizinhos e apresentou um ganho no desempenho na formação dos grupos.

Além dos algoritmos que fazem mudanças na função objetivo para que a informação dos dados não rotulados seja mais um parâmetro para sua otimização, há algoritmos que utilizam a informação dos dados não rotulados de outra maneira, geralmente na fase de treinamento desses algoritmos. Diferente dos algoritmos anteriores, o algoritmo adaptado é o clássico k-means [Mac67]. O algoritmo COP-k-means [WCRS01a] utiliza um conhecimento prévio de exemplos que devem estar num mesmo grupo (*must-link*) e de exemplos que não devem estar num mesmo grupo (*cannot-link*). Esse conhecimento é retirado de dados rotulados. O algoritmo k-means_{ki} [San03] seleciona os centróides iniciais nos dados rotulados. Difere do k-means original, no processo de agrupamento, porque ao invés de um dado ser associado ao centróide mais próximo, no k-means_{ki} o dado é associado a um grupo caso a distância seja menor ou igual a um limiar associado ao centróide. Ele também difere na formação dos grupos,

pois os centróides podem ser compostos apenas por padrões rotulados, enquanto no k-means original, todos os exemplos podem ser centróides.

Pode-se perceber que existem diferentes maneiras de utilizar a informação supervisionada nos algoritmos de agrupamento. Neste trabalho daremos atenção a informação dos rótulos de exemplos marcados na otimização de uma função objetivo baseada na função original do algoritmo clássico *Fuzzy C-Means*. No próximo capítulo esta abordagem será apresentada detalhadamente, mostrando a teoria e os conceitos desse tipo de abordagem. Além disso serão apresentados algoritmos de agrupamento semi-supervisionado consolidados na literatura além de um novo algoritmo de agrupamento semi-supervisionado proposto neste trabalho.

2.4 Questões na aprendizagem Semi-Supervisionada

Nesta seção final, respondemos a questões um pouco dissociadas da técnica de aprendizagem semi-supervisionada, porém, totalmente inclusas no porque utilizar uma aprendizagem deste tipo no pensamento lógico humano. Esta seção apresenta duas questões importantes. Dados rotulados realmente melhoram um algoritmo de aprendizagem de máquina? Seres humanos realizam aprendizagem semi-supervisionada? Essas questões são respondidas nas seções posteriores.

2.4.1 Dados rotulados trazem benefício ou prejuízo à aprendizagem de máquina?

A primeira impressão é que aprendizagem semi-supervisionada, utilizando dados rotulados e não rotulados na aprendizagem de um algoritmo, parece ser realmente benéfica no seu desempenho como classificador. Porém, algumas premissas têm que ser levadas em conta para que isso realmente aconteça. A maioria dos trabalhos argumentam em favor do uso de exemplos não rotulados na tarefa de classificação [CZS06][Zhu08], porém também existem trabalhos que mostram que esses dados podem trazer prejuízo ao desempenho do classificador [CCC03]. Um estudo sobre a degradação de desempenho na utilização de dados não rotulados no treinamento de um classificador pode ser encontrado em Cozman e Cohen [CC06].

A principal vantagem na utilização de dados não rotulados na aprendizagem é quando não existem dados rotulados suficientes para treinar um bom classificador. Quando a quantidade de dados rotulados já é suficiente para treinar um classificador com uma boa acurácia os dados não rotulados podem atrapalhar o desempenho do algoritmo, ou em menor grau, não ajudar a melhorar o desempenho do classificador. Assim, a melhor escolha quando há dados rotulados suficientes é utilizar algoritmos totalmente supervisionados ou utilizar apenas o termo supervisionado dos algoritmos semi-supervisionados, quando existe essa característica no algoritmo. Essa característica pode ser observada nos algoritmos de agrupamento explorados nesta dissertação.

No caso de algoritmos generativos, a hipótese do modelo da distribuição da base de dados deve ser correta. Por exemplo, caso os dados estejam construídos sobre uma mistura de gaussianas, o aprendizado desse modelo deve ser uma mistura de gaussiana. Se a hipótese do modelo tiver correta, provavelmente os dados não rotulados vão melhorar a eficácia do classificador. Quando as hipóteses sobre o modelo de distribuição da base de dados está incorreto, provavel-

mente o acréscimo de dados não rotulados no treinamento vão atrapalhar o desempenho do classificador. O problema de prever um modelo incorreto não é específico da aprendizagem semi-supervisionada, é fácil perceber que praticamente qualquer algoritmo treinado com um modelo incorreto de distribuição de dados não terá um desempenho satisfatório. Nesse caso, é melhor escolher um algoritmo discriminativo que não necessita da distribuição dos dados para obter um bom desempenho e possui uma generalização melhor que algoritmos generativos.

Desse modo, podemos afirmar que a utilização de dados não rotulados melhoram o desempenho na maioria dos casos onde dados rotulados sejam caros de obter, e por conseguinte, existam poucos deles. Trabalhos de diversos autores têm mostrado que os dados não rotulados melhoram o desempenho dos classificadores, existindo diversas alternativas na utilização desses dados [Zhu08]. Assim, se dados não rotulados utilizados em conjunto com dados rotulados no treinamento de algoritmos de aprendizagem de máquina melhoram significativamente o desempenho desses algoritmos somos levados ao questionamento final deste capítulo mas básico na motivação inicial da criação da inteligência artificial. Seres humanos realizam aprendizagem semi-supervisionada? Vamos responder esta questão na próxima sub-seção.

2.4.2 Humanos realizam aprendizagem Semi-Supervisionada?

Essa questão está relacionada ao objetivo da criação da aprendizagem de máquina. A aprendizagem de máquina inicialmente foi criada com o objetivo de reproduzir com os computadores, as diversas formas de cognição humana [Lan06]. Juntamente com a psicologia, os primeiros trabalhos estavam interessados em 4 principais objetivos: estruturar e organizar a forma do pensamento humano, resolver problemas humanos, pesquisar o raciocínio e a tomada de decisões e aprender como os humanos aprendem.

Na modelagem da aprendizagem humana, os primeiros sistemas reproduziam o modo incremental da aprendizagem humana e eram combinadas com conhecimentos base aliado com a experiência. O propósito era produzir taxas semelhantes às encontradas nas pessoas. Nos anos 90, os trabalhos nessa área foram gradualmente reduzidos e concentraram-se praticamente na indução supervisionada para classificação e reforço do controle reativo da aprendizagem. Essa mudança foi acompanhada por uma ênfase crescente em métodos estatísticos que exigem grandes quantidades de dados e aprendem muito mais lentamente do que os seres humanos [Lan06]. Também há a aprendizagem não supervisionada, aprendizagem pela observação, sem supervisão ou reforço, de situações desconhecidas. A habilidade que permite o ser humano aprender sem supervisão é a categorização, conhecida como aprendizagem não supervisionada, aprendizagem pela observação, sem supervisão ou reforço, de situações desconhecidas. Categorias permitem generalizar o conhecimento de novas situações e inferir propriedades do ambiente [GL03]. A aprendizagem não supervisionada também teve uma grande concentração de estudos na mesma época, com ênfase nos algoritmos de agrupamento. Um estudo realizado por Love [Lov02] compara situações cognitivas das aprendizagens supervisionadas e não supervisionadas por seres humanos.

O caso da aprendizagem semi-supervisionada é particular, pois mistura os dois tipos de aprendizagens citadas anteriormente. De acordo com Zhu [ZRQK07], os seres humanos acumulam informações não rotuladas para quando tiverem acesso a alguma informação rotulada disponível, realizarem a conexão entre os rótulos.

Para ilustrar a aprendizagem semi-supervisionada por humanos, podemos citar algumas tarefas que são realizadas pelos mesmos, utilizando dados rotulados e não rotulados. A primeira, é a tarefa de discriminação visual [FEP95], onde as pessoas realizam a discriminações de diferentes objetos tendo acesso visual a apenas um pequeno conjunto de exemplos desses. Também, há um trabalho que mostra que na fase que começamos a mapear significados as palavras que são ouvidas. O trabalho de Estes [EEAS06] mostra que crianças de 17 meses conseguem associar palavras a objetos visuais melhor se tiveram ouvido o nome desses objetos diversas vezes antes. Nesse caso, o som da palavra é o dado não rotulado enquanto o objeto é o rótulo. No último exemplo, é mostrado que no paradigma de aprendizagem entre duas classes, o limite de decisão é determinado por dados rotulados e não rotulados [ZRQK07]. Participantes do experimento são apresentados a formas complexas e adivinhar de qual dos dois estímulos existentes essas formas são geradas. Dados rotulados são as representações de tentativas de adivinhação que há um retorno da resposta do participante, enquanto os dados não rotulados são as tentativas que não geram retorno. Eles mostram que se forem mostrados os mesmos dados rotulados e diferentes dados não rotulados, as pessoas formam diferentes limites de decisão. Desse modo, mostra que dados não rotulados influenciam na decisão de categorização dos seres humanos.

2.5 Considerações Finais

Este capítulo apresentou os conceitos básicos dos dois paradigmas de aprendizagem de máquina clássicos: supervisionado e não supervisionado. Depois, foram apresentados os principais conceitos da aprendizagem semi-supervisionada, que é um paradigma intermediário entre a aprendizagem supervisionada e não supervisionada. Por se tratar de uma aprendizagem intermediária, a aprendizagem semi-supervisionada pode realizar tarefas supervisionadas (classificação) e tarefas não supervisionadas (agrupamento). Por fim, finalizamos o capítulo respondendo questões inerentes a aprendizagem semi-supervisionada, pois se trata de um novo paradigma que utiliza diferentes tipos de dados em seu treinamento e também trata-se de uma abordagem da inteligência artificial, inspirada no comportamento humano de aprendizagem. No próximo capítulo trataremos de um algoritmo específico da aprendizagem semi-supervisionada: a aprendizagem de algoritmos de agrupamento semi-supervisionado.

Agrupamento Semi-Supervisionado

Este capítulo abrange questões relevantes na abordagem de agrupamento semi-supervisionado. Uma pequena introdução ao método de agrupamento é apresentado na Seção 3.1. Após a explicação inicial, na Seção 3.2 o algoritmo de agrupamento clássico cujo todos os outros se baseiam é apresentado. Logo depois, o algoritmo de agrupamento semi-supervisionado proposto e os algoritmos analisados de agrupamento semi-supervisionado nesta dissertação são explorados na Seção 3.3.

3.1 Agrupamento

A tarefa de agrupamento tem sido aplicada em diversos problemas. Para citar alguns, temos a mineração de textos, expressões gênicas, processamento de imagens, entre outras. O agrupamento é a aglomeração de objetos (ou exemplos) em grupos, de modo que os objetos pertencentes ao mesmo grupo sejam mais similares entre si de acordo com alguma medida de similaridade, enquanto os objetos pertencentes a grupos diferentes tenham uma similaridade menor. O objetivo do processo de agrupamento é maximizar a homogeneidade dos objetos de um mesmo grupo enquanto maximiza a heterogeneidade entre objetos de grupos diferentes [SM86]. Segundo Jain e Dubes [JD88], agrupamento é o estudo formal de algoritmos e métodos para agrupar exemplos que não estão rotulados com uma classe correspondente.

De modo simples, agrupamento é a criação de grupos que aglomeram objetos similares num mesmo grupo e objetos não similares em grupos diferentes. Everitt [Eve80] diz que a definição formal de um grupo é inoportuna e resume a definições de grupos como segue:

Definição 1: um grupo é um conjunto de entidades que são similares, e entidades de diferentes grupos não são similares.

Definição 2: Um grupo é uma aglomeração de pontos no espaço de teste de tal forma que a distância entre quaisquer dois pontos em um mesmo grupo é menor que a distância entre qualquer ponto desse grupo a outro ponto não pertencente a ele.

Definição 3: Grupos podem ser descritos como regiões conectadas de um espaço multidimensional contendo uma alta densidade relativa de pontos, separadas de outras regiões por uma região contendo baixa densidade relativa de pontos.

Na Figura 3.1 são ilustrados alguns exemplos de agrupamentos, onde cada exemplo é definido por dois atributos e está representado pelo ponto no espaço bidimensional. Nessa figura, percebe-se a existência de quatro grupos. Porém, pode-se perceber, com uma visão mais local, que pode existir um maior número de grupos.

Aqui, foram apresentadas algumas definições do processo de agrupamento. Nas próximas



Figura 3.1 Agrupamentos em duas dimensões [JD88]

seções serão descritos conceitos encontrados na análise de agrupamento.

3.1.1 Representação de Objetos do Mundo Real

A representação de um objeto no mundo na maioria dos algoritmos de aprendizagem de máquina são descritos por um vetor de atributos. Os exemplos desses objetos podem estar rotulados ou não com o atributo da classe. Uma das maneiras mais utilizadas para representar os exemplos é uma tabela no formato atributo-valor. A Tabela 3.1.1 representa o formato geral de uma tabela atributo-valor de um conjunto com n exemplos $E = E_1, E_2, \dots, E_n$ e m atributos $A = A_1, A_2, \dots, A_m$. Nessa tabela, a linha i refere-se ao i -ésimo $i = (1, 2, \dots, n)$ exemplo E , a coluna j refere-se ao j -ésimo $j = (1, 2, \dots, m)$ atributo X e o valor x_{ij} refere-se ao valor do j -ésimo atributo do exemplo i . Já o valor h_i representa a classe do exemplo i . Essa classe é discreta e pertence ao conjunto dos P possíveis valores de classes prévias h_1, h_2, \dots, h_p . Para a aprendizagem não supervisionada, a última coluna que contém os valores da classe H não existe. Desse modo, a tabela teria apenas os atributos dos exemplos do objeto X . Na aprendizagem semi-supervisionada, a tabela mistura exemplos que possuem o valor da classe H com exemplos que não possuem essa informação.

Tabela 3.1 Exemplos no formato atributo-valor

	X_1	X_2	\dots	X_m	H
E_1	x_{11}	x_{12}	\vdots	x_{1m}	h_1
E_2	x_{21}	x_{22}	\vdots	x_{2m}	h_2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
E_n	x_{n1}	x_{n2}	\vdots	x_{nm}	h_n

3.1.2 Distâncias e Similaridades

O objetivo do agrupamento é justamente agrupar exemplos de acordo com a similaridade entre eles. Desse modo, a similaridade é um conceito importante no processo de agrupamento. Na literatura, medidas de similaridade, coeficientes de similaridade, medidas de dissimilaridade ou distâncias são utilizadas para descrever quantitativamente a similaridade e a dissimilaridade entre dois exemplos, dois grupos ou entre um exemplo e um grupo.

Em geral, similaridade e distância são conceitos recíprocos. A similaridade e os coeficientes de similaridade são utilizados para descrever quantitativamente o quanto dois exemplos, dois grupos ou um exemplo e um grupo são similares entre si. Ou seja, quanto maior a similaridade entre eles, maior o valor da medida de similaridade. A medida de dissimilaridade e a distância são utilizadas para medir quantitativamente o quanto dois exemplos, dois grupos ou um exemplo e um grupo não são similares. Ou seja, quanto menor a similaridade entre eles, maior o valor da medida de dissimilaridade. Para exemplificar, considere dois pontos $x_j = (x_1, x_2, \dots, x_m)$ e $y_j = (y_1, y_2, \dots, y_m)$, sendo m o número de atributos. A distância euclidiana entre os pontos x e y é calculado como segue:

$$d_{xy} = \sqrt{\sum_{j=1}^m (x_j - y_j)^2} \quad (3.1)$$

Mais detalhes sobre distâncias podem ser encontrados no trabalho de Hathaway et al [HBH00].

3.1.3 Centróide, Partição e Classe e Grupo

Os grupos são aglomerados de exemplos similares entre si. A representação desse grupo de exemplos similares é usualmente realizada por um vetor da mesma dimensão dos exemplos contidos nesse grupo, o centróide. O cálculo do centróide depende do algoritmo utilizado para o agrupamento.

Um outro fator importante a ser observado é que um ou mais grupos podem conter exemplos que representem a mesma instância de um determinado conceito (chamado de classe no caso supervisionado). A Figura 3.2(a) apresenta a formação original dos dados, onde cada classe representada pelos símbolos + e -. A Figura 3.2(b) apresenta 4 grupos distintos, sendo que cada classe é representada por 2 desses grupos. A formação de grupos é o resultado do algoritmo de agrupamento. Essa formação é chamada de partição. A partição é o conjunto de grupos resultado do algoritmo de agrupamento.

3.1.4 Agrupamento *Hard* e Agrupamento *Fuzzy*

No agrupamento clássico [DHS01] o exemplo da base de dados pertence a apenas um grupo (agrupamento *hard*). Assim, um exemplo pode ser associado à apenas um dos grupos formados pelo algoritmo de agrupamento. O número de grupos formados pelo algoritmo de agrupamento é dado pelo número C e o número de exemplos da base de dados é dado por N . No agrupamento *fuzzy* a condição de um exemplo pertencer a apenas um grupo é relaxada. Desse modo, um exemplo pode estar associado à um ou mais grupos de acordo com um grau de pertinência *fuzzy* (u_{ik}) no grupo \mathbf{v}_k . O valor do grau de pertinência está no intervalo $[0, 1]$, onde 1 representa

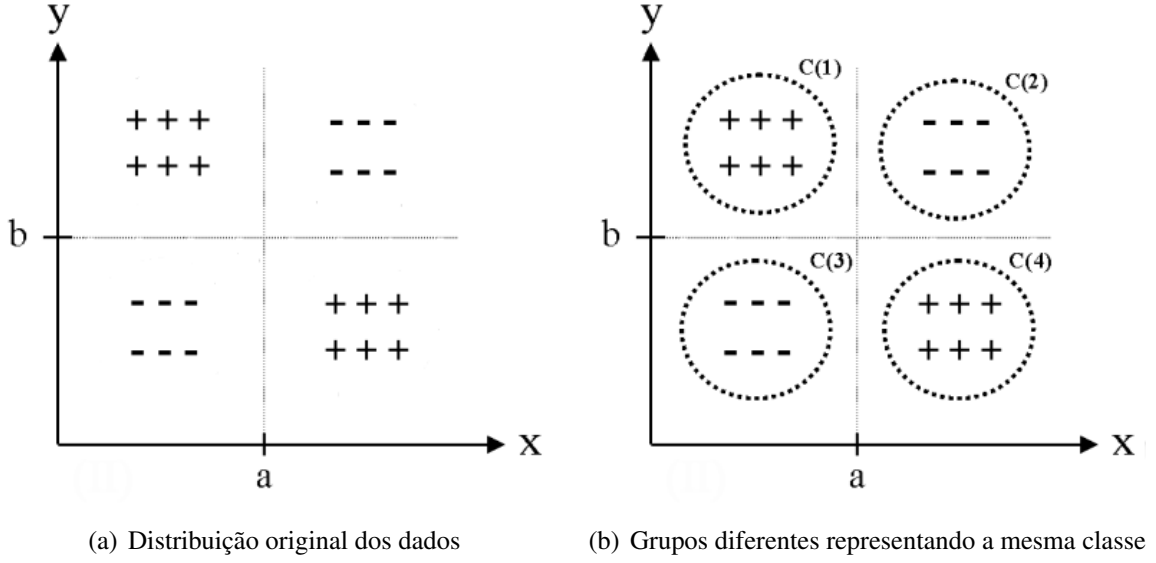


Figura 3.2 Grupos *versus* classes

pertinência total do padrão \mathbf{x}_i no grupo \mathbf{v}_k , enquanto 0 indica que este padrão não pertence ao grupo \mathbf{v}_k . Caso o valor esteja no intervalo $0 < u_{ik} < 1$ significa que o padrão \mathbf{x}_i pertence parcialmente ao grupo \mathbf{v}_k . O grau de pertinência obedece as seguintes condições:

$$0 \leq u_{ik} \leq 1 \quad (3.2)$$

$$\sum_{k=1}^C u_{ik} = 1 \quad \forall i \quad (3.3)$$

$$0 < \sum_{i=1}^N u_{ik} < N \quad \forall k \quad (3.4)$$

Uma ilustração desses dois paradigmas é apresentado na Figura 3.3. Para o conjunto *fuzzy*, o padrão x_1 está nas 3 classes ao mesmo tempo. Podemos quantificar isso à partir do grau de pertinência, que para a classe 1 é $u_{11} = 0.22$, enquanto para a classe 2 é $u_{12} = 0.63$ e para a classe 3 é $u_{13} = 0.15$, sendo a soma deles igual a 1. No conjunto *hard*, o padrão x_1 está em apenas uma classe.

3.1.5 Processo de Agrupamento

A tarefa de agrupamento não supervisionado é composta por 4 etapas [JMF99]. Porém neste trabalho a tarefa de agrupamento é semi-supervisionada. Desse modo, uma etapa foi adicionada. As etapas desse processo de agrupamento semi-supervisionado é descrito a seguir.

1. **Representação dos dados:** Nesta etapa, exemplos de objetos reais são transformados em alguma representação (ver Seção 3.1.1). Na representação, os parâmetros escolhidos

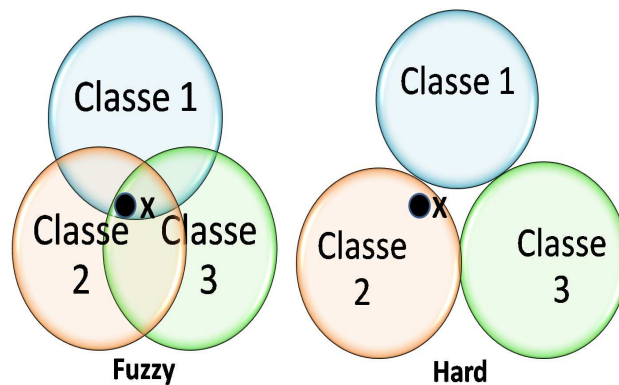


Figura 3.3 *Fuzzy Versus Hard*

nesta etapa são o número de classes, o número padrões disponíveis, bem como o número, tipo, e a escala das características disponíveis para representação da informação. A representação dos dados determina o tipo de estrutura de grupos que serão encontrados nos dados e será a entrada do algoritmo de agrupamento.

2. **Rotulação dos dados:** Para utilizarmos um algoritmo semi-supervisionado, uma porcentagem da base de dados deve estar rotulada. Caso não existam dados rotulados, o algoritmo terá o mesmo efeito de um algoritmo não supervisionado. Desse modo, um especialista é convidado a rotular uma parte da base de dados. Esses dados rotulados vão guiar o restante do processo de rotulação.
3. **Agrupamento:** Nesta etapa, executamos o algoritmo com as informações disponíveis. Esta etapa envolve o treinamento do algoritmo e a formação de grupos com base nas informações dos padrões rotulados e não rotulados. O processamento consiste de diversas iterações onde a função de otimização converge até que um valor ótimo. A função objetivo pode mudar de acordo com a constituição de cada algoritmo, mas o processo de agrupamento é semelhante para todos eles. O resultado do processo de agrupamento pode ser um agrupamento *hard*, exemplos contidos em apenas um dos grupos formados, ou *fuzzy*, onde um exemplo pertence a cada um dos grupos formados baseado num grau de pertinência.
4. **Abstração dos Dados:** Quando o algoritmo termina sua execução, os grupos estão formados. Assim, são utilizadas métricas para associar um determinado grupo a uma classe pré-definida. Desse modo, os padrões são rotulados de acordo com o grupo que ele estiver associado no final do processo.

Para ilustrar o processo de rotulação, considere a figura Figura 3.4. Nessa figura os padrões não rotulados são representados pelos triângulos, e os padrões rotulados são representados pela estrela vermelha e o sinal positivo verde que representam duas classes respectivamente. Na primeira figura, os conjuntos originais de dados com seus respectivos padrões rotulados e não

rotulados são apresentados. Na segunda figura, os padrões rotulados guiam o processo de agrupamento. Na terceira figura, os padrões rotulados afetam os padrões não rotulados na formação dos grupos. E assim, na última figura, os padrões não rotulados são atribuídos a grupos cujo a similaridade seja maior de acordo com alguma medida (esses padrões são representados por triângulos azul claro). Os padrões rotulados são destacados pelas setas. Os algoritmos que procedem dessa forma são chamados de algoritmos de agrupamento. Há diversos tipos de algoritmos desse tipo [DHS01][Mit97]. Neste trabalho vamos explorar algoritmos de agrupamento que utilizam a abordagem semi-supervisionada e são algoritmos *fuzzy*.

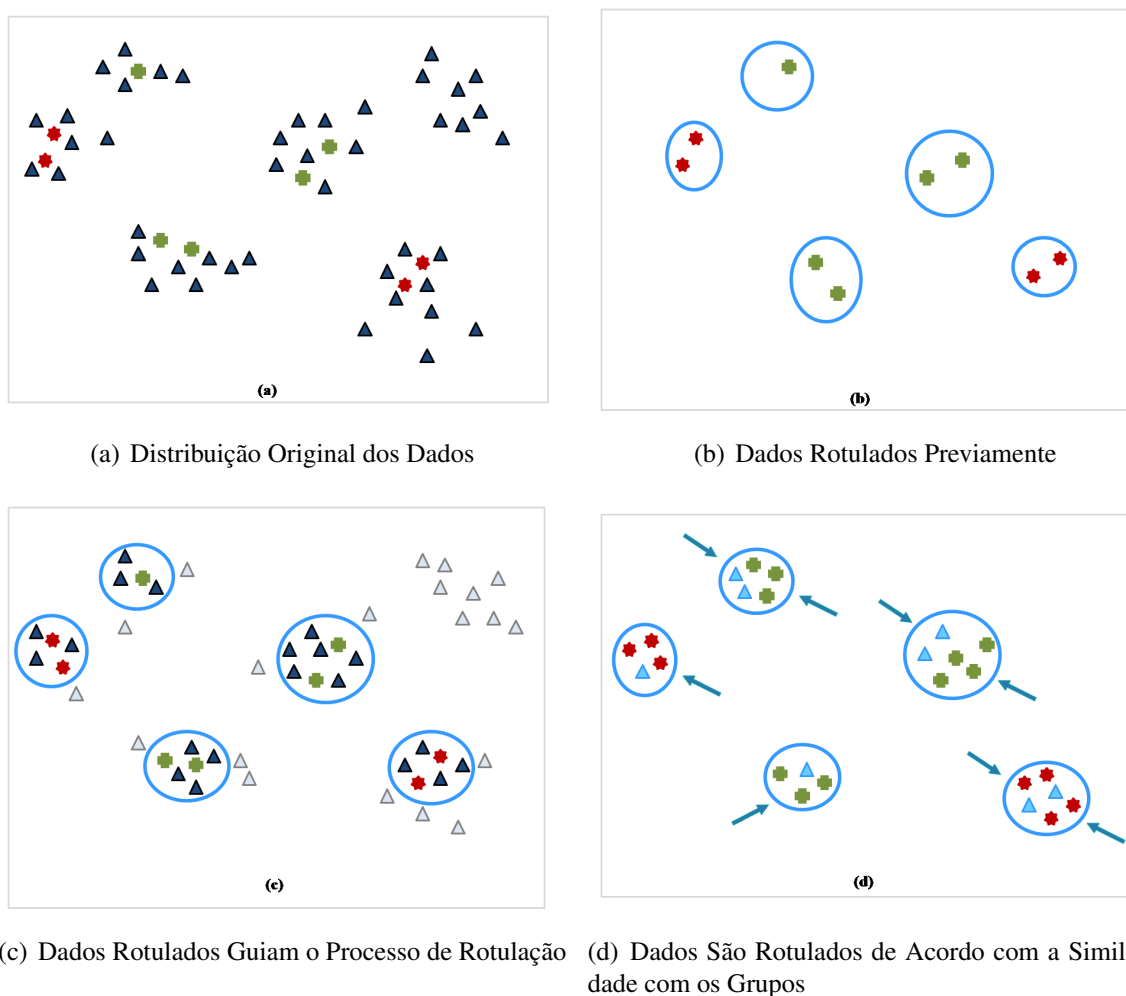


Figura 3.4 Processo Geral de Rotulação por Algoritmos de Agrupamento

Após explicar características do processo de rotulação e de explicar características dos algoritmos de agrupamento, podemos começar a descrever detalhadamente algoritmos desse tipo. O primeiro algoritmo desse capítulo é um algoritmo clássico, o *Fuzzy C-Means* (FCM), cujo todos os outros algoritmos se baseiam. Esse algoritmo é não supervisionado. Através das mudanças na função objetivo desse algoritmo, todos os outros algoritmos se tornam semi-supervisionados. As mudanças e detalhes desses algoritmos são apresentadas nas seções posteriores.

3.2 Algoritmo de Agrupamento *Fuzzy C-Means*

Antes de começar a descrever os algoritmos de agrupamento semi-supervisionados, é preciso falar do algoritmo que influenciou todos os algoritmo deste capítulo. O algoritmo *Fuzzy C-Means* (FCM)[Bez81] é o ponto de partida para os algoritmos de agrupamento semi-supervisionados descritos.

O algoritmo FCM otimiza uma função, chamada de função objetivo, através das iterações para calcular seus parâmetros desconhecidos. Tornando valor dessa função menor a cada iteração. As equações do algoritmo FCM são explicadas logo abaixo.

Com respeito a notação que será utilizada neste capítulo, temos que V é um conjunto de vetores que representam os grupos do algoritmo (protótipo), $V = [V_1^T, \dots, V_k^T]^T$. U é uma matriz $N \times C$ cujos elementos são os graus de pertinência u_{ik} , N é o número de exemplos (padrões) presentes na base de dados e C o número de grupos formados pelo algoritmo. A dissimilaridade d_{ik} entre os vetores do padrão \mathbf{x}_i e do protótipo \mathbf{v}_k é representada pela distância entre eles. $m(> 1)$ é chamado de fuzificador, ele ajusta o *blending* de grupos diferentes. A função objetivo do algoritmo FCM é:

$$J(U, V) = \sum_{i=1}^C \sum_{k=1}^N u_{ik}^m d_{ik}^2 \quad (3.5)$$

com respeito a U , respeita as condições das equações (3.2), (3.3) e (3.4).

A distância é calculada pela equação (3.6):

$$d_{ik}^2 = \|\mathbf{x}_i - \mathbf{v}_k\|_2^2 = (\mathbf{x}_i - \mathbf{v}_k)^T (\mathbf{x}_i - \mathbf{v}_k) \quad (3.6)$$

Diferentes valores de m na equação (3.5) leva $J_m(U, V)$ a ter agrupamentos *hard* ou *fuzzy*. Para um m fixo, $m = 1$, nenhum agrupamento *fuzzy* é melhor que o agrupamento *hard*. Entretanto, se $m > 1$, há casos onde os agrupamentos *fuzzy* lidam com valores baixos de $J_m(U, V)$ que são melhores que os agrupamentos *hard* [TK06].

Para encontrar as equações dos parâmetros desconhecidos, primeiro, é assumido que nenhum padrão coincide com um protótipo. Assim, nenhuma distância de uma padrão ao protótipo é zero. A minimização de (3.5) utilizando o multiplicador de lagrange com respeito a U leva a seguinte função:

$$J(U, V, \lambda) = \sum_{i=1}^N \sum_{k=1}^C u_{ik}^m d_{ik}^2 - \sum_{i=1}^N \lambda_i \left(\sum_{k=1}^C u_{ik} - 1 \right) \quad (3.7)$$

O resultado da derivada parcial da equação (3.7) em relação a U igualado a zero leva a equação do cálculo do grau de pertinência:

$$u_{ik} = \frac{1}{\sum_{j=1}^C \left(\frac{d_{ik}}{d_{ij}} \right)^{\frac{1}{m-1}}} \quad (3.8)$$

A dissimilaridade é uma distância entre dois pontos. No caso do algoritmo FCM, essa distância é dada pela equação (3.6). Minimizando a equação objetivo em relação a V , temos a fórmula que calcula o protótipo:

$$\mathbf{v}_k = \frac{\sum_{i=1}^N (u_{ik})^2 \mathbf{x}_i}{\sum_{i=1}^N (u_{ik})^2} \quad (3.9)$$

Em geral, o critério da função objetivo é otimizado quando o protótipo dos grupos estão próximos daquele ponto que possui a maior probabilidade de ser o representante do grupo \mathbf{v}_k .

Agora, com todas as equações definidas, as etapas do algoritmo FCM podem ser descritas. As etapas para esse algoritmo são as mesmas para todos os algoritmos de agrupamento semi-supervisionado descritos abaixo. A características desses algoritmos é que após o passo de inicialização do algoritmo, o cálculo da matriz de graus de pertinência U e da matriz de protótipos se alternam até que seja alcançado algum critério de parada. Esses passos são descritos logo abaixo e também servem para os algoritmos que vão ser descritos logo abaixo:

- 1 *Iniciar os parâmetros do algoritmo: Fixar k , tal que $1 < k \leq n$; Fixar $MaxIter$ (número de iterações); Atribuir um valor para ε , tal que $0 < \varepsilon < 1$; Iniciar Matriz de Protótipos; Iniciar Matriz de Graus de Pertinência;*
- 2 *Atualizar o valor da matriz de graus de pertinência U utilizando (3.8);*
- 3 *Obter o valor da matriz de protótipos V utilizando (3.9);*
- 4 *Repetir os passos 2 e 3 até que o algoritmo obedeça algum critério de parada;*

Algoritmo 1: Algoritmo de Agrupamento Fuzzy C-Means

3.3 Algoritmos de Agrupamento Semi-Supervisionado

Na classificação de padrões, as duas abordagens clássicas mais utilizadas são a aprendizagem não supervisionada e a supervisionada. Porém, existem algumas limitações inerentes a cada uma dessas abordagens. A classificação não supervisionada não possui informações descritivas acerca dos padrões, desse modo, muitas vezes a tarefa de associar clusters a uma categoria pré-definida é uma tarefa complexa. Quando se possui informações sobre os padrões, a abordagem supervisionada é geralmente utilizada. Mas existem bases de dados onde essas informações são custosas e às vezes até impossíveis de se obter. Por exemplo, imagine rotular uma base contendo 10000 textos ou imagens, ou ainda, rotular cadeias genéticas contendo milhares de genes. A classificação semi-supervisionada surgiu como uma opção para atenuar algumas das limitações das duas abordagens acima.

O algoritmo semi-supervisionado é importante justamente por precisar de pouca informação sobre a base de dados que vai ser classificada. A abordagem semi-supervisionada, geralmente, obtém um resultado melhor do que se utilizassem apenas padrões não rotulados numa abordagem não supervisionada ou se utilizasse os poucos padrões rotulados em uma abordagem supervisionada.

Nesta seção são descritos os algoritmos de agrupamento semi supervisionados explorados nesta dissertação. A diferença dos diferentes algoritmos estão nas equações e algumas características inerentes a função objetivo de cada um dos algoritmos. Os algoritmos são:

1. Algoritmo de Pedrycz
2. Algoritmo de Bouchachia
3. Algoritmo Baseado em Sementes
4. Algoritmo de Agrupamento Semi-Supervisionado Proposto

3.3.1 Algoritmo de Pedrycz

O método proposto por Pedrycz [Ped85][PW97] é um dos primeiros algoritmos à utilizar a abordagem de agrupamento para algoritmos semi-supervisionados. Este algoritmo, assim como outros desse paradigma, foi concebido para atenuar os defeitos dos paradigmas mais utilizados, supervisionado e não supervisionado. De posse de bases de sinais biológicos (eletrocardiogramas) onde havia poucos elementos rotulados, Pedrycz formulou um novo algoritmo modificando a função objetivo (3.5) do algoritmo FCM. O termo adicionado a função objetivo do algoritmo FCM utiliza os rótulos disponíveis dos exemplos da base de dados para aumentar os graus de pertinência atribuídos pelo algoritmo para os grupos que representam a classe que o exemplo pertence. A função objetivo desse algoritmo depois de modificada do original *Fuzzy C-Means* (FCM) [Bez81] se torna:

$$J(U, V) = \sum_{i=1}^C \sum_{k=1}^N u_{ik}^2 d_{ik}^2 + \alpha \sum_{i=1}^c \sum_{k=1}^N (u_{ik} - f_{ik} b_i)^2 d_{ik}^2 \quad (3.10)$$

tal que U respeita as condições das equações (3.2), (3.3) e (3.4).

Aqui, α ($\alpha \geq 0$) é um fator que mantém o balanceamento entre os componentes supervisionados e não supervisionados. Neste trabalho, $\alpha = 1$ para todos os experimentos. Os parâmetros que conseguem tirar vantagem da informação dos rótulos são o termo binário b_i e a matriz f_{ik} . O termo b_i , $i = 1, 2, \dots, N$, onde N é o número de padrões da base de dados, distingue padrões rotulados e não rotulados:

$$b_i = \begin{cases} 1 & \text{se padrão } \mathbf{x}_i \text{ é rotulado} \\ 0 & \text{senão} \end{cases}$$

A matriz $F = [f_{ik}]$ sendo $i = 1, 2, \dots, N$ e $k = 1, 2, \dots, C$, onde C é o número de grupos formados pelo algoritmo, contém os valores dos graus de pertinência dos padrões rotulados, sendo 1 o valor do grau de pertinência para o grupo que representa a classe desse padrão e 0 para os outros grupos. U representa a matriz que contém os graus de pertinência. Cada u_{ik} representa o grau de pertinência do padrão \mathbf{x}_i no grupo \mathbf{v}_k . V representa o conjunto de protótipos \mathbf{v}_k associado a cada grupo. O sobrescrito m é o grau de fuzificação e $m = 2$. Por último temos a distância euclidiana entre o padrão \mathbf{x}_i e o grupo \mathbf{v}_k representado por d_{ik} . Neste algoritmo, o número de grupos formados pelo algoritmo é igual ao número de classes conhecidas previamente.

Após a explicação da função objetivo, agora é descrito como calcular as matrizes de pertinência e de protótipos que são os parâmetros calculados iterativamente nos algoritmos de agrupamento. Utilizando a técnica padrão do multiplicador de lagrange para minimizar a função (3.10) em relação a U , a equação do grau de pertinência é:

$$u_{ik} = \frac{1}{1 + \alpha} \left\{ \frac{1 + \alpha(1 - b_i \sum_{l=1}^C f_{il})}{\sum_{l=1}^C (\frac{d_{ik}^2}{d_{il}^2})} + \alpha f_{ik} b_i \right\} \quad (3.11)$$

A otimização de (3.10) em relação a V obtém a equação dos protótipos igual ao algoritmo FCM:

$$\mathbf{v}_k = \frac{\sum_{i=1}^N u_{ik}^2 \mathbf{x}_i}{\sum_{i=1}^N u_{ik}^2} \quad (3.12)$$

O algoritmo completo é resumido em etapas e descrito logo abaixo:

- 1 *Iniciar os parâmetros do algoritmo: Fixar C , tal que $1 < C \leq N$; Fixar $MaxIter$ (número de iterações); Iniciar contador de iterações $s = 0$; Atribuir um valor para ε , tal que $0 < \varepsilon \ll 1$; Iniciar Matriz de Protótipos; Iniciar Matriz de Graus de Pertinência, incluindo todos os valores de pertinência conhecidos;*
- 2 *Obter o valor da matriz de protótipos V dos grupos utilizando (3.12);*
- 3 *Atualizar o valor da matriz de partição U utilizando (3.11);*
- 4 *Repetir os passos 2 e 3 até que o algoritmo obedeça algum critério de parada;*

Algoritmo 2: Algoritmo de Supervisão Parcial de Pedrycz

3.3.2 Algoritmo de Bouchachia

O algoritmo proposto por Bouchachia e Pedrycz [BP06] basicamente estende a função objetivo (3.5) do algoritmo *Fuzzy C-Means*. O objetivo dessa mudança é capturar estruturas dos dados escondidas e visíveis através de dois termos da função objetivo. As estruturas escondidas são adquiridas pelo primeiro termo da função objetivo, que é igual a função objetivo do algoritmo FCM. O segundo termo leva em conta as estruturas refletidas pela avaliação dos rótulos disponíveis. Desse modo, a função objetivo se torna:

$$J(U, V) = \sum_{i=1}^C \sum_{k=1}^N u_{ik}^2 d_{ik}^2 + \alpha \sum_{i=1}^C \sum_{k=1}^N (u_{ik} - \tilde{u}_{ik})^2 d_{ik}^2 \quad (3.13)$$

Assim, U representa a matriz de graus de pertinência u_{ik} do padrão \mathbf{x}_i no grupo \mathbf{v}_k . V representa o conjunto de protótipos \mathbf{v}_k associados aos grupo. N é o tamanho da base de dados e C o número de grupos. O sobrescrito m é o grau de fuzzificação e $m = 2$. Por último temos a distância euclidiana entre o padrão \mathbf{x}_i e o grupo \mathbf{v}_k representado por d_{ik} . O α é um parâmetro maior que 0 que realiza o balanceamento entre os termos supervisionado e não supervisionado. Neste trabalho, o α é igual a 1. O parâmetro \tilde{u} será explicado logo abaixo. Os graus de pertinências u_{ik} obedecem as restrições (3.2), (3.3) e (3.4).

Uma das forças desse algoritmo é o estudo da possibilidade de mais de um grupo representar uma determinada classe. Assim, a próxima equação demonstra essa característica. H designa

o número de classes (rótulos), então $C \geq H$. Cada classe h contém o número de grupos C_h , assim:

$$\sum_{h=1}^H C_h = C \quad (3.14)$$

O termo \tilde{u}_{ik} da matriz \tilde{U} é iterativamente calculado como segue:

$$\tilde{u}_{ik}^{(s)} = \tilde{u}_{ik}^{(s-1)} - \beta \frac{\delta Q(F, \tilde{U})}{\delta \tilde{u}_{ik}} \quad (3.15)$$

onde s é contador da iteração e

$$Q(F, \tilde{U}) = \sum_{h=1}^H \sum_{i=1}^N \delta_i (f_{ih} - \sum_{k \in \pi_h} \tilde{u}_{ik}) \quad (3.16)$$

tal que $\tilde{u}_{ik} \in [0, 1]$

Também, em (3.16), $F = [f_{ih}]$ é uma matriz binária $H \times N$ tal que $f_{ih} = 1$ se o padrão x_i pertence à classe h , senão $f_{ih} = 0$. Esta matriz serve para representar a informação dos rótulos disponíveis. π_h é o conjunto de grupos que pertencem a classe h . δ_i é um valor binário que indica se o padrão é rotulado ou não:

$$\delta_i = \begin{cases} 1 & \text{se o padrão } x_i \text{ é rotulado} \\ 0 & \text{senão} \end{cases}$$

Aplicando o multiplicador de lagrange para minimizar a equação objetivo (3.13) para cada $i = 1, 2, \dots, N$ e derivando em relação a matriz U , a equação do grau de pertinência se torna:

$$u_{ik} = \frac{\alpha \tilde{u}_{ik}}{(1 + \alpha)} + \frac{1 - \frac{\alpha}{(1 + \alpha)} \sum_{l=1}^C \tilde{u}_{il}}{\sum_{l=1}^C \frac{d_{ik}^2}{d_{il}^2}} \quad (3.17)$$

Otimizando (3.13) em relação a V e levando em conta a equação de d_{ik}^2 em (3.1), tem-se:

$$\mathbf{v}_k = \frac{\sum_{i=1}^N (u_{ik}^2 + \alpha(u_{ik} - \tilde{u}_{ik})^2) \mathbf{x}_i}{\sum_{i=1}^N (u_{ik}^2 + \alpha(u_{ik} - \tilde{u}_{ik})^2)} \quad (3.18)$$

O termo \tilde{u}_{ik} da matriz \tilde{U} é conseguido através da segunda otimização do algoritmo. A função (3.16) é otimizada em relação a \tilde{U} . Essa otimização explora o objetivo geral de reduzir a diferença entre o grau de pertinência atual u_{ik} de um padrão rotulado \mathbf{x}_i a um grupo \mathbf{v}_k . Desse modo, obtém a expressão que calcula o termo \tilde{u}_{ik} :

$$\tilde{u}_{ik}^{(s)} = \tilde{u}_{ik}^{(s-1)} - 2\beta \delta_i \sum_{h=1}^H (f_{hi} - \sum_{k \in \pi_h} \tilde{u}_{ik}^{(s-1)}) * \begin{cases} 1 & \text{se } k \in \pi_h \\ 0 & \text{senão} \end{cases} \quad (3.19)$$

O termo π_h é importante na equação (3.19), pois contém o conjunto de grupos que representam a classe h . O processo para se calcular a soma ψ_h é descrito logo abaixo:

$$\psi_h = \sum_{k \in \pi_h} \tilde{u}_{ik} \quad (3.20)$$

Duas matrizes tem que ser calculadas para se computar ψ , uma matriz de mapeamento $M_{H \times C}$ e uma matriz correspondente $P_{H \times C}$. A primeira especifica o relacionamento entre as classes e os grupos enquanto a última especifica o número de padrões de cada classe em cada grupo. A célula $P(h, k)$ representa o número de padrões de uma classe h no grupo \mathbf{v}_k . Um padrão \mathbf{x}_i pertence ao grupo \mathbf{v}_k se o grau de pertinência do padrão \mathbf{x}_i a \mathbf{v}_k é o maior. Assim, para cada classe, será construída uma lista de grupos e o número de padrões associados a essa classe. Ordenando a matriz P na ordem ascendente, cada linha de M contém a lista de grupos ordenados por dominância. O número de classes por grupo é especificado no começo do algoritmo e levado em conta. Assim, o conjunto π_h de ψ_h contém os grupos dominantes na classe h e assim, pode-se realizar a soma.

Depois de formular todas as expressões, o processo de agrupamento pode ser descrito. Os passos do algoritmo semi-supervisionado é mostrado logo abaixo. Note que este algoritmo possui um passo iterativo a mais em relação aos outros algoritmos de agrupamento deste capítulo. É o cálculo da matriz \tilde{U} que é realizado antes dos cálculos da matriz de protótipos e da matriz de graus de pertinência.

- 1 *Primeiro iniciar os parâmetros do algoritmo: Fixar C , tal que $1 < C \leq N$; Fixar $MaxIter$ (número de iterações); Iniciar contador de iterações $s = 0$; Atribuir um valor para ϵ , tal que $0 < \epsilon \ll 1$; Iniciar Matriz de Protótipos; Iniciar Matriz de Graus de Pertinência, incluindo todos os valores de pertinência conhecidos;*
- 2 *Determinar o conjunto π_h de cada classe de acordo com a noção de dominância explicada acima;*
- 3 *Computar a matriz de mapeamento $M_{H \times C}$ que relaciona classes a grupos:
 $M(h, k) = 1$ se grupo \mathbf{v}_k pertence a classe H , senão 0;*
- 4 *Inicializar $\tilde{U}^{(0)}$ com $U^{(0)}$ e contador de iterações $s = 1$;*
- 5 **repita**
- 6 **repita**
- 7 *Computar $\tilde{U}^{(s)}$ utilizando (3.19);*
- 8 **até** $\|\tilde{U}^{(s)} - \tilde{U}^{(s-1)}\| < \tau$ onde τ é um valor pequeno ;
- 9 **repita**
- 10 *Obter o valor da matriz de protótipos $V^{(s)}$ dos grupos utilizando (3.18);*
- 11 *Atualizar o valor da matriz de partição $U^{(s)}$ utilizando (3.17);*
- 12 **até** $\|U^{(s)} - U^{(s-1)}\| < \epsilon$;
- 13 *Computar a matriz $M^{(s)}$;*
- 14 **até** $M^{(s)} = M^{(s-1)}$ ou $s = MaxIter$;

Algoritmo 3: Algoritmo de Bouchachia

3.3.3 Algoritmo Baseado em Sementes

O algoritmo baseado em sementes [AYM⁺02] foi idealizado para resolver um problema de segmentação de imagens com problemas de ruído. O algoritmo baseado em sementes é formulado pela modificação da função objetivo da norma do algoritmo FCM. Essa nova formulação permite que a rotulação de um píxel (voxel) seja realizada sob o efeito dos rótulos na sua vizinhança imediata. O efeito da vizinhança atua como um regularizador e influencia a solução em direção a uma rotulação homogênea. Tal regularização é útil na segmentação de imagens corrompidas por ruído de sal e pimenta. Esse algoritmo foi testado em outras aplicações de agrupamento semi-supervisionado em [Bou07], o que motivou a utilização desse algoritmo neste trabalho.

Com a alteração realizada na equação (3.5) do algoritmo FCM a fórmula do algoritmo semi-supervisionado baseado em sementes se tornou:

$$J(U, V) = \sum_{i=1}^C \sum_{k=1}^N u_{ik}^2 d_{ik}^2 + \frac{\alpha}{N_R} \sum_{i=1}^C \sum_{k=1}^N u_{ik}^2 \left(\sum_{x_r \in N_k} d^2 \right) \quad (3.21)$$

tal que os graus de pertinências u_{ik} obedecem as restrições (3.2), (3.3) e (3.4)

onde N é o tamanho da base de dados e C é o número de grupos. N_k representa o conjunto de vizinhos que existem na janela em torno de x_r e N_R é a cardinalidade de N_k . O conjunto de vizinhos são os exemplos que estão próximos entre si de acordo com o critério de similaridade. O efeito do termo que leva em conta a influência de vizinhos é controlado pelo parâmetro α , nos experimentos este α possui valor 1. Continuando, U representa a matriz de graus de pertinência u_{ik} do padrão \mathbf{x}_i no grupo \mathbf{v}_k . V representa o conjunto de protótipos \mathbf{v}_k associado a cada grupo. O sobrescrito m é o grau de fuzificação e $m = 2$. Por último temos a distância euclidiana entre o padrão \mathbf{x}_i e o grupo \mathbf{v}_k representado por d_{ik} .

Derivando a função objetivo (3.21) em relação a u temos a equação do grau de pertinência u_{ik} deste algoritmo. Esta equação é:

$$u_{ik} = \frac{1}{\sum_{l=1}^C \left(\frac{d_{ik} + \frac{\alpha}{N_R} \gamma_k}{d_{il} + \frac{\alpha}{N_R} \gamma_l} \right)^2} \quad (3.22)$$

onde $\gamma_k = \sum_{x_k \in N_k} d_{ik}^2$.

O protótipo é dado pela derivação de (3.21) em relação a \mathbf{v}_k . A equação do protótipo é:

$$\mathbf{v}_k = \frac{\sum_{i=1}^N u_{ik}^2 (\mathbf{x}_i + \frac{\alpha}{N_R} \sum_{\mathbf{x}_r \in N_k} \mathbf{x}_r)}{(1 + \alpha) \sum_{i=1}^N u_{ik}^2} \quad (3.23)$$

Este algoritmo se chama baseado em sementes por causa de sua inicialização. Para os algoritmos de agrupamento não supervisionados, os exemplos utilizados para iniciar a matriz de protótipos são todos não rotulados, porém neste algoritmo são utilizados apenas os exemplos rotulados na inicialização da matriz de protótipos. Neste algoritmo, os grupos formados são iguais ao número de classes conhecidas previamente. Os passos do algoritmo baseado em sementes são:

- 1 *Primeiro iniciar os parâmetros do algoritmo: Fixar C , tal que $1 < C \leq N$; Fixar $MaxIter$ (número de iterações); Iniciar contador de iterações $s = 0$; Atribuir um valor para ϵ , tal que $0 < \epsilon \ll 1$;*
- 2 *Iniciar Matriz de Protótipos utilizando apenas os exemplos rotulados; Iniciar Matriz de Graus de Pertinência, incluindo todos os valores de pertinência conhecidos;*
- 3 *Atualizar o valor da matriz de partição U utilizando (3.22);*
- 4 *Obter o valor da matriz de protótipos V utilizando (3.23);*
- 5 *Repetir os passos 3 e 4 até que o algoritmo obedeça algum critério de parada;*

Algoritmo 4: Algoritmo Baseado em Sementes

3.3.4 Algoritmo de Agrupamento Semi-Supervisionado Proposto

A idéia principal dos algoritmos semi-supervisionados é aproveitar alguma informação descritiva disponível em alguns dados. Geralmente essa informação é o rótulo dos padrões. Aproveitando essa informação, o algoritmo otimiza sua função objetivo para marcar exemplos de forma precisa. O algoritmo proposto acrescenta um segundo termo supervisionado à função original de Bezdek [Bez81]. No segundo termo supervisionado, o novo algoritmo parte do princípio que a informação dos padrões vizinhos rotulados são importantes pra descobrir a estrutura dos dados. Assim, na função de otimização, os vizinhos são comparados entre si, e caso pertençam a uma mesma classe, a função é otimizada nessa direção. A otimização é realizada pela diferença entre os graus de pertinência de padrões que pertençam à mesma classe. Caso, os padrões pertençam à mesma classe é suposto que a distância entre eles seja pequena. Desse modo, a diferença entre os graus de pertinência para a classe que os dois pertencem é penalizada caso essa diferença seja grande. A função objetivo do novo algoritmo de agrupamento semi-supervisionado, dado que N padrões são agrupados em C grupos é:

$$J(U, V) = \sum_{k=1}^C \sum_{i=1}^N u_{ik}^2 d_{ik}^2 + \sum_{k=1}^C \sum_{l=1}^P \sum_{i=1}^N \sum_{j=1}^N t_{il} t_{jl} (u_{ik} - u_{jk})^2 d_{ij}^2 \quad (3.24)$$

tal que U respeita as condições das equações (3.2), (3.3) e (3.4).

N é o tamanho da base de dados, C o número de grupos formados pelo algoritmo e P o número de classes previamente conhecidas. Note que no algoritmo proposto, assim como o algoritmo de Bouchachia, uma classe pode ser representada por um ou mais grupos. A matriz U possui como elemento u_{ik} que representa o grau de pertinência do padrão \mathbf{x}_i no grupo \mathbf{v}_k , sendo $i = (1, 2, \dots, N)$ e $k = (1, 2, \dots, C)$. V representa o conjunto de protótipos \mathbf{v}_k associado a cada grupo ou partição. O sobrescrito m é o grau de fuzificação e $m = 2$. Note que este algoritmo não possui o fator de balanceamento α presente nos outros algoritmos. Por último temos a distância entre o padrão \mathbf{x}_i e outro padrão \mathbf{x}_j representada por d_{ij} . A informação supervisionada é utilizada pelo termo bi valorado t_{ik} , $i = 1, 2, 3, \dots, N$. Esse valor é 1 caso o padrão \mathbf{x}_i pertença ao grupo \mathbf{v}_k e 0 senão:

$$t_{ik} = \begin{cases} 1 & \text{se padrão } \mathbf{x}_i \text{ pertence ao grupo } \mathbf{v}_k \\ 0 & \text{senão} \end{cases}$$

Para otimizar a equação (3.24) é utilizado o multiplicador de lagrange com respeito a matriz U e a matriz de protótipos V , obtendo as equações para o cálculo iterativo dos graus de pertinência u_{ik} e dos protótipos \mathbf{v}_k . Otimizando em relação a matriz U temos a equação:

$$u_{ik}^{(s)} = \frac{1 + \sum_{h=1}^C \frac{\sum_{l=1}^P \sum_{j=1}^N t_{il} t_{jl} [|u_{jk}^{(s-1)} - u_{jh}^{(s-1)}|] d_{ij}^2}{d_{ih}^2 + \sum_{l=1}^P \sum_{j=1}^N t_{il} t_{jl} d_{ij}^2}}{\sum_{h=1}^C \frac{d_{ik}^2 + \sum_{l=1}^P \sum_{j=1}^N t_{il} t_{jl} d_{ij}^2}{d_{ih}^2 + \sum_{l=1}^P \sum_{j=1}^N t_{il} t_{jl} d_{ij}^2}} \quad (3.25)$$

onde s é o contador de iterações. Uma das características desse cálculo é a necessidade do grau de pertinência de uma iteração anterior. Portanto, há necessidade de popular a matriz U antes de utilizar esta formula pela primeira vez. Algumas abordagens são explicadas na Seção 3.3.5.3.

Agora, otimizando em relação à matriz V temos a equação para o cálculo do protótipo. Note que esta equação para o cálculo do protótipo é igual a equação utilizada no algoritmo base *Fuzzy C-Means*:

$$\mathbf{v}_k = \frac{\sum_{i=1}^N u_{ik}^2 \mathbf{x}_i}{\sum_{i=1}^N u_{ik}^2} \quad (3.26)$$

Após formular todas as expressões, o processo de agrupamento semi-supervisionado pode ser descrito. Os passos do algoritmo semi- supervisionado proposto é apresentado no pseudo-código logo abaixo.

```

1  Iniciar os parâmetros do algoritmo: Fixar  $C$ , tal que  $1 < C \leq N$ ; Fixar  $MaxIter$ 
   (número de iterações); Iniciar contador de iterações  $s = 0$ ; Atribuir um valor para  $\epsilon$ ,
   tal que  $0 < \epsilon \ll 1$ ; Iniciar Matriz de Protótipos; Iniciar Matriz de Graus de
   Pertinência, incluindo todos os valores de pertinência conhecidos;
2  Obter o valor da matriz  $V$  de protótipos dos grupos utilizando (3.26);
3  Calcular o valor da função objetivo  $J$  utilizando (3.24);
4  se  $(|J^{(s)} - J^{(s-1)}| \leq \epsilon$  ou  $s \geq MaxIter)$  e  $(s > 1)$  então
5     Pare o algoritmo
6  fim
7  senão
8     Atualizar o valor da matriz de partição  $U^{(s)}$  utilizando (3.25);
9     Vá para o passo 2;
10 fim

```

Algoritmo 5: Algoritmo Semi Supervisionado Proposto

3.3.5 Parâmetros Importantes nos Algoritmos de Agrupamento

Nesta seção serão apresentados dois parâmetros que são importantes para obter um bom resultado em algoritmos de agrupamento. Os algoritmos de agrupamento são sensíveis à normalização da base de dados, às inicializações das matrizes de protótipo, de grau de pertinência e

ao momento de parada do algoritmo. Esses parâmetros alteram significativamente o resultado final desses algoritmos. Por isso, nesta seção abordamos algumas abordagens importantes para esses parâmetros.

3.3.5.1 Normalização da Base de Dados

É indicado que a normalização dos dados seja utilizada. A normalização possui como objetivo evitar discrepância entre a relevância dos atributos de uma base de dados. Por exemplo, no mundo real, pode-se facilmente encontrar atributos como idade e peso de uma pessoa. Diferenças de escala entre esses atributos podem levar a uma distorção no cálculo da distância euclidiana. Existem algumas opções para se fazer a normalização. a primeira pode transformar os valores em escala decimal, dividindo os valores dos atributos pela mesma potência de 10. Também existe a normalização Min-Max, onde os valores são substituídos por valores entre $[Novo_Min, Novo_Max]$. Esses valores são encontrados a partir do cálculo de onde se utiliza o valor mínimo e o valor máximo da base de dados. E ainda há o *Z-score* onde os valores são padronizados através da diferença da média do atributo. E para finalizar, a normalização logarítmica, que substitui o valor pelo logaritmo de uma base. O objetivo dessa normalização é transformar os valores de base de dados para uma escala proporcional.

3.3.5.2 Inicialização da Matriz de Protótipos

A matriz de protótipos é inicializada geralmente escolhendo aleatoriamente um exemplo da base de dados. Para que essa abordagem consiga algum efeito, é importante repetir a execução do algoritmo diversas vezes com diferentes exemplos escolhidos aleatoriamente da base de dados para compor a matriz inicial de protótipos. Algumas outras abordagens também podem ser realizadas. Em Bouchachia e Pedrycz [BP06], por exemplo, a matriz de protótipos é inicializada a partir dos resultados do algoritmo *Fuzzy C-Means* original.

3.3.5.3 Inicialização da Matriz de Grau de Pertinência

A inicialização da matriz de grau de pertinência pode ser inicializada em diversas maneiras. Essa matriz é característica de algoritmos que obtêm partição *fuzzy* como resultado. Nesta seção abordamos 3 abordagens para iniciar essa matriz.

A primeira, é chamada aqui de inicialização *fuzzy*. Nessa abordagem, o grau de pertinência u_{ik} de um padrão \mathbf{x}_i a cada um dos grupos \mathbf{v}_k é atribuído aleatoriamente e possui valor no intervalo $[0, 1]$. A soma dos graus de pertinência de um padrão aos grupos deve ser 1.

A segunda abordagem, é chamada aqui de inicialização *hard*. O grau de pertinência u_{ik} de um padrão \mathbf{x}_i a um grupo \mathbf{v}_k é atribuído aleatoriamente à apenas um dos grupos. O valor desse grau de pertinência possui o valor 1. Para manter a condição da soma dos graus de pertinência de um padrão igual a 1, o os graus de pertinência desse padrão \mathbf{x}_i aos outros grupos será 0.

A terceira abordagem é um pouco mais elaborada. Nessa abordagem, chamada aqui de inicialização por agrupamento, é calculada a similaridade do padrão \mathbf{x}_i ao grupo \mathbf{v}_k . O grau de pertinência do padrão \mathbf{x}_i ao grupo \mathbf{v}_k é calculado pela similaridade do padrão \mathbf{x}_i ao grupo \mathbf{v}_k dividido pela soma das similaridades desse padrão a todos os grupos. Há uma excessão para o caso de o padrão ser igual a um grupo ou mais. Nesse caso, a similaridade é igual a 1

dividido pelo número de grupos iguais, para esse(s) grupo(s) e 0 nos demais. As equações dessa abordagem são mostradas abaixo com d_{ik} representando o valor da similaridade do padrão \mathbf{x}_i ao grupo \mathbf{v}_k e N representando o número de padrões.

Para o caso onde o padrão não é igual ao protótipo do grupo:

$$u_{ik} = \left[\frac{d_{ik}}{\sum_{j=1}^K d_{ik}} \right]^{-1} \quad (3.27)$$

Para o caso do padrão ser igual a um ou mais protótipos dos grupos:

$$u_{ik} = \begin{cases} \frac{1}{\text{número de protótipos iguais ao padrão}} & \text{se padrão } \mathbf{x}_i \text{ igual ao protótipo } \mathbf{v}_k \\ 0 & \text{senão} \end{cases}$$

ambos os casos obedecem as condições:

$$0 \leq u_{ik} \leq 1, \sum_{k=1}^C u_{ik} = 1 \forall i, 0 < \sum_{i=1}^N u_{ik} < N \forall k \quad (3.28)$$

3.3.5.4 Critério de Parada

Ainda relativo aos parâmetros inerentes aos algoritmos de agrupamento temos o critério de parada. Esse parâmetro diz respeito a maneira de como os algoritmos param seu processamento. Aqui, são apresentados 3 opções para critério de parada. Para todos eles, utiliza-se um limiar ε positivo com um valor baixo. O primeiro critério, é a estabilização da norma da diferença das matrizes de graus de pertinências formadas da iteração anterior $U^{(s-1)}$ e da iteração atual $U^{(s)}$, sendo s o contador de iterações. O valor dessa diferença deve ser menor que o limiar ε para que o algoritmo pare. Este limiar deve ser positivo maior que 0 e muito menor que 1. Este critério é expresso na equação (3.29).

$$\|U^{(s-1)} - U^{(s)}\| = \sum_{k=1}^K \sum_{i=1}^N (u_{ik}^{(s-1)} - u_{ik}^{(s)})^2 \quad (3.29)$$

O segundo critério segue o mesmo princípio do primeiro, porém a matriz considerada é a do protótipo. A norma da diferença entre as matrizes de protótipos da iteração anterior $V^{(s-1)}$ e atual $V^{(s)}$ deve ser menor que o limiar ε . O critério é apresentado na equação (3.30).

$$\|V^{(s-1)} - V^{(s)}\| = \sum_{k=1}^K \sum_{j=1}^m (v_{kj}^{(s-1)} - v_{kj}^{(s)})^2 \quad (3.30)$$

onde s é o contador de iterações e m o número de atributos do vetor \mathbf{v}_k .

O terceiro critério é mais direto. Esse critério interrompe o processamento do algoritmo quando os valores da diferença do valor da função de otimização da iteração anterior $J^{(s-1)}$ e da atual $J^{(s)}$ for menor que o limiar ε . Após aplicar o critério de parada, o algoritmo de agrupamento cessa o processamento formando as partições (grupos) e seus respectivos protótipos e a matriz final de graus de pertinência dos padrões a cada um dos grupos formados.

3.3.6 Resumo dos Algoritmos de Agrupamento Semi-Supervisionados

Todos os algoritmos explicados nesta seção se baseiam no algoritmo clássico de agrupamento *fuzzy*, o *Fuzzy C-Means* (FCM). Todos eles adicionam um segundo termo a função objetivo do algoritmo FCM que possui como objetivo otimizar a função levando em conta os rótulos conhecidos dos exemplos das bases de dados. Apenas o baseado em sementes funciona um pouco diferente dos demais, pois nele, o rótulo é levado em conta apenas na inicialização da matriz de protótipos quando apenas exemplos rotulados são utilizados. Nos demais algoritmos, os rótulos são utilizados para otimizar a função objetivo por algum método particular a cada um dos algoritmos.

No algoritmo de Pedrycz, o número de grupos deve ser igual ao número de classes. A otimização do termo supervisionado é realizada pela maior penalização caso um grau de pertinência seja pequeno em relação a classe que este padrão pertença. Assim, é realizada a diferença deste padrão a 1 e a multiplicação pela distância entre este padrão e o protótipo da classe ao qual ele pertence. No algoritmo de Bouchachia, a diferença para a otimização é entre o grau de pertinência e uma segunda otimização que leva em conta que uma classe pode representar mais de um grupo. Esta diferença é multiplicada pela distância entre o padrão e o protótipo do grupo. Para o algoritmo baseado em sementes, a otimização é realizada pela multiplicação do grau de pertinência de um padrão pela soma das distâncias dos padrões mais próximos a esse padrão. Para o algoritmo baseado em sementes, o número de grupos deve ser igual ao número de classes conhecidas, assim como o algoritmo de Pedrycz. No algoritmo proposto, a otimização é realizada pela diferença entre os graus de pertinência entre padrões que pertencem a uma mesma classe multiplicado pela distância entre esses dois padrões. Esta otimização parte da premissa que dois padrões que estão na mesma classe possuem uma pequena distância entre eles e desse modo diminui a diferença entre os graus de pertinência entre o padrão e o grupo que representa a classe que ele pertence. O algoritmo proposto possui a liberdade de poder representar uma classe por um ou mais grupos.

Podemos ainda falar das equações que utilizamos para calcular iterativamente os graus de pertinência e os protótipos dos algoritmos. No caso da equação de protótipos, os algoritmos de Pedrycz e o algoritmo proposto possuem uma equação igual a do algoritmo original FCM. Já os algoritmos de Bouchachia e baseado em sementes os cálculos utilizam as particularidades de cada um dos algoritmos. Para o caso do cálculo dos graus de pertinência, todos os algoritmos utilizam uma abordagem diferente. Porém, os algoritmos de Bouchachia e o algoritmo proposto precisam de informações de uma iteração anterior para realizar o cálculo.

O algoritmo proposto é uma abordagem nova e original de algoritmos de agrupamento semi-supervisionado. Ele possui algumas características que não são encontradas em nenhum algoritmo citado neste trabalho. Podemos citar aqui a otimização levando em conta apenas os padrões que são de uma mesma classe, e ainda, a utilização da distância entre estes dois padrões ao invés da distância entre o padrão e o protótipo do grupo. Ainda podemos citar, a ausência do termo de balanceamento para o termo supervisionado da função objetivo. Esta ausência elimina um parâmetro livre que o usuário tem que ajustar em outros algoritmos.

Assim, este capítulo detalhou a abordagem de agrupamento semi-supervisionado. Introduziu a abordagem de agrupamento e descreveu um algoritmo de agrupamento *fuzzy* clássico. Apresentou algoritmos de agrupamento semi-supervisionados já consolidados na literatura e

apresentou um algoritmo original de agrupamento semi-supervisionado. Portanto, com todos os algoritmos explicados em detalhes, vamos comparar estes algoritmos em termos de desempenho em tarefas de agrupamento e de classificação de padrões. A validação experimental é explicada no próximo capítulo.

Validação Experimental

Neste capítulo são apresentados detalhes dos experimentos realizados com os algoritmos de aprendizagem semi-supervisionada. A Seção 4.1 apresenta a metodologia de validação proposta nesse trabalho para algoritmos de agrupamento semi-supervisionados. Logo após, a Seção 4.2 descreve as bases de dados utilizadas nos experimentos. Por fim, a Seção 4.3 detalha cada experimento realizado nesse trabalho.

4.1 Metodologia de Validação

Nesta seção, a metodologia de validação para comparar o desempenho dos algoritmos semi-supervisionados é descrita. Essa metodologia utiliza uma adaptação da validação cruzada por k vezes para métodos semi-supervisionados. Essa adaptação é inspirada no trabalho de Costa et al [CCdS03]. A precisão dos resultados obtidos é medida comparando-se a partição de teste com uma partição classificada previamente utilizando uma taxa de acerto, explicada na Seção 4.1.2, ou um índice externo, explicado na Seção 4.1.3. Um intervalo de confiança é construído para validar estatisticamente os experimentos. É explicado na Seção 4.1.4.

4.1.1 Validação Cruzada

Em métodos supervisionados, a comparação entre métodos de aprendizagem é, geralmente, realizada através da relevância estatística da diferença entre médias das taxas de erros de classificação da partição de testes resultantes dos métodos avaliados. Para avaliar a média da taxa de erro, vários conjuntos de dados distintos são necessários. No entanto, o número de conjuntos de dados disponíveis é muitas vezes limitada.

Uma das abordagens para superar essa limitação é a validação cruzada por k vezes [Mit97]. Esse método consiste numa divisão da base de dados em k grupos de tamanhos aproximadamente iguais. Os padrões contidos nos grupos $k - 1$ são utilizados para treinar o algoritmo e os padrões do grupo k são utilizados para testar o classificador. Esse processo é repetido para todas as combinações de $k - 1$ grupos. O desempenho do algoritmo é medido através das médias das taxas de erro dos resultados das $k - 1$ iterações.

Assim como no método supervisionado, na aprendizagem não supervisionada, quando existe uma classificação prévia disponível do conjunto de dados, a relevância estatística da diferença entre médias de taxa de erros de classificação do conjunto de testes também pode ser utilizada. O método de validação cruzada pode ser adaptado para essa finalidade [CCdS03]. O conjunto de treinamento é apresentado ao método de agrupamento, o resultado é uma partição

(partição de treinamento). Depois, a técnica de centróide mais próximo é utilizada para construir um classificador a partir da partição de treinamento. A técnica de centróide mais próximo calcula a proximidade de cada padrão do conjunto de teste ao protótipo, ou o centro, de cada agrupamento formado na partição de treinamento. Desta forma, cada padrão do conjunto de teste é atribuído ao agrupamento cuja a proximidade seja a menor calculada. Depois, o conjunto de teste é comparado com a partição prévia utilizando um índice externo.

No meio termo das duas aprendizagens anteriores, na aprendizagem semi-supervisionada com métodos baseados em agrupamento, adaptamos a metodologia da validação de métodos não supervisionados. Uma das adaptações é em relação à formação da partição de treinamento. Como o objetivo de um método semi-supervisionado é melhorar a classificação com poucos dados rotulados disponíveis, os experimentos possuem diferentes porcentagens destes dados rotulados para a formação da partição de treinamento. Nos conjuntos de treinamento, as porcentagens de dados rotulados constituem de (0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100) do total da base disponível. Os padrões que irão ter seus rótulos disponibilizados são escolhidos aleatoriamente. Desse modo, para cada uma das diferentes configurações de dados rotulados, a base é particionada em k grupos de tamanhos aproximadamente iguais.

Após essa etapa, a técnica de centróide mais próximo é adaptada. No original proposto por Costa [CCdS03], o padrão pertence ao grupo cuja similaridade for maior. A similaridade é calculada utilizando a forma original do algoritmo, ou seja, pela equação do algoritmo que calcula a similaridade entre o exemplo e o protótipo.

Para os algoritmos semi-supervisionados, o padrão não pertence apenas a um grupo, ele pode pertencer a mais de um deles de acordo com o grau de pertinência calculado pelo algoritmo original. Porém, para que a taxa de acerto seja calculada, um padrão deve pertencer à apenas um grupo. Desse modo, o padrão irá pertencer ao grupo cujo o grau de pertinência seja maior.

Após a adaptação do centróide, nos deparamos com casos onde o cálculo da similaridade entre um padrão e o centróide na forma proposta do algoritmo não pode ser realizada num conjunto de testes. Por exemplo, há casos onde esse cálculo precisa de informações de uma iteração anterior. No conjunto de treinamento, essa informação é disponível, pois o treinamento é realizado em várias iterações, mas no caso de um conjunto de testes, há apenas uma iteração para calcular a similaridade entre esses padrões e as partições formadas no treinamento. Desse modo, atribuímos um valor igual para cada grau de pertinência do padrão ao grupo da pertinência da iteração anterior, a soma dos graus de pertinência de um padrão é igual a 1. O procedimento do algoritmo é explicado logo abaixo.

Formalmente, seja D o conjunto de dados, n o número de clusters; L_i a porcentagem i -ésima de dados rotulados; F_i , o i -ésimo conjunto de teste; R_i , a partição resultante do conjunto treino $D - F_i$; C_i o conjunto de centróides da partição R_i ; T_i a partição resultante do conjunto de teste F_i ; e P_i uma partição prévia com os objetos de F_i , para $i = 1, \dots, K$, então, o a validação cruzada por k vezes semi-supervisionada funciona como segue:

A idéia geral da validação cruzada é observar quão bem, os dados, a partir de um conjunto independente F_i estão agrupados, dada a formação dos resultados. O objetivo desse procedimento é obter k observações da acurácia dos métodos semi-supervisionados com respeito à uma classificação prévia, tudo isso utilizando conjuntos de testes independentes.

Entrada: Base de Dados D

```

1 para cada  $L_i$  faça
2   Rotular  $D$  de acordo com a porcentagem  $L_i$ ;
3   Dividir  $D$  em  $k$  conjuntos  $F_i$  de tamanhos aproximadamente iguais;
4   para  $i \leftarrow 1$  até  $k$  faça
5     Aplicar o método de clustering para o conjunto  $D - F_i$  para obter a partição
        $R_i$  com  $n$  clusters como resultado;
6     Calcular os  $n$  centróides dos clusters em  $R_i$ , formando  $C_i$ ;
7     Calcular os graus de pertinência entre os centróides em  $C_i$  e os padrões em
        $F_i$  utilizando a equação original do algoritmo;
8     Atribuir os objetos de  $F_i$ , de acordo com o maior grau de pertinência em  $C_i$ 
       para obter a partição  $T_i$  como resultado;
9     Medir a coesão das partições  $T_i$  e  $P_i$  com um índice externo;
10   fim
11 fim

```

Algoritmo 6: Validação Cruzada

4.1.2 Taxa de Acerto

É importante ressaltar o modo que um grupo é rotulado por uma classe pré-definida. Os grupos são formados pelo algoritmo, e a classe que esses grupos representam são desconhecidas. Depois dos grupos formados pelo algoritmo, uma matriz de confusão é construída. A matriz de confusão é caracterizada pelas linhas l que representam o grupo atribuído pelo algoritmo e a coluna c , que representa o grupo ao qual o padrão pertence originalmente. Assim, o valor da matriz da linha 1' e da coluna 2, representa a quantidade de padrões que foram alocados para o grupo 1' e pertencem originalmente a classe 2. Após essa etapa, as diferentes combinações de linha e coluna da matriz são testadas de forma que a taxa de acerto seja máxima. Um exemplo dessa etapa é apresentado logo abaixo. Considere a Tabela 4.1.2 abaixo:

Tabela 4.1 Matriz de Confusão

i/j	1	2	3
1'	50	20	10
2'	10	20	30
3'	0	60	0

As combinações são apresentadas logo abaixo, representando o valor da linha versus a coluna. A soma de cada combinação é mostrada na coluna final:

1-1' 2-2' 3-3' 70
 1-1' 2-3' 3-2' 140
 1-2' 2-1' 3-3' 30
 1-2' 2-3' 3-1' 50

1-3' 2-1' 3-2' 80

1-3' 2-2' 3-1' 30

A soma da combinação da segunda linha é a maior. Desse modo, essa combinação é escolhida. O grupo 1' representa a classe 1, o grupo 3' representa a classe 2 e o grupo 2' representa a classe 3. Assim, a taxa de acerto é realizada pela equação (4.1). Nesse exemplo, o valor final da taxa de acerto é 70,0%.

$$Tx = \frac{\text{Soma das Combinações} * 100}{\text{Total de Padrões}} \quad (4.1)$$

4.1.3 Índice Externo - Índice de Rand Corrigido

Há uma série de índices externos definidos na literatura, tais como Hubbert, Jaccard, Rand e Rand Corrigido [JD88]. Uma característica da maioria destes índices é que eles podem ser sensíveis ao número de classes nas partições ou a distribuições de elementos nos agrupamentos. Por exemplo, alguns índices possuem uma tendência a apresentar valores mais elevados para as partições com mais classes (Hubbert e Rand), outros para as partições com um menor número de classes (Jaccard) [Dub87]. O índice de Rand Corrigido, possui seus valores corrigidos de acordo com acertos nas comparações das partições, por isso não possui nenhuma destas características indesejáveis [MC86]. Assim, o índice Rand corrigido é o índice externo utilizado na validação do método proposto neste trabalho.

Um índice externo de adequação de partições avalia o grau em que duas partições de n objetos se correspondem. Uma partição é proveniente do resultado de um algoritmo de agrupamento. A segunda partição é construída previamente, independente dos dados e da primeira partição, com os rótulos das classes. Em [HA85] cuidadosamente define-se vários índices com o objetivo de comparar duas partições. Em teoria, quando as partições são "independentes", no sentido de que uma depende dos valores dados e a outra não, a distribuição de alguns destes índices pode ser estabelecida. Todos esses índices são obtidos a partir da tabela de contingência e construídos a partir de duas partições mostradas na Tabela 4.1.3. As duas partições de n objetos são U e V .

Tabela 4.2 Tabela de contingência para duas partições

	v_1	v_2	...	v_c	
u_1	n_{11}	n_{12}	...	n_{1c}	$n_{1.}$
u_2	n_{21}	n_{22}	...	n_{2c}	$n_{2.}$
...	
u_R	n_{R1}	n_{R2}	...	n_{Rc}	$n_{R.}$
	$n_{.1}$	$n_{.2}$		$n_{.c}$	$n_{..} = n$

Seja $U = u_1, \dots, u_r, \dots, u_R$, $V = v_1, \dots, v_c, \dots, v_C$ e a entrada n_{ij} na Tabela 4.1.3, o número de objetos que estão tanto no grupo u_i quanto no grupo v_i . O termo $n_{i.}$ representa o número de objetos no grupo u_i ou a soma dos objetos da linha i -ésima e $n_{.j}$ é o número de objetos no grupo

v_j , e n é o número total de objetos nas partições. O índice Rand Corrigido (CR) é definido de acordo com a equação (4.2) [HA85].

$$CorrectedRand = \frac{\sum_i^R \sum_j^C (\frac{n_{ij}}{2}) - (\frac{n}{2})^{(-1)} \sum_i^R (\frac{n_{i\cdot}}{2}) \sum_j^C (\frac{n_{\cdot j}}{2})}{\frac{1}{2} [\sum_i^R (\frac{n_{i\cdot}}{2}) + \sum_j^C (\frac{n_{\cdot j}}{2})] - (\frac{n}{2})^{(-1)} (\frac{n_{i\cdot}}{2}) \sum_j^C (\frac{n_{\cdot j}}{2})} \quad (4.2)$$

Esse índice possui um valor no intervalo $[-1, 1]$, onde o valor 1 indica uma coesão perfeita entre as partições e -1 indica que não existe coesão entre as partições. Os valores próximos a 1 indicam boa coesão. O trabalho de [MC86] indica que valores próximos de 0 são gerados por partições geradas por dados aleatórios e abaixo de 0.05 indica partições geradas por acaso.

4.1.4 Intervalo de Confiança

Intervalo de confiança é uma maneira para calcular uma estimativa de um parâmetro desconhecido. Muitas vezes pode funcionar como um teste de hipótese. A idéia é construir um intervalo de confiança para o parâmetro desconhecido com uma probabilidade $(1 - \alpha)$ de que o intervalo contenha o verdadeiro parâmetro [BS01].

O nível de significância α é o erro que se comete ao afirmar que, por exemplo, 95% das vezes o intervalo $\theta_1 < \theta < \theta_2$ contém θ . Nesse caso o erro seria de 5%. O intervalo $\theta_1 < \theta < \theta_2$ representa o intervalo de confiança, sendo que θ_1 é o limite inferior, θ_2 o limite superior e θ é o parâmetro desconhecido. A precisão do estimador é representada pela metade do intervalo, ou seja, $\theta - \theta_1$ ou $\theta_2 - \theta$.

Assim, quanto maior o intervalo de confiança, mais confiante estaremos que o intervalo vai conter o valor real de θ . Por outro lado, quanto maior o intervalo, menos informação teremos sobre o verdadeiro valor de θ . Sendo assim, numa situação ideal teremos um intervalo relativamente curto com uma confiança alta.

O intervalo de confiança é calculado através da média (\bar{X}) e do desvio padrão σ de uma amostra de tamanho n . O intervalo de confiança $(1 - \alpha)\%$ da média μ é dado por

$$\bar{X} - Z_{\alpha/2} \frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{X} + Z_{\alpha/2} \frac{\sigma}{\sqrt{n}} \quad (4.3)$$

onde $Z_{\alpha/2}$ é obtido da distribuição normal reduzida. A média é calculada por

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n} \quad (4.4)$$

e o desvio padrão calculado por

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{X})^2}{n - 1}} \quad (4.5)$$

Para exemplificar, considere um algoritmo em que aplicou-se a validação cruzada por 10 vezes, repetida 30 vezes. Assim, teremos os valores da taxa de acerto dos quais obtemos a média, dada por 90.0, e o desvio padrão, dado por 6.7. Com intervalo de confiança de 95% teremos:

Intervalo de confiança (95%): $89.2388 < \mu \leq 90.7612$

Também podemos utilizar intervalos de confiança como teste de hipótese, já que a construção de intervalo de confiança advém da teoria dos testes de hipótese. O teste de hipótese é uma regra de decisão para aceitar ou rejeitar uma hipótese estatística a ser testada com base nos elementos amostrais. No teste de hipótese designa-se, geralmente, H_0 (hipótese nula) a hipótese estatística a ser testada e por H_1 a hipótese alternativa. A rejeição de H_0 implicará a aceitação de H_1 . A hipótese alternativa geralmente representa a suposição que o pesquisador quer provar, sendo H_0 formulada com o propósito expresso de ser rejeitada.

Quando comparamos amostras de populações, construímos o intervalo de confiança para cada uma das amostras. Assim, podemos dizer que a hipótese nula H_0 pode ser aceita caso os intervalos de confiança tenham alguma intersecção, por mínima que seja, ou rejeitada caso os intervalos não tenham nenhuma intersecção. Desse modo, considere a Figura 4.1 onde plotamos os intervalos de confiança de 4 algoritmos num gráfico e podemos avaliar as hipóteses. Nesse gráfico, podemos afirmar, por exemplo, que a hipótese de o algoritmo 1 ter desempenho melhor que o algoritmo 2 (hipótese alternativa) é verdadeira, pois não há intersecção entre os intervalos de confiança dos dois algoritmos. Porém, a hipótese de o algoritmo 1 ser melhor ou pior que algoritmo 3 (hipóteses alternativas) não é verdadeira, pois há intersecção entre os intervalos de confiança desses dois algoritmos e assim a hipótese de que o algoritmo 1 possui o mesmo desempenho do algoritmo 3 (hipótese nula) é aceita.

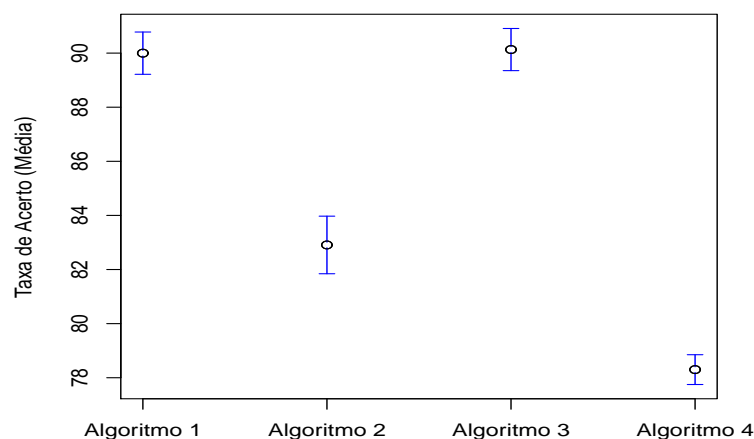


Figura 4.1 Intervalos de Confiança (95%)

4.2 Bases de Dados

Nesta seção, as bases de dados utilizadas nesse trabalho são apresentadas. As bases escolhidas para este trabalho possuem como principal característica serem criadas para participar de experimentos de classificação ou reconhecimento de padrões. Ao todo, foram escolhidas 4 bases de dados de *benchmark* retiradas do repositório digital UCI (<http://archive.ics.uci.edu/ml/>) mais uma base de dados sintética. Nenhuma dessas bases possuem padrões faltando, ou seja, são bases completas. As bases são detalhadas nos tópicos a seguir.

4.2.1 Base de Dados Iris

A base de dados *Iris* é uma das bases de dados mais utilizadas em reconhecimento de padrões. Essa base consiste num conjunto de 150 padrões que descrevem 3 tipos de plantas da família *Iris*. Existem 4 atributos que descrevem as medidas de largura e comprimento de pétalas e sépalas da planta. Existem 50 padrões para representar cada uma das espécies. A tarefa é formar 3 grupos que representam cada um dos tipos de *Iris*.

4.2.2 Base de Dados Diabetes

A base de dados *Diabetes* consiste numa coleção de padrões que indicam a existência da diabetes ou não. Essa base contém um conjunto de 768 padrões, sendo que 500 representam a ausência de diabetes e os outros 268 representam a presença da diabetes. Existem 8 atributos com características principalmente sanguíneas e corporais, mais a idade. O objetivo é informar se um padrão pertence ao grupo de diabéticos ou não.

4.2.3 Base de Dados Wine

A base de dados *Wine* é uma das bases de dados mais utilizadas para testar novos algoritmos. Essa base consiste num conjunto de 178 padrões que descrevem 3 tipos de vinhos. Existem 13 atributos que descrevem fatores químicos desses vinhos. As classes são desbalanceadas, sendo 59 padrões para representar o vinho 1, 71 para o vinho 2 e 48 padrões para o vinho 3. A tarefa é formar 3 grupos que representem cada um dos tipos de vinho.

4.2.4 Base de Dados Spam

A base de dados *Spam* consiste de uma coleção de mensagens eletrônicas chamadas de *e-mails*. O conteúdo dessas mensagens é classificado como *spam* (lixo eletrônico) ou não *spam*. A base contém 4601 mensagens descritas através de 57 atributos. Os atributos, em sua maioria representam o TF-IDF *Term-frequency Inverse-Document-Frequency* [SB88] de uma determinada palavra. A tarefa do algoritmo é classificar em 2 grupos, um representando o grupo de *spam* e outro representando o grupo de mensagens não *spam*.

4.2.5 Base de Dados Sintética

A base de dados sintética é gerada de acordo com algumas características estatísticas, chamados de média e desvio padrão (μ, Σ). Esses parâmetros foram retirados do trabalho de Bouchachia [BP06] que também utilizou essa base de dados para testar o poder de classificação de seu algoritmo. Essa base é formada por duas classes mostradas na Tabela 4.2.5 e plotadas na Figura 4.2. A primeira classe H_1 consiste de um grupo e a segunda classe consiste de dois grupos descritas por dois atributos (chamados de att na tabela). Cada grupo é formado por 100 padrões.

Tabela 4.3 Base de dados Sintética

Características		$\mu(\text{att } 1)$	$\mu(\text{att } 2)$	$\Sigma(\text{att } 1)$	$\Sigma(\text{att } 2)$
Classes	H_1	3.0	1.0	7.0	0.5
	H_2	4.0	4.5	1.0	1.0
		2.0	-2.5	1.0	1.0

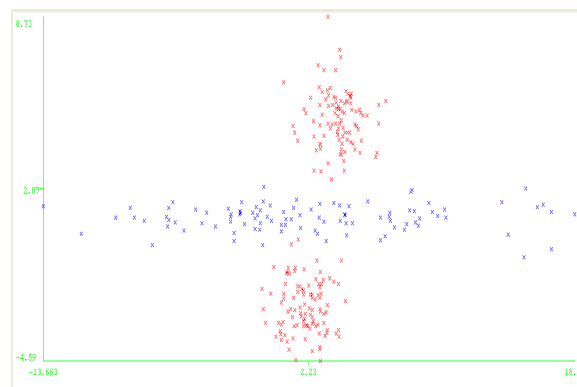


Figura 4.2 Base de dados Sintética (Pontos em azul: Classe 1; Pontos em vermelho: Classe 2)

4.3 Experimentos

Os experimentos são divididos em 3 tipos. O primeiro, avalia o poder de classificação do novo algoritmo. O algoritmo proposto é comparado com outros algoritmos totalmente supervisionados considerados clássicos na literatura. A descrição desse experimento é realizada na Seção 4.3.1. O segundo e terceiro experimentos são detalhados na Seção 4.3.2. Neles, o novo algoritmo é comparado com outros algoritmos de agrupamento semi-supervisionado nas tarefas de classificação e agrupamento. Na primeira tarefa (agrupamento), toda a base disponível é utilizada para treinar e testar os algoritmos e na segunda tarefa (classificação), a validação cruzada por 10 vezes é utilizada.

4.3.1 Experimento 1: Classificação

Nesta seção o algoritmo proposto é utilizado como se fosse um algoritmo supervisionado. O objetivo desse experimento é validar a classificação induzida, então vamos utilizar aqui apenas dados rotulados para treinar o algoritmo. Este experimento é baseado no experimento realizado por Bouchachia [BP06] com o mesmo objetivo.

O desempenho do algoritmo é medido da forma:

$$A = \frac{\text{Número de padrões atribuídos corretamente}}{\text{Tamanho do conjunto de Teste}} \quad (4.6)$$

Para um acerto ser calculado, um padrão precisa ser adicionado ao grupo correto. Para um padrão ser atribuído a um grupo, o grau de pertinência do padrão a esse grupo deve ser maior que o grau de pertinência para os outros grupos. Desse modo, o padrão é atribuído ao grupo que possuir o maior grau de pertinência. Esse grupo representa uma classe, então essa classe é chamada de classe vencedora. Para ilustrar esse processo, a Figura 4.3 apresenta todo o processo de classificação.

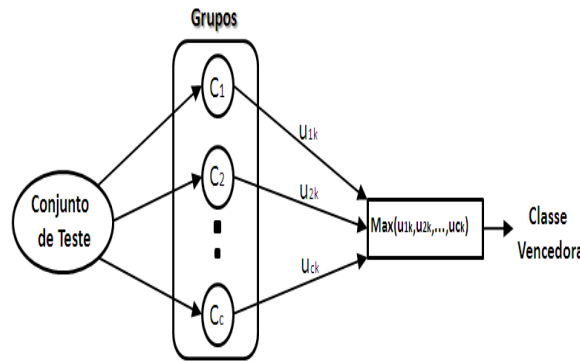


Figura 4.3 Usando o algoritmo de agrupamento como um classificador fuzzy

Os algoritmos supervisionados comparados nesse experimento foram a rede neural *multi-layer perceptron* (MLP), o algoritmo probabilístico *naive bayes* e o algoritmo de aprendizagem de regras Part. As configurações de cada algoritmo são explicadas nos parágrafos abaixo.

As redes MLP [Bis95] são amplamente utilizadas para resolução de problemas não lineares. Uma MLP consiste de uma série de camadas, cada camada consiste de um número de nós. Cada nó calcula sua ativação, utilizando diversas funções ativação. Neste experimento, se aplica a função sigmóide (sigm). O número de camadas, o número de nós por camada e taxa de aprendizagem são parâmetros. Aqui, foram realizados 6 experimentos para cada uma das base de dados. Os parâmetros de cada experimento são mostrados na Tabela 4.3.1 abaixo. Os experimentos de 1 a 4 são sempre executados com os mesmos parâmetros. Nos experimentos 5 e 6, as quantidade de nós na camada escondida são executados de acordo com o melhor resultado encontrado nos 4 primeiros experimentos.

O segundo tipo de algoritmo pertence a classe dos algoritmos probabilísticos. Os algoritmos probabilísticos consideram o valor do estado da classe como a probabilidade $P(\frac{c}{d})$. Desse modo

Tabela 4.4 Parâmetros dos experimentos para MLP

Experimentos	1	2	3	4	5	6
Número de Camadas Escondidas	1	1	1	1	1	1
Nós por camada	1	2	3	5	Melhor Resultado	Melhor Resultado
Taxa de Aprendizagem	0.1	0.1	0.1	0.1	0.3	0.01

a probabilidade que o dado d pertença a classe c é calculada através do teorema de Bayes [DHS01]:

$$P\left(\frac{c}{d}\right) = \frac{P\left(\frac{d}{c}\right)P(c)}{P(d)} \quad (4.7)$$

A probabilidade $P(d)$ não precisa ser calculada porque é constante para todas as classes. Para calcular $P\left(\frac{d}{c}\right)$ é preciso fazer algumas suposições sobre a estrutura dos dados. A representação do vetor de características é dada por $d = (w_1, w_2, \dots)$ e todos os atributos são independentes. Essa suposição é chamada de *bayes ingênuo* porque relaxa na verificação de suposições como a dependência dos atributos. Assim, a probabilidade é calculada como:

$$P\left(\frac{d}{c}\right) = \prod_j P\left(\frac{w_j}{c}\right) \quad (4.8)$$

O último de algoritmo supervisionado avaliado é um algoritmo de aprendizagem de regras e é do tipo simbólico. Os classificadores simbólicos são mais fáceis de entender por humanos. A decisão é realizada escolhendo a melhor regra do conjunto de todas as regras (regras que classificam todos os exemplos de treinamento) que otimizam um critério. Cada padrão de treinamento é visto por uma regra que indica a presença ou ausência de características e é associado a uma classe. O aprendizado aplica uma série de generalizações compactando as regras enquanto mantém a abrangência dessas regras. No final, é realizada uma etapa de poda e troca-se abrangência por generalização. As regras de decisão variam em termo de métodos, heurísticas e critérios aplicados na poda e na generalização. O algoritmo PART[FW98] utiliza a abordagem dividir pra conquistar. Constrói uma árvore de decisão parcial C4.5 [Qui93] e a cada iteração transforma o melhor ramo numa regra. Esse algoritmo não possui parâmetros livres e desse modo realizamos apenas um experimento para cada base de dados. No próximo parágrafo iremos explicar os parâmetros do algoritmo proposto.

Por fim, agora explicaremos alguns parâmetros do algoritmo proposto. O primeiro parâmetro é a maneira que os padrões são inicializados. Nesse experimento foi utilizada a abordagem de inicialização *hard* explicada no capítulo anterior. O segundo parâmetro testado nesse algoritmo diz respeito a etapa de teste da validação cruzada. Esse algoritmo possui a característica de no cálculo do grau de pertinência, precisar do valor do grau de pertinência da iteração anterior. Desse modo, na etapa de rotulação do conjunto de teste, o cálculo da similaridade entre o padrão pertencente ao conjunto de teste e o protótipo da partição formada no treinamento é calculada pela equação original do algoritmo. Assim, o grau de pertinência da etapa anterior é

atribuído pelo cálculo da divisão de 1 pelo número de grupos formados. Além desses parâmetros, pode-se testar diferentes critérios de parada, porém aqui também decidimos manter esse parâmetro fixo. Escolhemos o critério de parada que estabiliza a diferença entre os resultados da diferença entre o resultado da função de otimização da iteração anterior e o da atual.

Para finalizar, as configurações gerais do experimento são explicadas. Nesse experimento utilizamos 4 bases de dados apresentadas na Seção 4.2: Iris, Diabetes, *Wine* e a base de dados Sintética. Os experimentos utilizam a validação cruzada por 10 vezes repetida 30 vezes. Para cada iteração da validação cruzada, a inicialização dos padrões é repetida por 20 vezes. Dessas 20 repetições, o resultado escolhido é o da iteração que conseguir obter o menor valor da função objetivo. Desse modo, são gerados os valores para obtenção da média e do desvio padrão para cada uma das configurações do experimento. De posse desses valores, construímos intervalos com 5% de confiança com a finalidade de comparar os algoritmos estudados.

4.3.2 Agrupamento Semi-Supervisionado

Depois de estudar o poder de classificação do algoritmo com a base totalmente rotulada, nesta seção vamos focar na classificação/agrupamento de dados parcialmente rotulados. A metodologia utilizada nesse trabalho possui como objetivo principal validar os rótulos encontrados pelo algoritmo proposto com diferentes porcentagens de dados rotulados disponível para o treinamento do algoritmo nas tarefas de classificação e de agrupamento. A metodologia utilizada nesse trabalho foi baseada nos trabalhos de Amini e Gallinari [AG05] e de Pedrycz [PW97] e é composta por 2 experimentos chamados no texto de experimentos 1 e 2 respectivamente. Para que o objetivo seja alcançado, os padrões da base de dados são divididos em 2 conjuntos, o conjunto de dados rotulados e o conjunto de dados não rotulados. O conjunto de dados rotulados contém uma certa porcentagem de exemplos rotulados do total da base de dados. A porcentagem de dados rotulados variam de (0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100) do total da base. O restante do conjunto de dados faz parte do segundo conjunto, o de dados não rotulados. Assim, para os valores respectivos de base rotulada, esses conjuntos utilizam as porcentagens (100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 0) do total da base. Os experimentos são realizados submetendo os dois conjuntos de dados ao algoritmo de agrupamento semi-supervisionado com as devidas proporções de cada um dos conjuntos. Desse modo, observamos a influência da quantidade de dados rotulados e não rotulados no desempenho dos algoritmos. As etapas dos experimentos são explicadas abaixo:

1. **Particionamento dos Dados Originais:** Os dados originais são particionados em dois subconjuntos (rotulados e não rotulados). Essa primeira etapa é igual para os experimentos 2 e 3. Nesta etapa, os dados originais são particionados de forma estratificada. Dessa maneira, a distribuição de dados das classes original é mantida, ou seja, se existe originalmente 30% de uma classe *A* e 70% de uma classe *B*, esses valores são mantidos nos conjuntos de dados rotulados e não rotulados. Os exemplos que compõem cada um dos dois conjuntos são selecionados aleatoriamente na base de dados.
2. **Execução do Algoritmo:** Execução do algoritmo tendo como entrada os dois subconjuntos obtidos na etapa 1 e tendo como saída uma partição rotulada. Nesta etapa, o segundo

experimento executa a tarefa de classificação utilizando validação cruzada. No terceiro experimento, é realizada a tarefa de agrupamento que utiliza toda a base de dados para treinar e testar o algoritmo.

3. **Análise dos Rótulos:** Nesta etapa, os rótulos obtidos são comparados com os rótulos originais. A qualidade dos grupos formados também são analisados. Como resultado temos a taxa de acerto dos rótulos gerados pelo algoritmo para o caso da tarefa de classificação e uma taxa de qualidade da formação dos grupos chamada de índice de rand corrigido para o caso da tarefa de agrupamento. Um intervalo de 5% de confiança é construído para avaliar o desempenho dos algoritmos.

As etapas dos experimentos 2 e 3 são parecidas, além disso, os algoritmos empregados nos dois experimentos são os mesmos. Esses algoritmos já foram explicados no capítulo anterior. São eles:

1. **Algoritmo Baseado em Sementes:** Algoritmo que modifica a maneira que o Fuzzy C-means original [Bez81] inicializa o grupo, utilizando dados onde os rótulos são conhecidos. Para maiores detalhes ver [Bou07].
2. **Algoritmo de Bouchachia e Pedrycz:** Algoritmo que modifica a função objetivo do algoritmo Fuzzy C-Means [Bez81]. Para maiores detalhes, ver [BP06].
3. **Algoritmo de Pedrycz e Waletzky:** Algoritmo que também modifica a função objetivo do algoritmo Fuzzy C-Means [Bez81]. Para maiores detalhes, ver [PW97].

Mesmo com características semelhantes entre os experimentos 2 e 3, a abordagem utilizada em cada um deles é diferente devido a particularidade de cada uma das abordagens. Existem diferenças na execução de cada um dos dois experimentos. As subseções subsequentes explicam os detalhes da execução de cada um desses experimentos. As diferenças entre o segundo e terceiro experimentos são explicadas na Seção 4.3.2.1 e na Seção 4.3.2.2 respectivamente.

A Figura 4.4 ilustra o processo de rotulação realizado nesse experimento. Primeiro, os dados são particionados em conjuntos rotulados e não rotulados. O conjunto contendo triângulos representa os dados não rotulados, enquanto o conjunto contendo estrelas vermelhas e sinais positivos verdes representam os dados rotulados de duas classes respectivamente. Depois, esses dados são inicializados com uma das três abordagens: *fuzzy*, *hard* ou agrupamento. Nos experimentos escolhemos a inicialização *hard*. Então, o algoritmo utiliza os dois conjuntos inicializados como entrada em sua execução. O resultado do algoritmo são protótipos que representam as duas classes com estrelas vermelha e verde respectivamente. Os dados do conjunto de teste são atribuídos aos grupos de acordo com a similaridade do protótipo. A similaridade é calculada utilizando a equação do grau de pertinência pela fórmula original do algoritmo. Então, nesse exemplo são formadas 3 partições que representam duas classes representadas pelos protótipos das estrelas vermelha e verde respectivamente juntamente com os dados do conjunto de testes, agora rotulados.

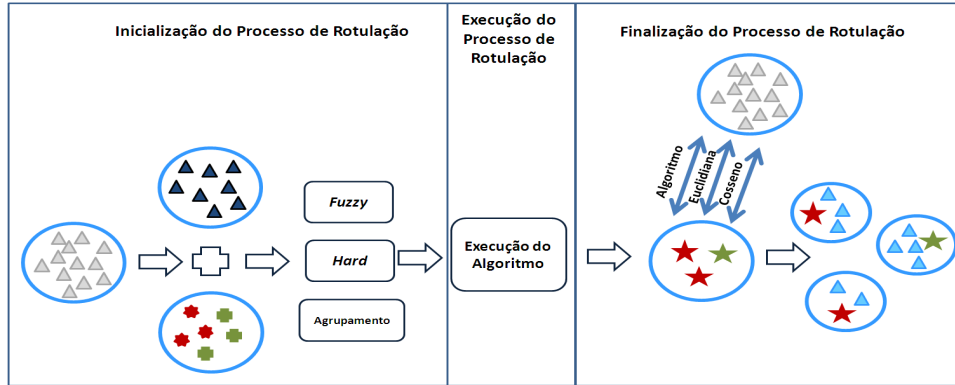


Figura 4.4 Processo de Rotulação por Agrupamento

4.3.2.1 Experimento 2: Tarefa de Classificação Semi-Supervisionada

Este experimento emprega a validação cruzada para avaliar os algoritmos estudados. A validação cruzada emprega dois conjuntos de dados separados, um para treinamento e outro para teste. Nesta abordagem a base de dados é dividida em k conjuntos de tamanhos aproximadamente iguais, mantendo também a proporção das configurações de dados rotulados e não rotulados.

Nos experimentos, os parâmetros inerentes aos algoritmos de agrupamento são iguais para todos os algoritmos, para manter a igualdade de condições e de parâmetros aos quais os algoritmos são submetidos. O primeiro parâmetro é a maneira que os padrões são inicializados. Assim como no experimento de classificação a inicialização *hard* é utilizada. Além desse parâmetro, o critério de parada empregado por todos os algoritmos é o da estabilização da diferença entre os resultados obtidos pela função objetivo da etapa anterior e da etapa atual.

Neste experimento são avaliadas 4 bases de dados apresentadas na Seção 4.2: Iris, Diabetes, Wine e Spam. Os experimentos utilizam a validação cruzada por 10 vezes repetida 30 vezes. Para cada iteração da validação cruzada, a inicialização dos padrões é repetida por 20 vezes para que os parâmetros de inicialização do algoritmo tenham pouca influência no resultado final do algoritmo. Dessas 20 repetições, o resultado escolhido é o da iteração que conseguir obter o menor valor da função objetivo. Desse modo, são gerados os valores dos quais obtemos a média e o desvio padrão para cada uma das configurações do experimento. De posse desses valores, calculamos a média da taxa de acerto e construímos intervalos com 5% de confiança com a finalidade de comparar os algoritmos estudados.

4.3.2.2 Experimento 3: Tarefa de Agrupamento Semi-Supervisionada

Este experimento possui como principal característica a utilização de toda a base de dados disponível para o treinamento e para o teste do algoritmo, respeitando as configurações já apresentadas dos conjuntos de dados rotulados e não rotulados.

Os parâmetros inerentes aos algoritmos de agrupamento são iguais para todos os algoritmos envolvidos no experimento, para que os testes realizados sejam justos, em igualdade de

condições e de parâmetros. O primeiro parâmetro é a maneira que os padrões são inicializados. Assim como no experimento de classificação, a inicialização *hard* é utilizada. Além desse parâmetro, o critério de parada empregado por todos os algoritmos é o da estabilização da diferença entre os resultados obtidos pela função objetivo da etapa anterior e da etapa atual.

Por fim, as configurações gerais do experimento são explicadas. Neste experimentos são avaliadas 4 bases de dados apresentadas na Seção 4.2: *Iris*, *Diabetes*, *Wine* e *Spam*. O experimento é repetido 100 vezes para obter uma significância estatística. Para cada iteração do experimento, a inicialização dos padrões é repetida 20 vezes para que o melhor resultado dessas repetições seja escolhido. O resultado selecionado é o da iteração cuja função objetivo obtenha o menor valor em relação aos demais. De posse dos resultados, calculamos a média do índice de rand corrigido para avaliar a partição rotulada pelo algoritmo em relação partição original da base de dados. São calculados 3 índices de rand corrigido, o índice de rand corrigido global, que utiliza toda a base de dados em seu cálculo, o índice de rand da base rotulada, que utiliza apenas exemplos que eram inicialmente rotulados e o índice de rand da base não rotulada, que utiliza apenas a base que inicialmente não era rotulada. Desse modo, podemos observar o comportamento do algoritmo separadamente na base de dados total e nos dois subconjuntos de dados rotulados e não rotulados. Depois, intervalos com 5% de confiança são construídos para comparar o desempenho dos algoritmos de agrupamento semi-supervisionado na tarefa de agrupamento.

Neste capítulo explicamos em detalhes a metodologia empregada em todos os experimentos realizados neste trabalho. A execução dos experimentos produziram resultados que são apresentados através de gráficos, explicações e discussões no próximo capítulo.

Resultados e Discussão

Os resultados da análise comparativa entre os algoritmos explorados neste trabalho são apresentados e analisados neste capítulo. A Seção 5.1 descreve os resultados obtidos na análise comparativa do desempenho na tarefa de classificação entre o algoritmo proposto e algoritmos totalmente supervisionados. A Seção 5.2 apresenta os resultados obtidos na análise comparativa dos métodos de agrupamento semi-supervisionados utilizando validação cruzada na tarefa de classificação. Enquanto a Seção 5.3 descreve os resultados obtidos na análise comparativa dos métodos de agrupamento semi-supervisionados na tarefa de agrupamento.

5.1 Do Agrupamento a Classificação

Nesta seção, os experimentos são para avaliar o desempenho na tarefa de classificação do algoritmo proposto. O algoritmo proposto é comparado com os algoritmos totalmente supervisionados: Bayes, MLP e Part. Para o algoritmo proposto a base disponível no treinamento é totalmente rotulada. Os experimentos utilizaram validação cruzada por 10 vezes, repetida 30 vezes. Assim, calculamos a média e o desvio de padrão para cada um dos experimentos. Os resultados são apresentados em 2 gráficos para cada base. No primeiro, as médias da taxa de acerto são apresentadas para cada um dos algoritmos. No segundo gráfico, o intervalo de confiança de 5% é construído para avaliar os resultados encontrados para cada uma das bases de dados. As bases utilizadas são 4: Iris, Diabetes, *wine* e a base sintética.

A Figura 5.1 apresenta os melhores resultados encontrados para diferentes configurações de todos os algoritmos envolvidos nesse experimento para cada uma das quatro bases de dados utilizadas.

Nos testes realizados para a base de dados Iris, os algoritmos supervisionados conseguiram resultados acima de 90% na média das taxas de acerto. O melhor algoritmo supervisionado foi a rede neural MLP que obteve aproximadamente 97.178% e o pior foi o algoritmo Part que obteve 94.489%. O algoritmo proposto conseguiu se sobressair entre todos os algoritmos envolvidos, obtendo o melhor desempenho com uma média de taxa de acerto de 98.95% para esta base.

Na base de dados Diabetes, o algoritmo proposto alcançou o melhor resultado com uma boa margem de diferença em relação aos algoritmos supervisionados com o valor de 97.4% de média da taxa de acerto. O melhor algoritmo supervisionado foi mais uma vez a rede neural MLP com 76.81%, seguido de perto pelo algoritmo Bayes que obteve 75.75%. O algoritmo Part teve 73.45% de média de taxa de acerto.

Para a base de dados *Wine*, houve uma queda no rendimento do algoritmo proposto. O

algoritmo proposto obteve um resultado abaixo do esperado com uma média de taxa de acerto de 90.15%. O melhor desempenho entre os algoritmos supervisionados foi do MLP que obteve 98.14% seguido pelo algoritmo de Bayes com 97.49% e o Part com 92.17%.

Na base de dados Sintética o algoritmo proposto alcançou o desempenho de 95,94% de média na taxa de acerto. O desempenho foi o terceiro melhor, dentre os algoritmos deste experimento. Dentre os algoritmos totalmente supervisionados, o melhor algoritmo foi a rede neural MLP que conseguiu 98.8% seguido pelo algoritmo Part com 98.31% e o Bayes com 94.42%.

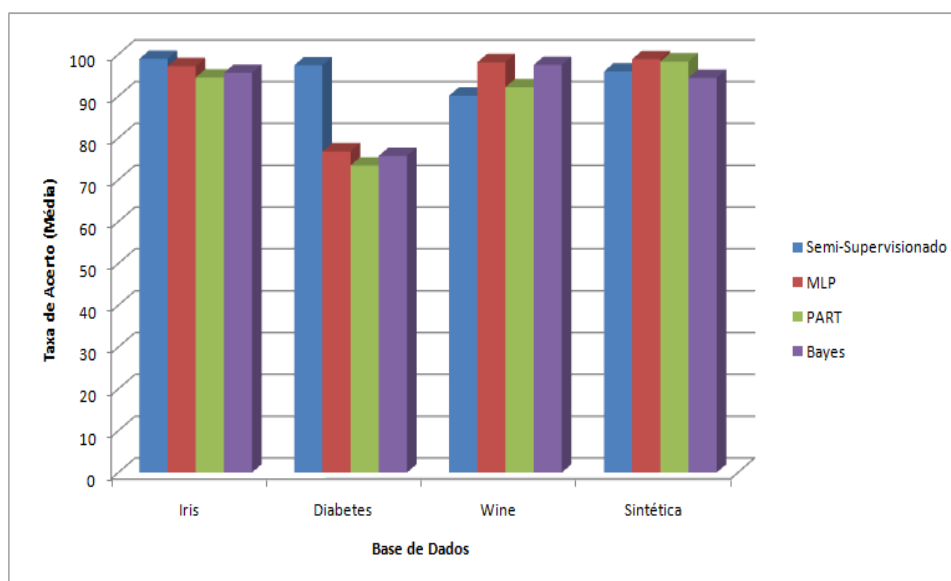


Figura 5.1 Estudo Comparativo da Classificação: Semi-Supervisionado x MLP x Bayes x Parts

Para checar os resultado encontrados, construímos intervalos de confiança com 5% de confiança para cada uma das bases avaliadas neste experimento. Os intervalos de confiança são apresentados na Figura 5.2.

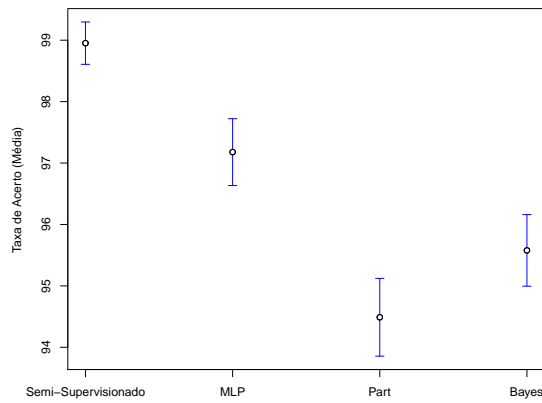
Para a base de dados Iris, o intervalo de confiança confirmou que o algoritmo proposto obteve o melhor resultado entre os algoritmos envolvidos. A rede neural MLP foi o melhor algoritmo dentre os totalmente supervisionados e entre os algoritmos Part e Bayes não se pode afirmar que um é melhor que o outro, já que os intervalos de confiança deles se interceptam.

Na base de dados diabetes, ficou claro que o algoritmo proposto alcançou o melhor resultado. O intervalo de confiança do algoritmo proposto ficou bem acima e isolado em relação aos demais. A rede neural MLP e o algoritmo Bayes são compatíveis, pois seus intervalos de confiança se interceptam. O algoritmo Part obteve o pior desempenho nesse experimento.

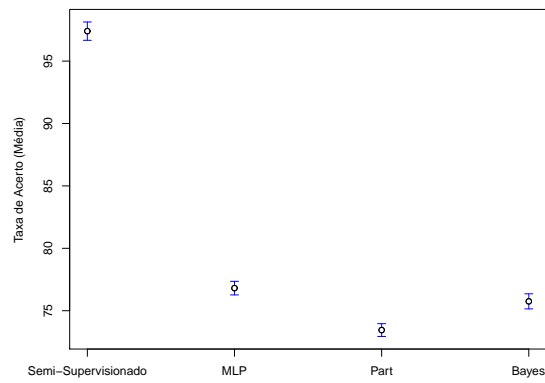
Para a base de dados Wine, o algoritmo proposto obteve um desempenho abaixo do esperado, porém conseguiu obter desempenho compatível com o algoritmo Part. A rede neural MLP obteve o melhor resultado dentre os algoritmos, seguido de perto pelo desempenho do algoritmo de Bayes e depois pelo algoritmo Part.

A base de dados Sintética, os algoritmos totalmente supervisionados MLP e Part foram os

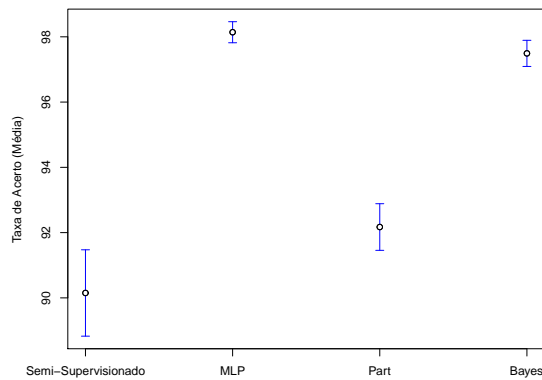
melhores algoritmos com seus desempenhos sendo equivalentes devido a intercessão de seus intervalos de confiança. O intervalo de confiança do algoritmo proposto possui intercessão com o intervalo do algoritmo Bayes, sendo equivalente a esse último algoritmo.



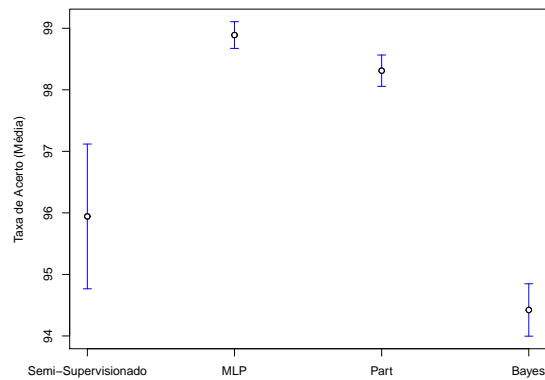
(a) Intervalo de Confiança para Base de Dados Iris



(b) Intervalo de Confiança para Base de Dados Diabetes



(c) Intervalo de Confiança para Base de Dados Wine



(d) Intervalo de Confiança para Base de Dados Sintética

Figura 5.2 Intervalos de Confiança Para o Comparativo do Algoritmo Semi-Supervisionado Proposto com Algoritmos Totalmente Supervisionados

5.1.1 Discussão

O objetivo principal deste experimento é comparar o desempenho de classificação do algoritmo de agrupamento semi-supervisionado proposto com outros algoritmos totalmente supervisionados na tarefa de classificação. O resultado deste experimento mostrou que o algoritmo proposto

é ainda melhor que algoritmos clássicos supervisionados para algumas das base de dados testadas ou obtém resultados compatíveis com algoritmos desse tipo.

O resultado obtido representa o bom desempenho que o algoritmo proposto possui de representar a distribuição dos dados quando este possui informações disponíveis suficientes. Neste caso, a base de dados no treinamento estava totalmente rotulada para todos os algoritmos envolvidos. Desse modo, o algoritmo de agrupamento semi-supervisionado proposto obtém bons resultados na classificação de padrões no ambiente supervisionado.

5.2 Resultados da Tarefa de Classificação Semi-Supervisionada

Nesta seção, os experimentos foram realizados com o objetivo de comparar o algoritmo proposto com algoritmos de agrupamento semi-supervisionado consolidados na literatura. Aqui, os testes são realizados utilizando validação cruzada por 10 vezes e repetida 30 vezes. A validação cruzada utilizada aqui foi explicada no capítulo anterior. Os resultados para esses experimentos são apresentados para 4 base de dados: Iris, Diabetes, *Wine* e a base de dados sintética. Os algoritmos avaliados foram o novo algoritmo de agrupamento semi-supervisionado proposto, o algoritmo de Pedrycz [PW97], o algoritmo de Bouchachia [BP06] e o algoritmo baseado em sementes [Bou07].

5.2.1 Resultados para a base de dados Iris

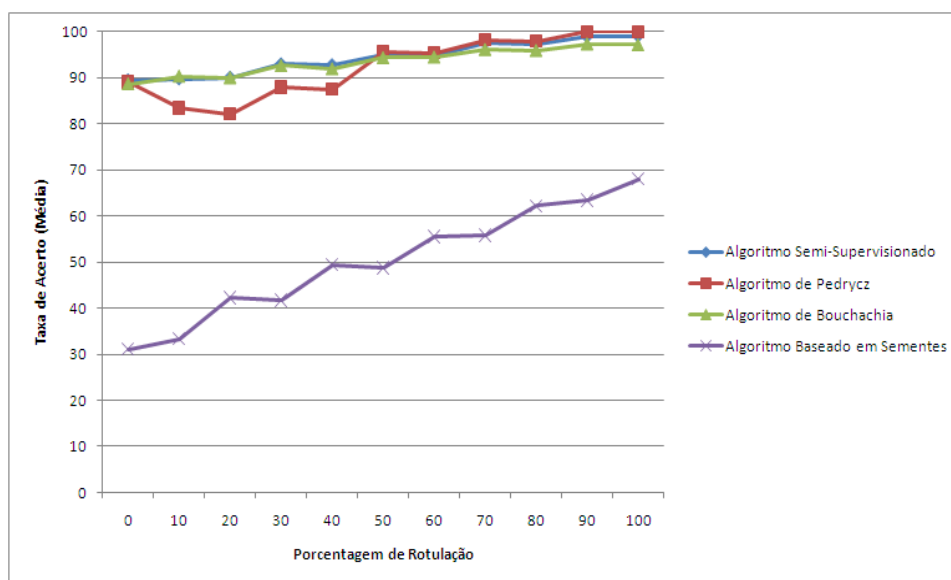
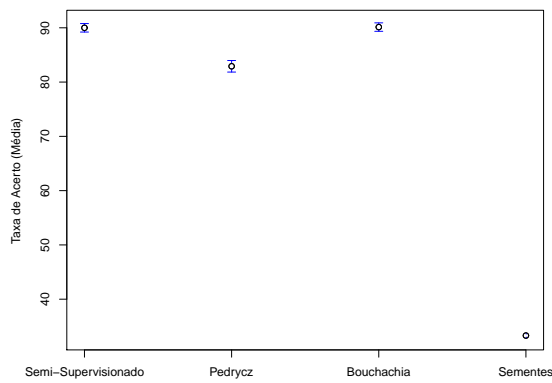


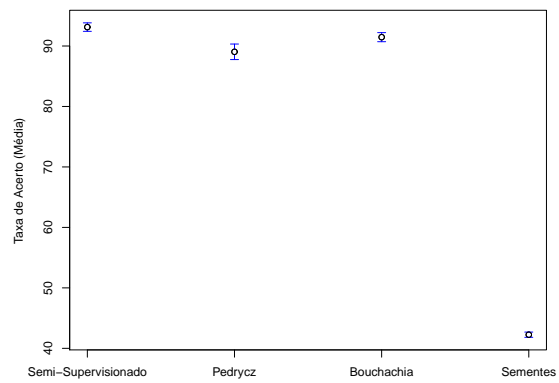
Figura 5.3 Validação Cruzada de Algoritmos De Agrupamento Semi-Supervisionado: Base de Dados Iris

Os experimentos do teste de validação para a base de dados Iris foram realizados com as configurações explicadas no capítulo anterior. Como pode-se observar, para poucos da-

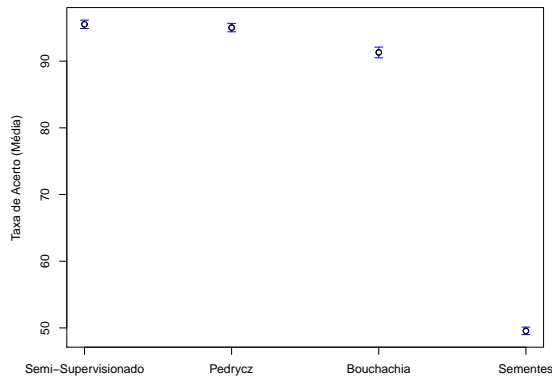
dos rotulados, os melhores algoritmos são o semi-supervisionado proposto e o algoritmo de Bouchachia que começam praticamente com o mesmo desempenho. O algoritmo de Pedrycz segue um pouco abaixo do desempenho dos dois e o baseado em sementes obteve o pior desempenho. Na medida que o experimento segue, com 50% de dados rotulados, o algoritmo de Pedrycz consegue o melhor desempenho até os 100% de dados rotulados. O algoritmo proposto, obteve o segundo melhor desempenho para essas configurações, seguido pelo algoritmo de Bouchachia. O algoritmo baseado em sementes segue melhorando o desempenho mas fica muito abaixo de todos os outros algoritmos.



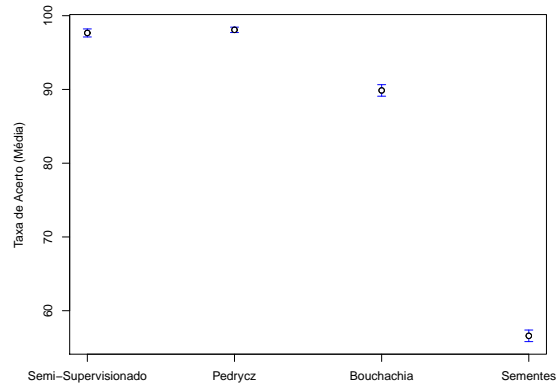
(a) Intervalo de Confiança para 10% de Rotulação



(b) Intervalo de Confiança para 30% de Rotulação



(c) Intervalo de Confiança para 50% de Rotulação



(d) Intervalo de Confiança para 70% de Rotulação

Figura 5.4 Intervalos de Confiança para a Base de Dados Iris Utilizando Validação Cruzada com Inicialização *Hard*

Para validar os experimentos, construímos intervalos com 5% de confiança para algumas configurações deste experimento: 10%, 30%, 50% e 70% de dados rotulados. Os intervalos são apresentados na Figura 5.4. Para poucos dados rotulados, podemos dizer que o algoritmo

proposto e o algoritmo de Bouchachia são as melhores escolhas, pois possuem melhores desempenhos. Para 30%, quando há um pouco mais de dados rotulados, nota-se que o algoritmo proposto possui a melhor configuração entre todos os algoritmos. Na medida que mais dados são rotulados, os melhores algoritmos são o proposto e o de Pedrycz, sendo que não se pode dizer que um é melhor que o outro de acordo com o intervalo de confiança. O algoritmo baseado em sementes obteve resultados muito ruins nesse experimento.

5.2.2 Resultados para a base de dados Diabetes

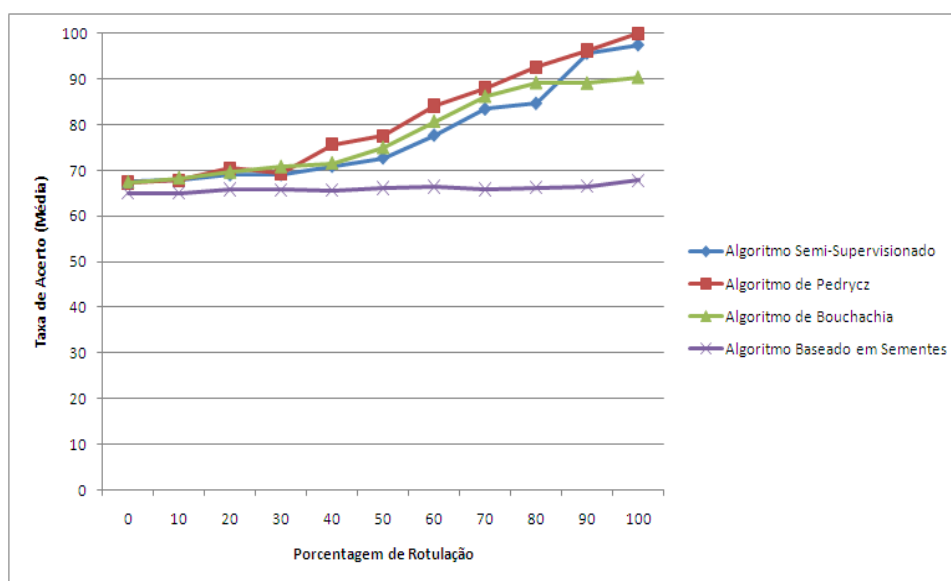
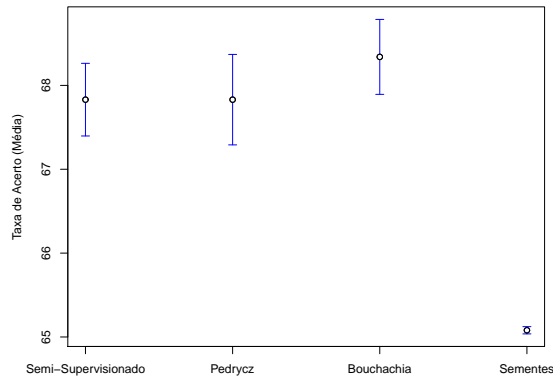


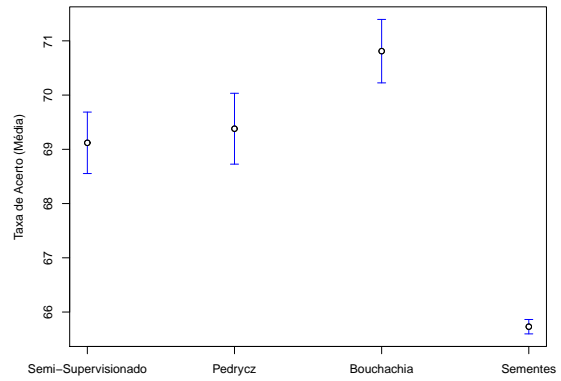
Figura 5.5 Validação Cruzada de Algoritmos De Agrupamento Semi-Supervisionado: Base de Dados Diabetes

O resultado para a base Diabetes é apresentado na Figura 5.5. Para poucos dados rotulados disponíveis no treinamento dos algoritmos semi-supervisionados, todos se comportaram de maneira similar, com pouca diferença no desempenho. A partir de 40% de dados rotulados, o algoritmo de Pedrycz conseguiu se sobressair, obtendo o melhor resultado. O desempenho entre 40% e 70% de dados rotulados, o algoritmo de Bouchachia alcançou o segundo melhor desempenho ficando a frente dos algoritmo proposto e do algoritmo baseado em sementes. A partir de 80% de dados rotulados o algoritmo proposto obteve o segundo melhor desempenho. O algoritmo baseado em sementes obteve o pior desempenho e não conseguiu melhorar o desempenho a medida que mais dados rotulados eram disponibilizados para o treinamento.

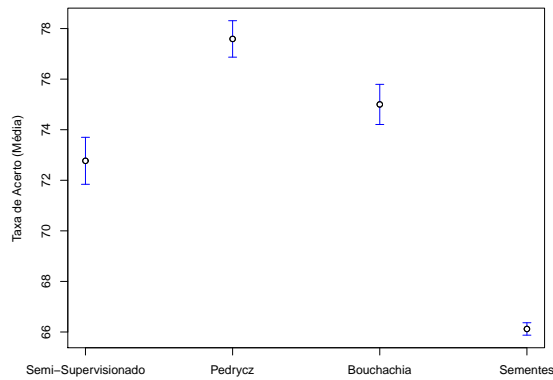
Intervalos de confiança foram construídos para avaliar algumas configurações do experimento. Construímos intervalos com 5% de confiança para 10%, 30%, 50% e 70% de dados rotulados. Como pode-se observar na Figura 5.6, para poucos dados rotulados, os algoritmos se equivalem, exceto o algoritmo baseado em sementes que foi o pior para todas as configurações. Com 30% de dados rotulados o algoritmo de Bouchachia foi o melhor e os algoritmos proposto e o de Pedrycz foram equivalentes. Para 50% e 70% o algoritmo de Pedrycz obteve



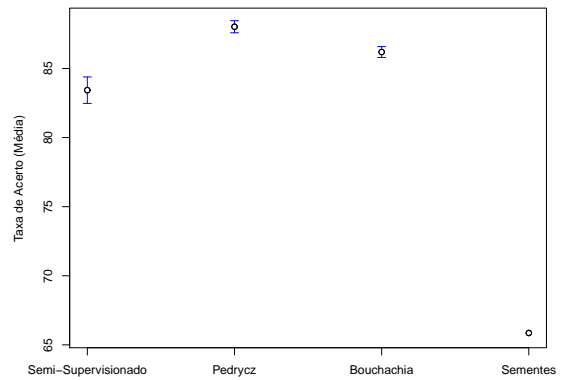
(a) Intervalo de Confiança para 10% de Rotulação



(b) Intervalo de Confiança para 30% de Rotulação



(c) Intervalo de Confiança para 50% de Rotulação



(d) Intervalo de Confiança para 70% de Rotulação

Figura 5.6 Intervalos de Confiança para a Base de Dados Diabetes Utilizando Validação Cruzada com Inicialização *Hard*

o melhor desempenho, o algoritmo de Bouchachia ficou logo abaixo e o algoritmo proposto obteve o terceiro melhor desempenho.

5.2.3 Resultados para a base de dados *Wine*

Os experimentos do teste de validação utilizando a base de dados *Wine* foram realizados com as configurações explicadas no capítulo anterior. Os resultados para este experimento são apresentados na Figura 5.7. Neste experimento, o algoritmo que se destacou foi o de Pedrycz que conseguiu melhor desempenho para as configurações acima de 40% de dados rotulados. O algoritmo proposto conseguiu obter o melhor desempenho com poucos dados rotulados, até 20% deles rotulados. Após essa configuração, o algoritmo proposto foi o segundo melhor em ter-

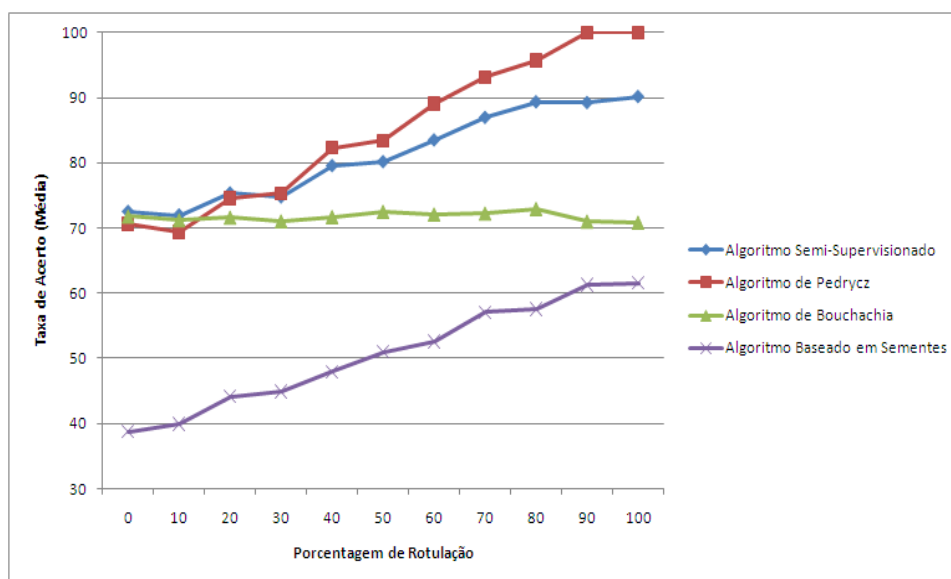


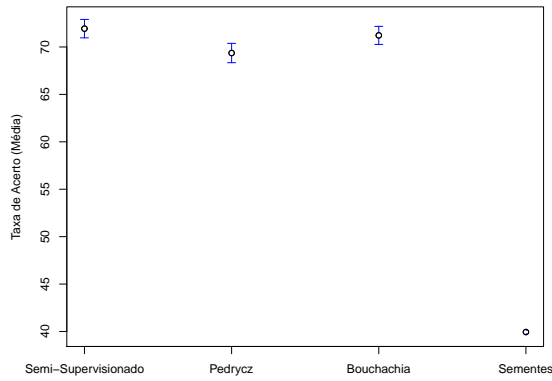
Figura 5.7 Validação Cruzada de Algoritmos De Agrupamento Semi-Supervisionado: Base de Dados Wine

mos de desempenho. O algoritmo de Bouchachia obteve bons desempenhos até 30% de dados rotulados, quando mais dados eram disponibilizados, esse algoritmo não conseguiu melhorar seu desempenho, havendo inclusive uma pequena queda com a maioria da base rotulada. O algoritmo baseado em sementes conseguiu melhorar o desempenho com o aumento de dados rotulados no treinamento, mas manteve-se com o pior desempenho.

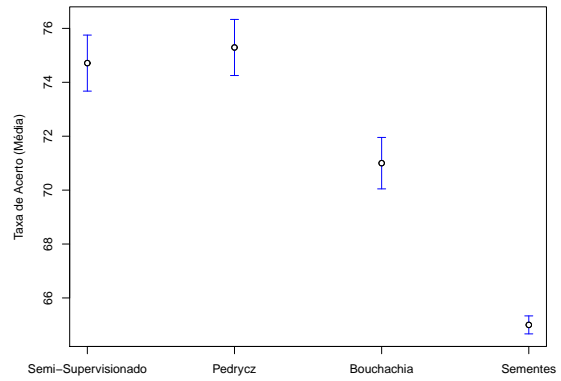
Foram construídos intervalos com 5% de confiança para as configurações de 10%, 30%, 50% e 70% de dados rotulados mostrados na Figura 5.8. O algoritmo de Pedrycz, confirmou ser o melhor algoritmo para essa base de dados. O algoritmo proposto foi a segunda melhor opção. Para 10% e 30% de dados rotulados o algoritmo proposto foi a melhor opção, sendo compatível com o algoritmo de Bouchachia e Pedrycz respectivamente. Com as outras duas configurações se manteve como o segundo melhor algoritmo. O algoritmo de Bouchachia obteve um bom resultado apenas para 10% dos dados rotulados, para 30%, 50% e 70% dos dados rotulados o algoritmo foi melhor apenas que o algoritmo baseado em sementes, que obteve o pior desempenho dentre os algoritmos estudados.

5.2.4 Resultados para a base de dados Sintética

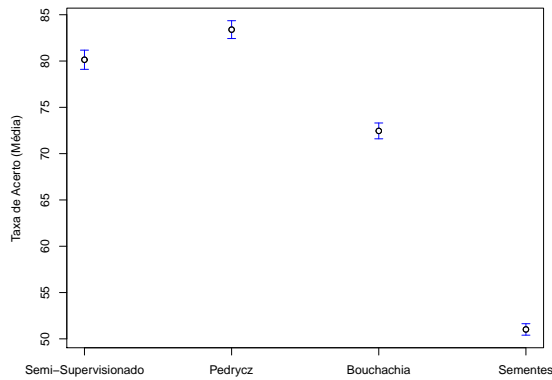
A base de dados sintética foi pensada para avaliar o poder de classificação do algoritmo. Os resultados são apresentados na Figura 5.9. Para poucos dados rotulados, 0% a 30% de dados rotulados, o algoritmo proposto consegue obter os melhores resultados dentre os algoritmos semi-supervisionados do experimento. O algoritmo de Bouchachia consegue ser a segunda melhor opção para essas configurações, e logo depois o algoritmo de Pedrycz obtém o terceiro melhor desempenho. Após 40% de dados rotulados, o algoritmo de Pedrycz teve o melhor desempenho. O algoritmo proposto manteve-se como o segundo melhor algoritmo e o algoritmo



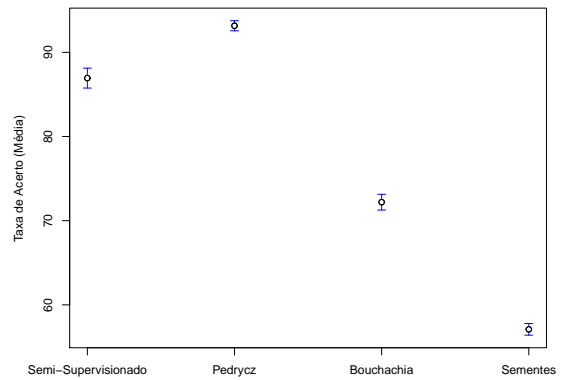
(a) Intervalo de Confiança para 10% de Rotulação



(b) Intervalo de Confiança para 30% de Rotulação



(c) Intervalo de Confiança para 50% de Rotulação



(d) Intervalo de Confiança para 70% de Rotulação

Figura 5.8 Intervalos de Confiança para a Base de Dados *Wine* Utilizando Validação Cruzada com Inicialização *Hard*

de Bouchachia o terceiro para as mesmas configurações. Todos os algoritmos conseguiram melhorar o desempenho com o aumento de dados rotulados disponíveis. O algoritmo baseado em sementes obteve o pior desempenho dentre os algoritmos.

Para validar os experimentos, foram construídos intervalos com 5% de confiança para 4 configurações do experimento mostrados na Figura 5.10. Para 10% de dados rotulados, o algoritmo proposto foi de fato o melhor algoritmo com poucos dados rotulados. Para 30% de dados rotulados, apesar do algoritmo proposto ter a melhor média de taxa de acerto, seu intervalo de confiança faz intercessão com o intervalo do algoritmo de Pedrycz, mas pode ser considerado melhor que os demais algoritmos. Para 50% de dados rotulados, o algoritmo de Pedrycz obteve o melhor resultado. Os algoritmos proposto e o de Bouchachia foram equivalentes nesse experimento. No experimento com 70% de dados rotulados, o algoritmo de Pedrycz manteve-se

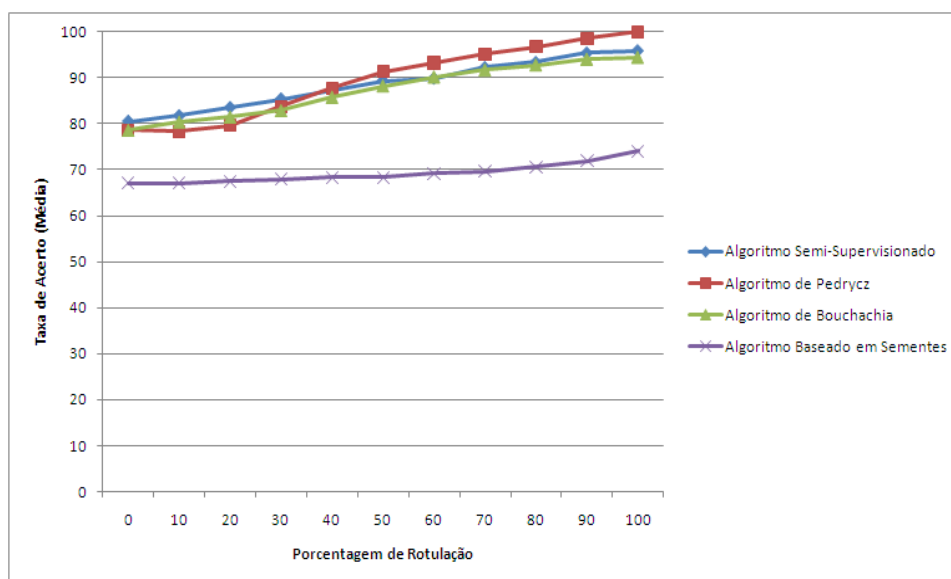


Figura 5.9 Validação Cruzada de Algoritmos De Agrupamento Semi-Supervisionado: Base de Dados Sintética

com o melhor desempenho. Os algoritmos proposto e o de Bouchachia se mantiveram equivalentes mais uma vez. O algoritmo baseado em sementes obteve o pior desempenho em todos os experimentos.

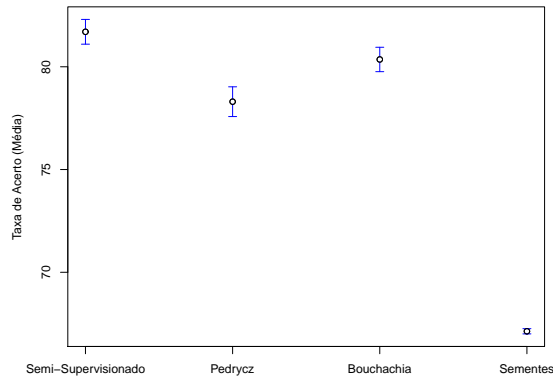
5.2.5 Discussão

Para esse experimento, podemos observar que o desempenho do algoritmo proposto é compatível com outros algoritmos de agrupamento semi-supervisionado consolidados na literatura. O algoritmo proposto alcançou os melhores resultados mesmo quando havia poucos, 0% a 30%, dados rotulados. Quando mais dados eram disponibilizados, o mesmo obteve bom desempenho, conseguindo estar sempre entre os melhores algoritmos avaliados.

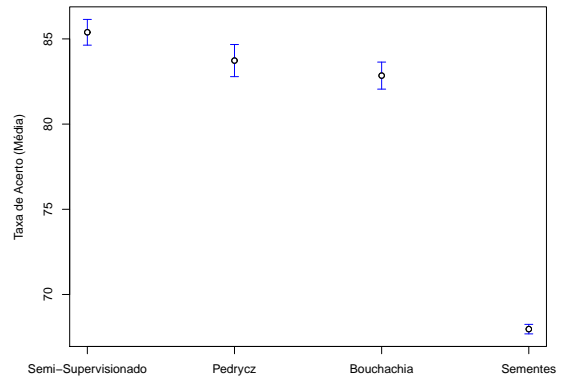
Nesse experimento também pode-se notar que o algoritmo proposto aumenta seu desempenho na medida que mais dados rotulados são disponibilizados para o treinamento. Para alguns algoritmos testados, algoritmo de Bouchachia e o baseado em sementes, em algumas configurações não conseguem melhorar, ou chegam a ter uma queda no desempenho para algumas bases de dados.

Desse modo, pode-se observar que o algoritmo proposto é uma ótima opção para casos onde existam poucos exemplos rotulados em relação ao total da base de dados. Na medida que mais dados forem rotulados, pode-se prever que o algoritmo também melhora o desempenho e ao lado do algoritmo de Pedrycz são as melhores opções para esses casos.

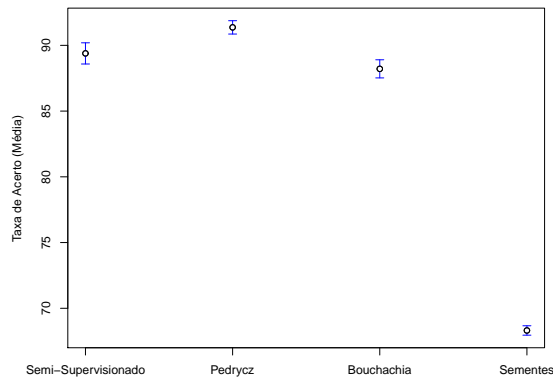
O resultado do treinamento dos algoritmos são partições geradas pelo algoritmo, cada partição é representada por um protótipo. Com a validação cruzada, podemos avaliar a eficácia desses protótipos, pois na etapa de teste, o conjunto de teste é atribuído a um grupo através da dissimilaridade do exemplo do conjunto de teste com esse protótipo. A similaridade é medida



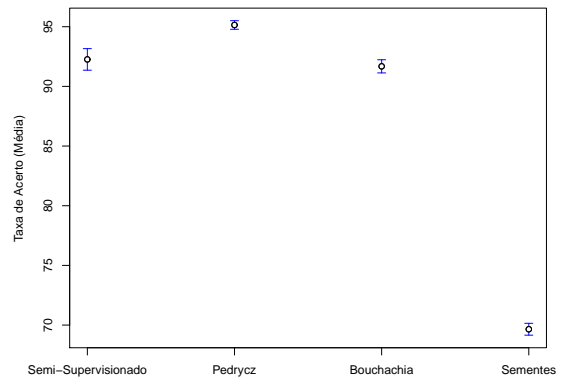
(a) Intervalo de Confiança para 10% de Rotulação



(b) Intervalo de Confiança para 30% de Rotulação



(c) Intervalo de Confiança para 50% de Rotulação



(d) Intervalo de Confiança para 70% de Rotulação

Figura 5.10 Intervalos de Confiança para a Base de Dados Sintética Utilizando Validação Cruzada com Inicialização *Hard*

pelo grau de pertinência desse exemplo nos grupos formados. Assim, podemos afirmar que os protótipos gerados pelo algoritmo proposto é eficaz, pois produziu bons resultados neste experimento.

Podemos afirmar que o algoritmo proposto pode ser utilizado em aplicações reais que seja custoso rotular dados e existam poucos dados disponíveis em relação ao total da base de dados. Após treinado, os protótipos gerados podem ser utilizados para rotular mais exemplos da base de dados. Assim, o algoritmo proposto mostrou ser uma ótima opção para aplicações de classificação de padrões.

5.3 Resultados da Tarefa de Agrupamento Semi-Supervisionado

Nesta seção, assim como na seção anterior, os experimentos foram realizados com o objetivo de comparar o algoritmo proposto com algoritmos de agrupamento consolidados na literatura. Aqui, os testes são realizados utilizando todos os padrões da base no treinamento e no teste dos algoritmos. O objetivo deste experimento é verificar o desempenho dos algoritmos semi-supervisionados na tarefa agrupamento com diferentes porcentagens de base de dados rotuladas para treinar o algoritmo. Os resultados para esses experimentos são apresentados para 5 bases de dados: Iris, Diabetes, *Wine*, Sintética e *Spam*. Os algoritmos avaliados foram o algoritmo de agrupamento semi-supervisionado proposto, o algoritmo de Pedrycz [PW97], o algoritmo de Bouchachia [BP06] e o algoritmo baseado em sementes [Bou07].

5.3.1 Resultados para a base de dados Iris

A base Iris, por ser uma base que possui algumas exemplos que estão claramente em outro grupo é apropriada para testar o desempenho do algoritmo para esse tipo de problema. Como possui uma distribuição fácil para algoritmos de aprendizagem de máquina, é obrigatório o bom desempenho nessa base. Os resultados para a base de dados Iris são apresentados na Figura 5.11.

Os resultados dos experimentos para a base de dados Iris demonstram que o algoritmo proposto possui melhor desempenho nos experimentos com poucos dados rotulados, até a configuração com 50% de dados rotulados. A partir de 60% de dados rotulados, o algoritmo proposto possui praticamente o mesmo desempenho em relação aos algoritmos de Pedrycz e de Bouchachia. O algoritmo de Bouchachia obteve o segundo melhor desempenho para as configurações abaixo de 50% de dados rotulados, sendo seguido pelo algoritmo de Pedrycz. O algoritmo baseado em sementes conseguiu resultados satisfatórios apenas no intervalo de 20% e 30% de dados rotulados, obtendo o terceiro melhor desempenho nessas configurações. No restante, não conseguiu obter bons resultados, não melhorando seu desempenho com rótulos disponíveis acima de 50%.

O resultado para o índice de rand corrigido global para os experimentos realizados na base Iris são apresentados na Figura 5.12. Observando a figura, o algoritmo proposto consegue melhores resultados que todos os outros algoritmos comparados na maioria das configurações do experimento, principalmente de 10% a 60% de dados rotulados. Há uma melhora no desempenho dos algoritmos proposto, de Pedrycz e de Bouchachia na medida que mais informações são repassadas para o algoritmo. O algoritmo de Bouchachia obteve o segundo melhor desempenho, sendo seguido de perto pelo algoritmo de Pedrycz. Este último consegue uma melhora constante de acordo com o aumento de informações repassadas ao algoritmo, exceto na queda quando há 10% dos dados rotulados, se recuperando logo em seguida, com 20% dos dados rotulados. Assim como o algoritmo proposto, ele atinge o valor máximo quando há 100% dos dados rotulados. O algoritmo baseado em sementes possui o pior desempenho entre os algoritmos testados. Ele só consegue um desempenho melhor que o algoritmo de Pedrycz para 20% e 30% de dados rotulados, porém a melhora do desempenho é muito pequena na medida que mais dados rotulados são disponibilizados para seu treinamento.

O gráfico dos resultados do índice de rand corrigido para o conjunto de dados rotulados são

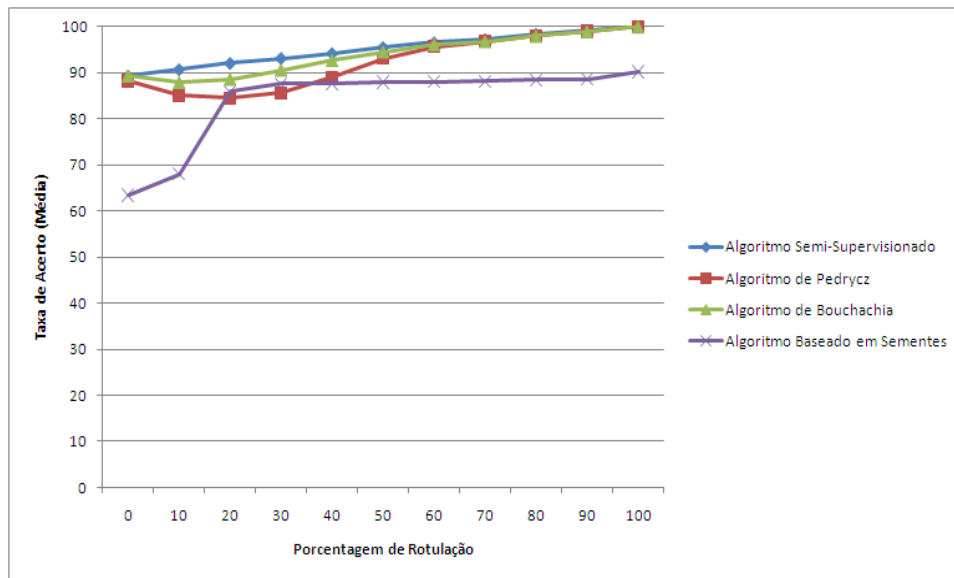


Figura 5.11 Estudo Comparativo: Base de Dados Iris

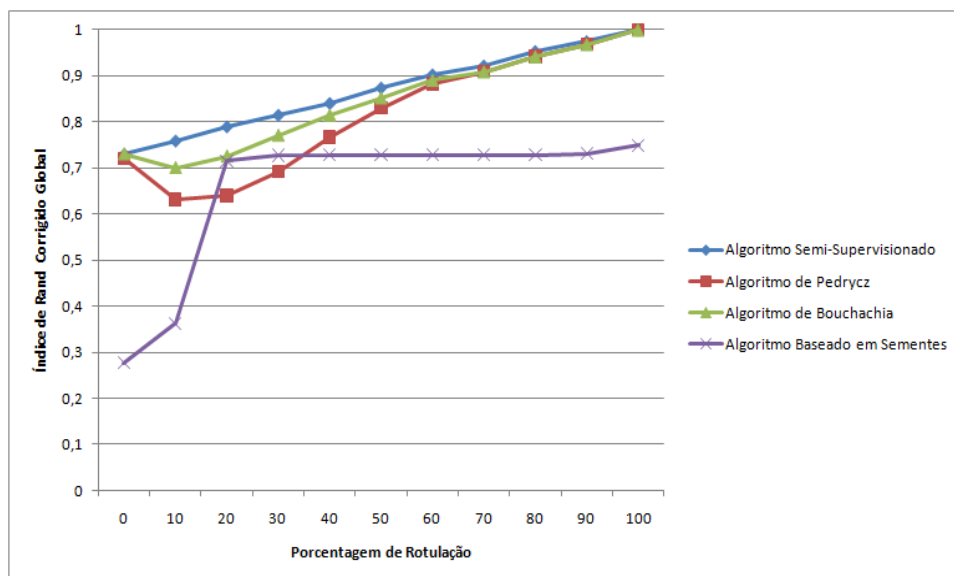


Figura 5.12 Estudo Comparativo do Índice de Rand Corrigido Global: Base de Dados Iris

apresentados na Figura 5.13. Os algoritmos proposto, de Bouchachia e de Pedrycz apresentam o valor máximo de índice de rand corrigido para todas as configurações do experimento. O algoritmo baseado em sementes obteve desempenho na casa de 0.7, aumentando um pouco na medida que mais dados rotulados são disponibilizados.

O resultado do índice de rand corrigido para o conjunto de dados não rotulados são apresentados na Figura 5.14. Os algoritmos proposto, de Pedrycz e de Bouchachia possuem um

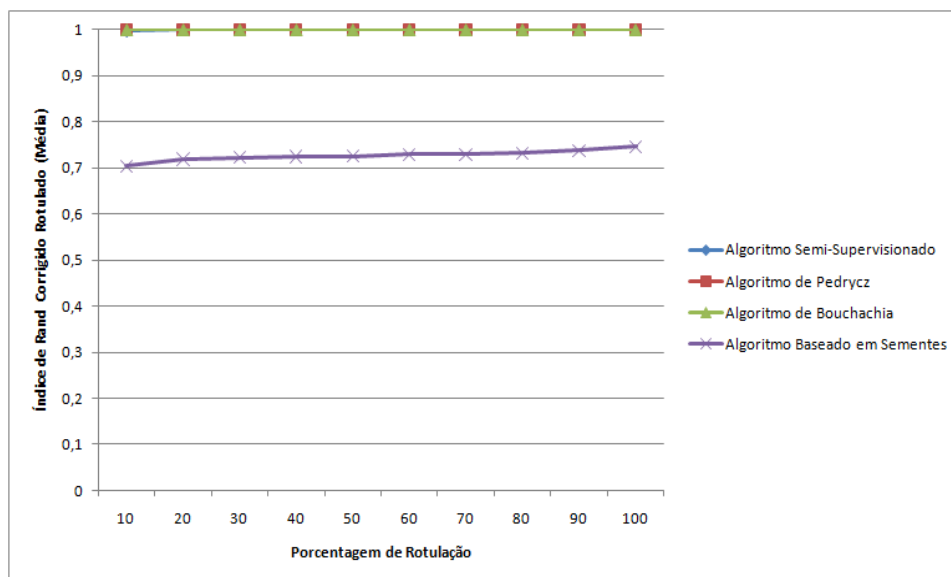


Figura 5.13 Estudo Comparativo do Índice de Rand Corrigido Rotulado: Base de Dados Iris

desempenho semelhante no intervalo de 0% a 30% de dados rotulados. Neste intervalo, o algoritmo baseado em sementes possui o pior desempenho. Nas configurações do experimento a partir de 40% de dados rotulados, o algoritmo proposto possui um desempenho melhor que os demais. Os outros 3 algoritmos possuem desempenho praticamente empatados nessas configurações.

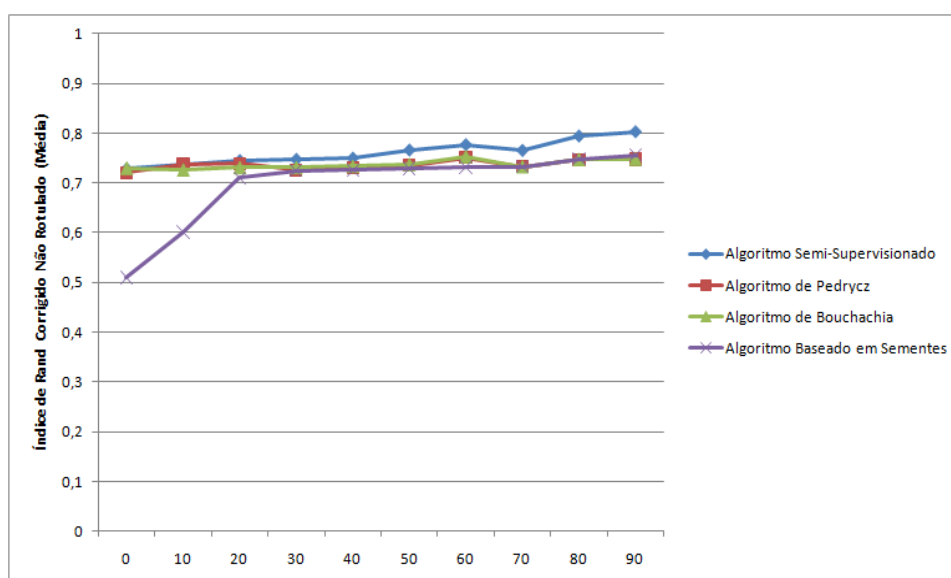
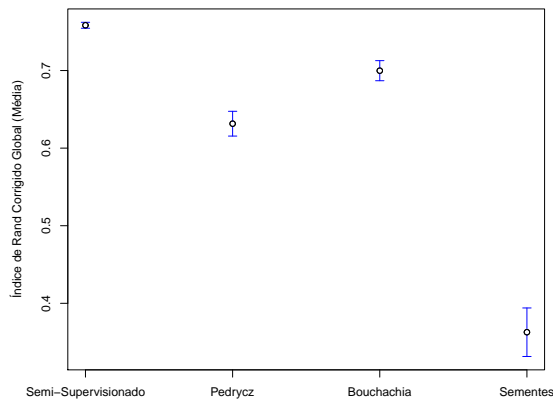
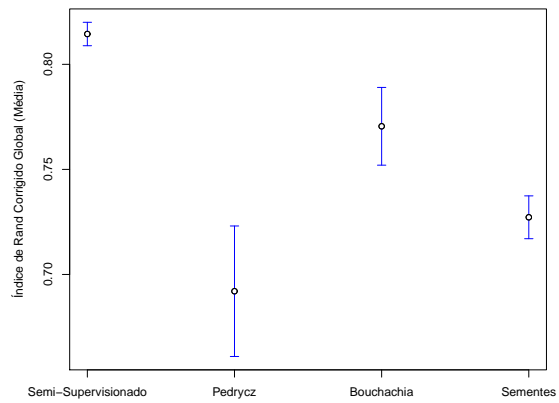


Figura 5.14 Estudo Comparativo do Índice de Rand Corrigido Não Rotulado: Base de Dados Iris

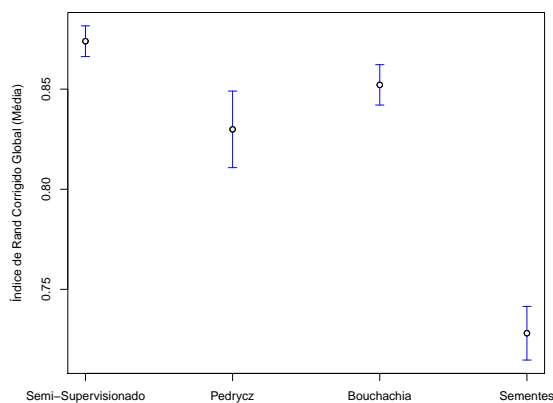
Intervalos com 5% de confiança foram construídos para o índice de rand corrigido global nas configurações de 10%, 30%, 50% e 70% de dados rotulados. Os intervalos de confiança para a base de dados Iris são apresentados na Figura 5.15. Os intervalos de confiança demonstram o bom desempenho do algoritmo proposto quando há poucos dados rotulados. Em todas as configurações mostradas o algoritmo proposto obteve o melhor desempenho. Apenas na configuração com 70% de dados rotulados há uma equivalência com o desempenho do algoritmo de Pedrycz. O algoritmo de Bouchachia obteve o segundo melhor desempenho para 10% e 30% de dados rotulados, sendo seguido pelo desempenho do algoritmo de Pedrycz e do algoritmo baseado em sementes, nessa ordem. Para 50% e 70% de dados rotulados, os algoritmos de Bouchachia e de Pedrycz obtêm resultados equivalentes. O algoritmo baseado em sementes teve o pior desempenho entre os algoritmos estudados.



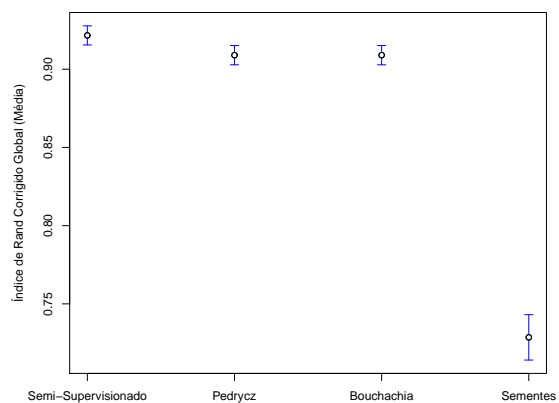
(a) Intervalo de Confiança para 10% de Rotulação



(b) Intervalo de Confiança para 30% de Rotulação



(c) Intervalo de Confiança para 50% de Rotulação



(d) Intervalo de Confiança para 70% de Rotulação

Figura 5.15 Intervalos de Confiança para a Base de Dados Iris na Tarefa de Agrupamento

5.3.2 Resultados para a base de dados Diabetes

O resultado para a base diabetes apresentado na Figura 5.16 demonstra um equilíbrio entre todos os algoritmos quando há poucos dados rotulados. Quando a porcentagem de rótulos é mais baixa, entre 10% e 30%, O algoritmo proposto obteve os melhores resultados. O desempenho dos algoritmos de Pedrycz e Bouchachia cai um pouco para 10% e 20% de dados rotulados. O algoritmo baseado em sementes obteve o segundo melhor desempenho para essas configurações. De 40% a 100% de dados rotulados, os algoritmos de Pedrycz e Bouchachia possuem praticamente o mesmo desempenho, alcançando a taxa de acerto máxima quando todos os rótulos estão disponíveis no treinamento. O algoritmo proposto consegue o terceiro melhor desempenho, caindo um pouco quando há 80% de dados rotulados. O algoritmo baseado em sementes não conseguiu melhorar seu desempenho na medida que mais dados são disponibilizados no treinamento, obtendo o pior desempenho.

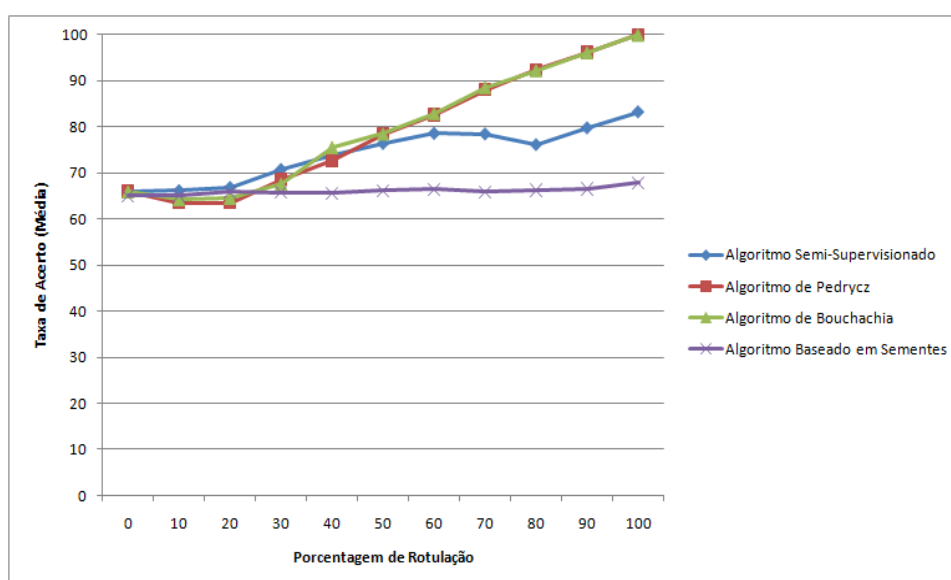


Figura 5.16 Estudo Comparativo: Base de Dados Diabetes

Para o índice de rand corrigido global, os resultados são mostrados na Figura 5.17. De 0% a 30% de dados rotulados, os algoritmos possuem um desempenho equivalente, exceto o algoritmo baseado em sementes que obteve o pior desempenho em todas as configurações, conseguindo aumentar um pouco seu desempenho quando mais de 50% dos dados rotulados estavam disponibilizados. O algoritmo de Bouchachia obtém o melhor desempenho para 40% de dados rotulados. Quando há mais de 50% de dados rotulados, os algoritmos de Pedrycz e Bouchachia possuem desempenhos equivalentes, aumentando significativamente a acurácia até todos os dados rotulados. O algoritmo proposto não consegue obter o mesmo desempenho para essas configurações, tendo o terceiro melhor desempenho com uma queda quando há 80% de dados rotulados.

O resultado do índice de rand corrigido dos dados rotulados são apresentados na Figura 5.18. Os algoritmos de Pedrycz e Bouchachia conseguem o valor máximo do índice para todas as

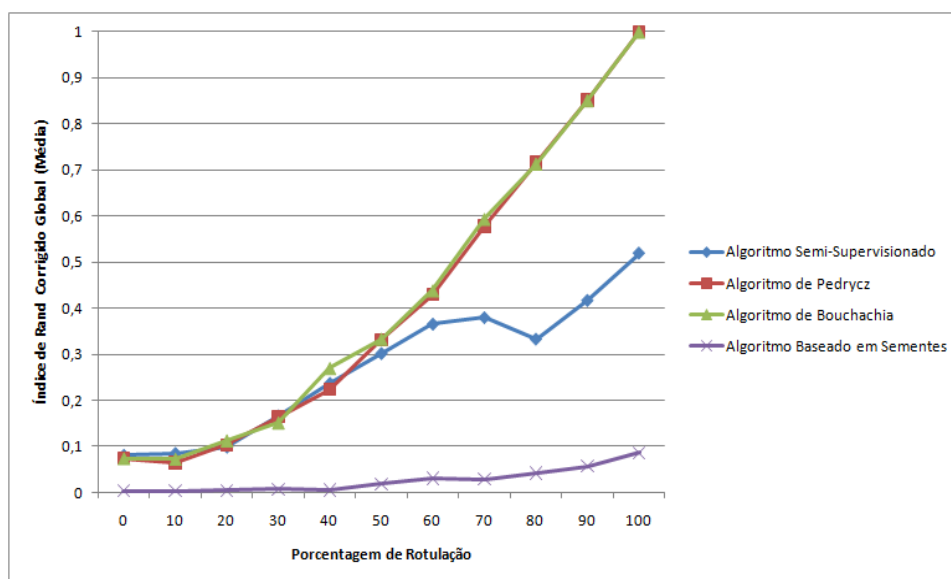


Figura 5.17 Estudo Comparativo do Índice de Rand Corrigido Global: Base de Dados Diabetes

configurações. O algoritmo proposto consegue melhorar seu desempenho até 40% de dados rotulados. Com mais de 50% de dados rotulados há uma queda no desempenho até 80% de dados rotulados. Para 90% e 100% de dados rotulados há um aumento no desempenho mas não alcança o valor máximo em nenhuma configuração. O algoritmo baseado em sementes consegue melhorar o desempenho na medida que mais dados rotulados estão disponibilizados mas não consegue obter bons resultados.

Para os dados não rotulados, os valores de índice de rand corrigido são mostrados na Figura 5.19. De 0% a 20% de dados rotulados todos os algoritmos possuem desempenho equivalentes. De 30% a 100% de dados rotulados, o algoritmo proposto obtém o pior resultados entre os algoritmos. Os algoritmos de Pedrycz e Bouchachia possuem desempenhos praticamente iguais para essas configurações. O algoritmo baseado em sementes consegue o melhor desempenho entre todos os algoritmos. Esse resultado demonstra o porque do aumento de desempenho do algoritmo baseado em sementes, porém, houve muitos erros na base rotulada de dados, o deixando na última colocação em termos de desempenho na configuração global do índice de rand.

O resultado dos intervalos com 5% confiança para o índice de rand global na base diabetes comprova o desempenho equivalente entre os algoritmos proposto, Pedrycz e Bouchachia. O algoritmo proposto obtém o melhor resultado juntamente com o algoritmo de Bouchachia quando há 10% de dados rotulados. Nessa mesma configuração, o algoritmo de Pedrycz obtém um resultado equivalente ao algoritmo de Bouchachia, mas ficou abaixo do algoritmo proposto. Para 30% e 50% de dados rotulados, os algoritmos obtém resultados equivalentes, exceto o algoritmo baseado em sementes que obteve o pior desempenho. Quando existe 70% de dados rotulados os algoritmos de Pedrycz e Bouchachia obtém o melhor resultado. O algoritmo proposto obteve o terceiro melhor desempenho, sendo seguido pelo desempenho do algoritmo

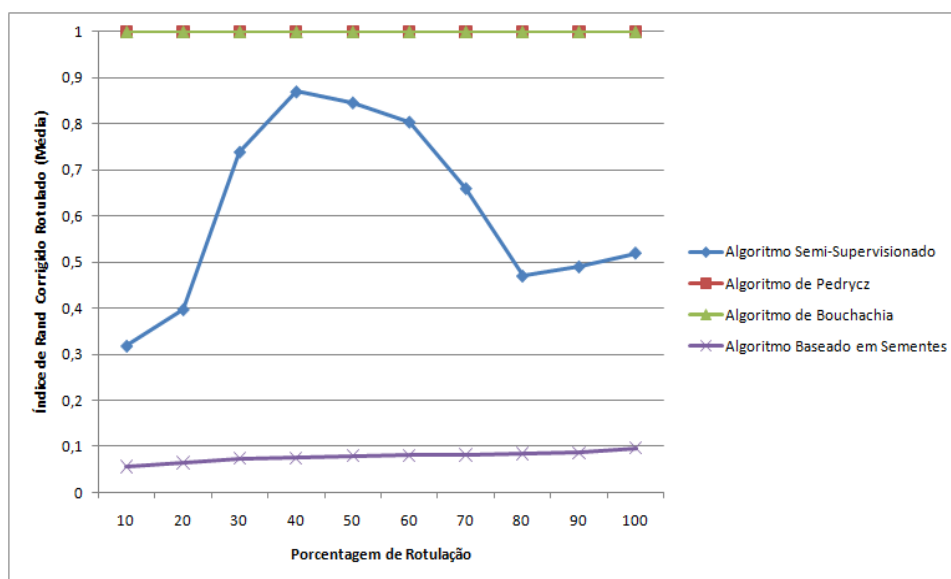


Figura 5.18 Estudo Comparativo do Índice de Rand Corrigido Rotulado: Base de Dados Diabetes

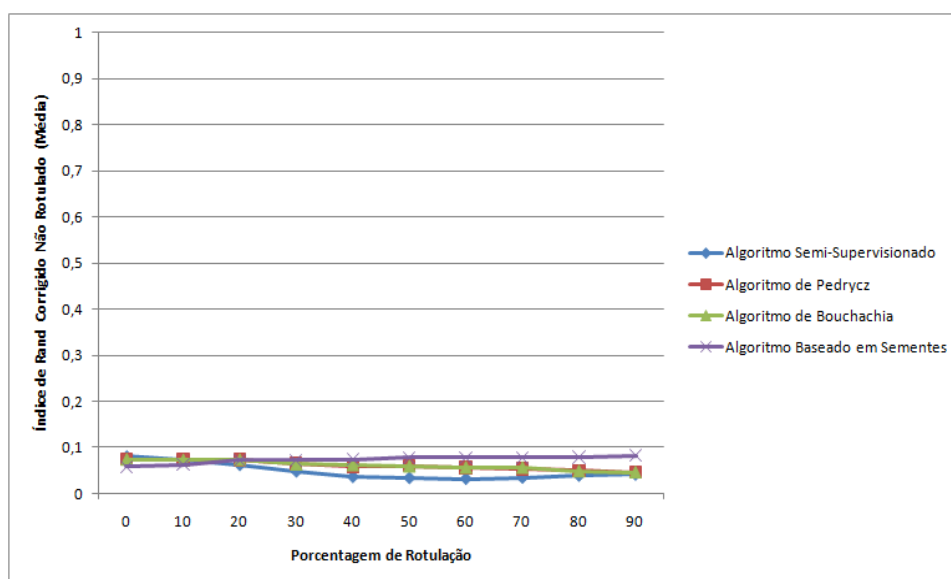
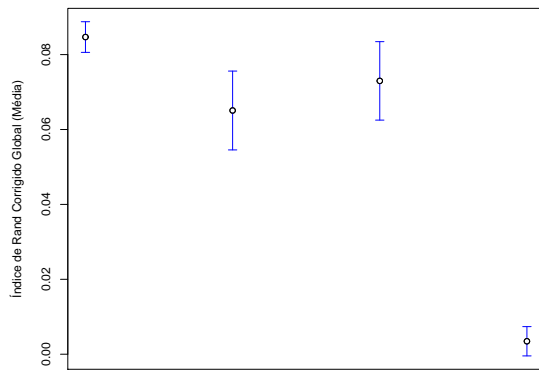


Figura 5.19 Estudo Comparativo do Índice de Rand Corrigido Não Rotulado: Base de Dados Diabetes

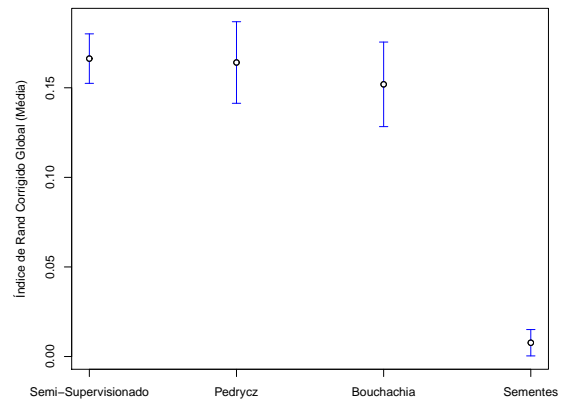
baseado em sementes.

5.3.3 Resultados para a base de dados Wine

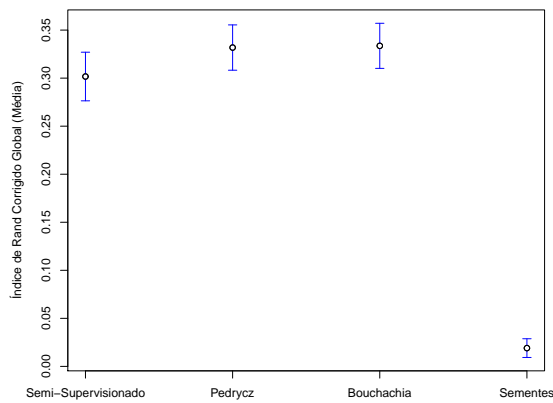
O resultado para a base de dados *Wine* mostrado na Figura 5.21 apresenta o algoritmo proposto obtendo a melhor precisão para as configurações com poucos dados rotulados, de 10% a 70% de



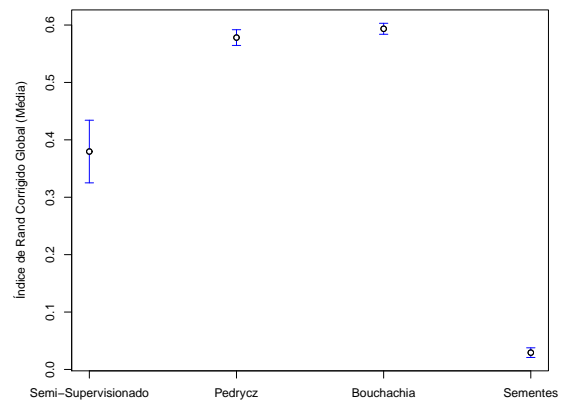
(a) Intervalo de Confiança para 10% de Rotulação



(b) Intervalo de Confiança para 30% de Rotulação



(c) Intervalo de Confiança para 50% de Rotulação



(d) Intervalo de Confiança para 70% de Rotulação

Figura 5.20 Intervalos de Confiança para a Base de Dados Diabetes na Tarefa de Agrupamento

dados rotulados. O algoritmo de Bouchachia, com 20% de dados rotulados consegue o melhor desempenho, tendo uma queda no desempenho em 30% de dados rotulados e melhorando seu desempenho logo em seguida até obter 100% de acerto quando todos os dados rotulados estão disponíveis para seu treinamento. O algoritmo de Pedrycz também cai com 30% de rótulos disponíveis e mantém uma taxa de acerto próxima ao algoritmo de Bouchachia no restante do experimento até obter 100% de acerto quando todos os dados rotulados estão disponíveis. O algoritmo baseado em sementes mais uma vez ficou abaixo dos outros algoritmos. O melhor resultado obtido por ele, é no intervalo de 20% e 30% de dados rotulados, não conseguindo um bom desempenho no restante das configurações do experimento.

Para o índice de rand corrigido global, o gráfico apresentado na Figura 5.22 confirma que o algoritmo proposto obteve melhores agrupamentos para as configurações com poucos dados

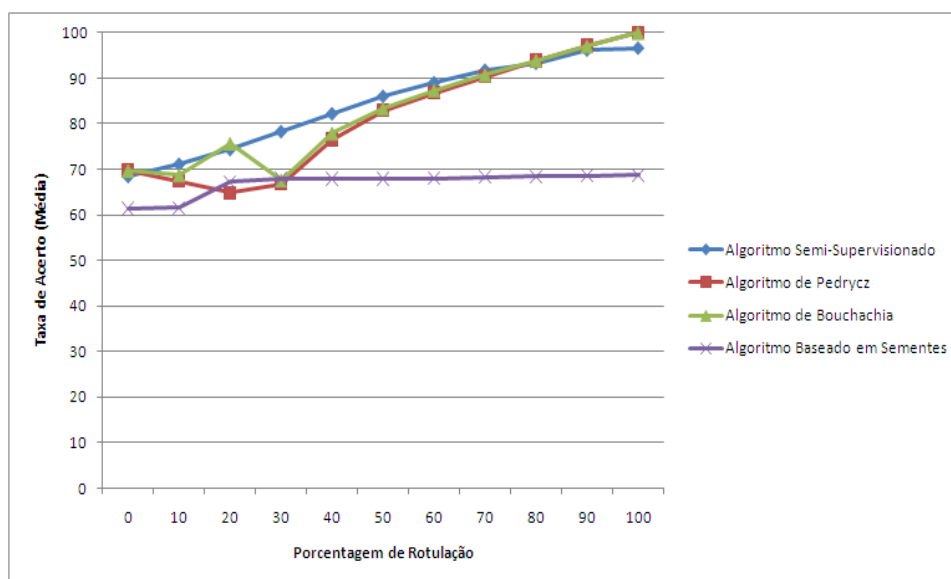


Figura 5.21 Estudo Comparativo: Base de Dados Wine

rotulados, até 70% de dados rotulados. Porém, mesmo não obtendo o melhor resultado nas configurações com mais dados rotulados, o algoritmo proposto mantém um aumento em seu desempenho. O algoritmo de Bouchachia obteve o melhor resultado na configuração de 20% de dados rotulados, tendo uma queda de desempenho em 30% de dados rotulados e se recuperando logo em seguida. Seu desempenho é melhorado na medida que mais dados rotulados são disponibilizados para seu treinamento. O algoritmo de Pedrycz teve um grande aumento em seu desempenho com mais de 30% de dados rotulados disponíveis. Os algoritmos de Pedrycz e de Bouchachia se mantêm um desempenho semelhante após essa configuração, até que os dois conseguem obter o valor máximo do índice quando todos os dados rotulados estão disponíveis para treinamento. O algoritmo baseado em sementes, apesar de um bom desempenho para as configurações de 20% e 30% de dados rotulados, não conseguiu acompanhar os resultados dos outros algoritmos, ficando abaixo deles em praticamente todo o experimento.

A Figura 5.23 apresenta o resultado para o conjunto de dados rotulados. Os algoritmos de Bouchachia e de Pedrycz conseguem o valor máximo do índice para todas as configurações. O algoritmo proposto parte do índice 0.8 quando não há dados rotulados até 1 quando há 50% de dados rotulados. Após 80% de dados rotulados o desempenho cai até quando todos os dados estão rotulados. O algoritmo baseado em sementes não obteve bons resultados, aumenta do índice na casa de aproximadamente 0.3 até aproximadamente 0.4.

O resultado do índice de rand corrigido para o conjunto de dados não rotulados é apresentado na Figura 5.24. Os 4 algoritmos apresentam desempenho semelhantes quando há poucos dados rotulados, até 30% de dados rotulados. Após 50% de dados rotulados o algoritmo proposto apresenta melhores resultados. Os algoritmos de Bouchachia e de Pedrycz apresentam desempenhos parecidos também após 50% de dados rotulados e o algoritmo baseado em sementes apresenta um desempenho um pouco menor que os demais algoritmos.

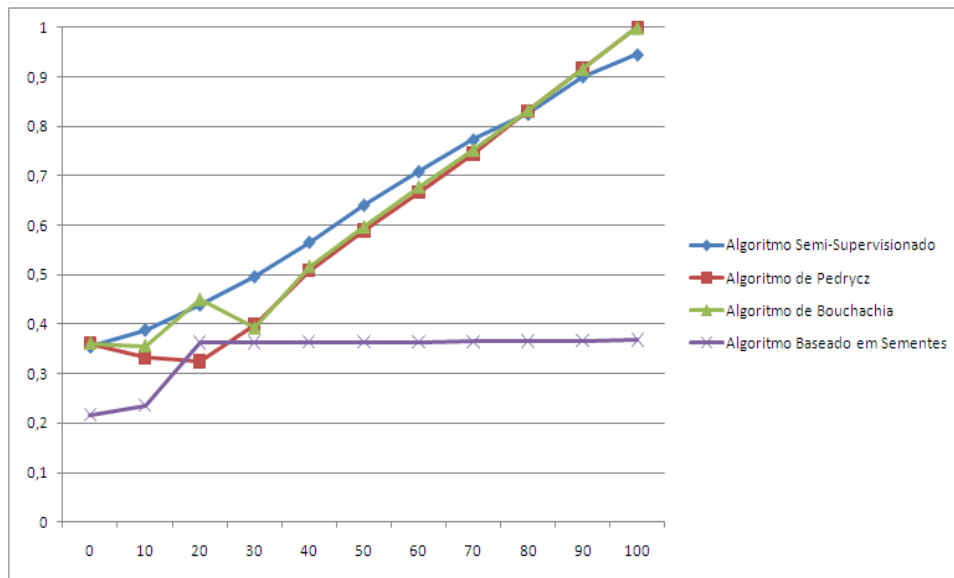


Figura 5.22 Estudo Comparativo do Índice de Rand Corrigido Global: Base de Dados Wine

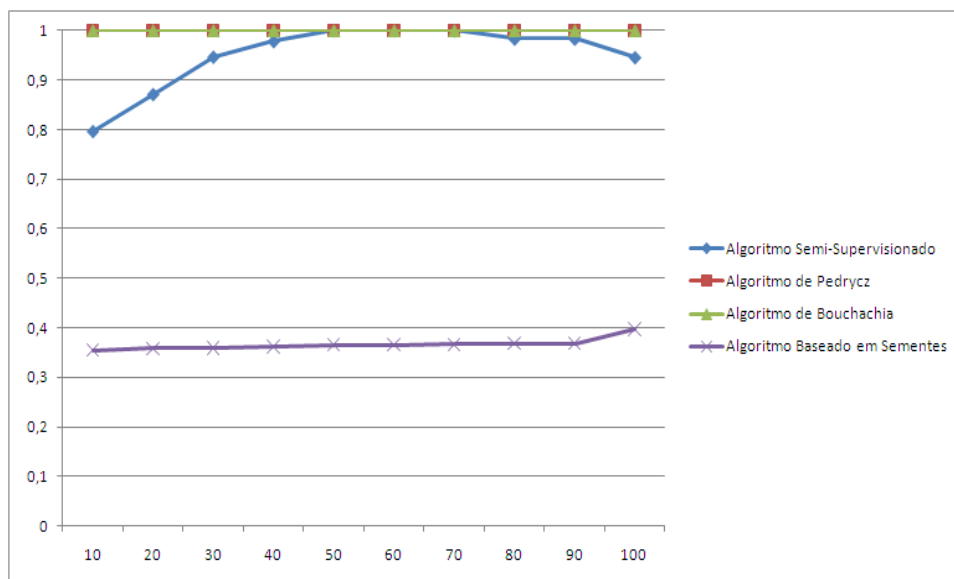


Figura 5.23 Estudo Comparativo do Índice de Rand Corrigido Rotulado: Base de Dados Wine

Foram construídos intervalos com 5% de confiança para o índice de rand corrigido global nas configurações 10%, 30%, 50% e 70% de dados rotulados. Os resultados são mostrados na Figura 5.25. Os resultados comprovam o bom desempenho do algoritmo proposto nessa base de dados. Em todos os intervalos de confiança, o algoritmo proposto alcançou o melhor desempenho. O algoritmo de Bouchachia teve o segundo melhor desempenho para 10% de dados rotulados. Nas demais configurações obteve desempenho equivalente ao algoritmo de

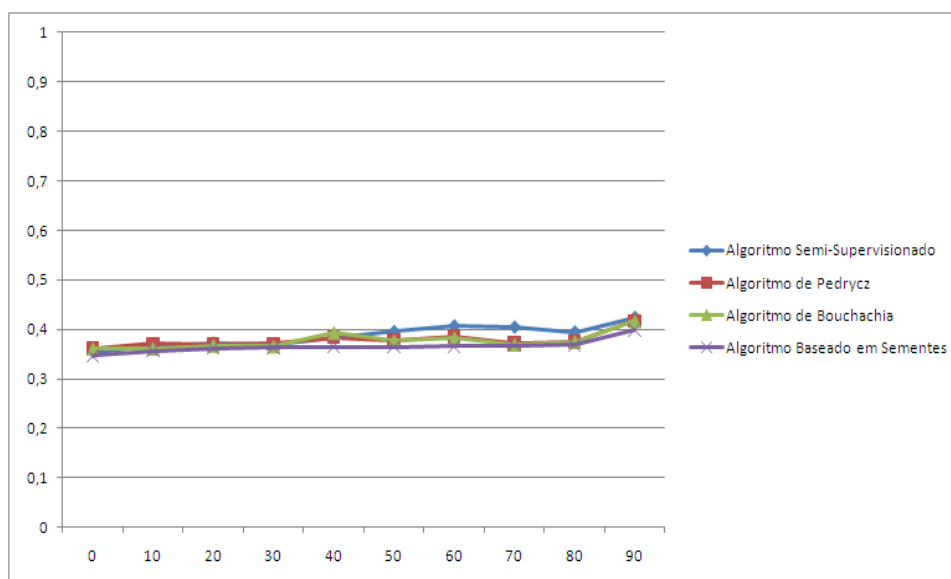


Figura 5.24 Estudo Comparativo do Índice de Rand Corrigido Não Rotulado: Base de Dados Wine

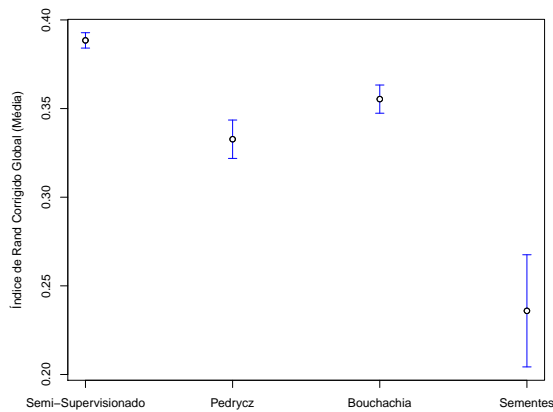
Pedrycz. O algoritmo baseado em sementes teve o pior desempenho comparado com os outros algoritmos.

5.3.4 Resultados para a base de dados Sintética

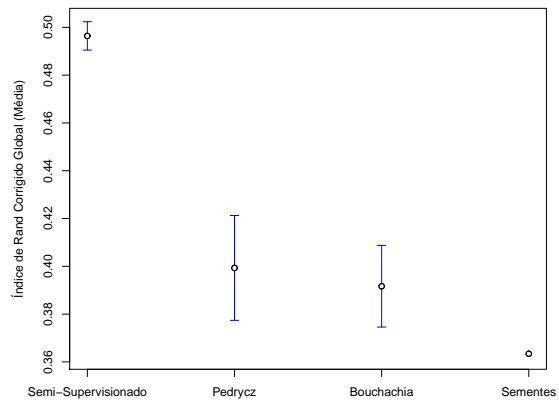
O resultado para a base de dados Sintética apresentado na Figura 5.26 apresenta que o algoritmo proposto obteve resultados compatíveis com o algoritmo de Pedrycz. O algoritmo de Bouchachia foi o terceiro melhor algoritmo, melhorando seu desempenho na medida que mais dados rotulados eram disponibilizados para seu treinamento. O algoritmo baseado em sementes mais uma vez ficou abaixo dos outros algoritmos, mesmo melhorando seu desempenho com mais dados rotulados disponíveis.

Para o índice de rand corrigido global, o gráfico apresentado na Figura 5.27 apresenta que o algoritmo proposto obteve melhores agrupamentos para a maioria das configurações do experimento, se destacando ao algoritmo de Pedrycz que conseguiu taxas semelhantes no gráfico anterior. O algoritmo de Pedrycz obteve o segundo melhor desempenho, sendo o melhor nas configurações de 30% e 40% de dados rotulados. Estes dois primeiros algoritmos conseguem o índice máximo quando todos os dados do treinamento são rotulados. O algoritmo de Bouchachia obteve o terceiro melhor desempenho e o baseado em sementes obteve o pior desempenho dentre os algoritmos estudados.

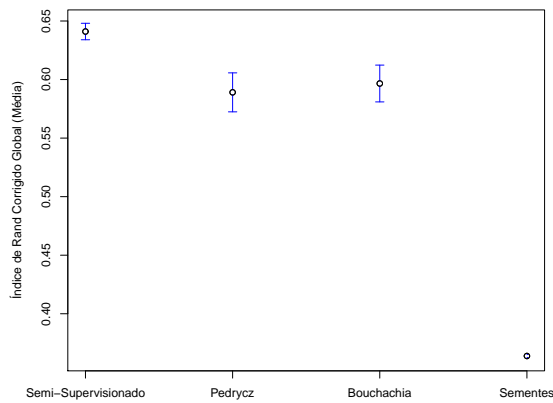
A Figura 5.28 apresenta o resultado do índice da rand corrigido para o conjunto de dados rotulados. O algoritmo de Pedrycz consegue o valor máximo do índice para todas as configurações. O algoritmo proposto parte de um índice acima de 0,9 e alcança o valor máximo em 60% de dados rotulados e permanece com o valor máximo até a configuração de 100% de dados rotulados. O algoritmo de Bouchachia consegue o terceiro melhor desempenho com o



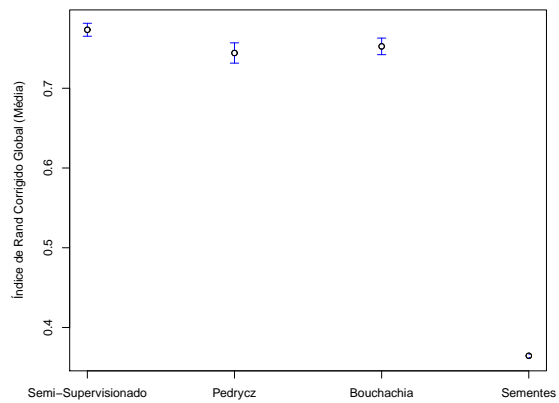
(a) Intervalo de Confiança para 10% de Rotulação



(b) Intervalo de Confiança para 30% de Rotulação



(c) Intervalo de Confiança para 50% de Rotulação



(d) Intervalo de Confiança para 70% de Rotulação

Figura 5.25 Intervalos de Confiança para a Base de Dados *Wine* na Tarefa de Agrupamento

índice partindo de um pouco mais de 0.4 e chegando próximo a 0.5 nas configurações finais. O algoritmo baseado em sementes obteve o pior desempenho dentre os algoritmos estudados.

O resultado do índice de rand corrigido para o conjunto de dados não rotulados é apresentado na Figura 5.29. O algoritmo de Pedrycz consegue o melhor desempenho nas configurações com menos dados rotulados até 60% de dados rotulados, conseguindo o segundo melhor desempenho nas configurações com mais dados rotulados. O algoritmo proposto obteve o segundo melhor desempenho para poucos dados rotulados. Nas configurações acima de 70% de dados rotulados, o algoritmo proposto conseguiu o melhor desempenho. O algoritmo de Bouchachia e baseado em sementes obtiveram desempenhos parecidos, sendo o baseado em sementes um pouco pior que o algoritmo de Bouchachia.

Intervalos com 5% de confiança foram construídos para o índice de rand corrigido em 4

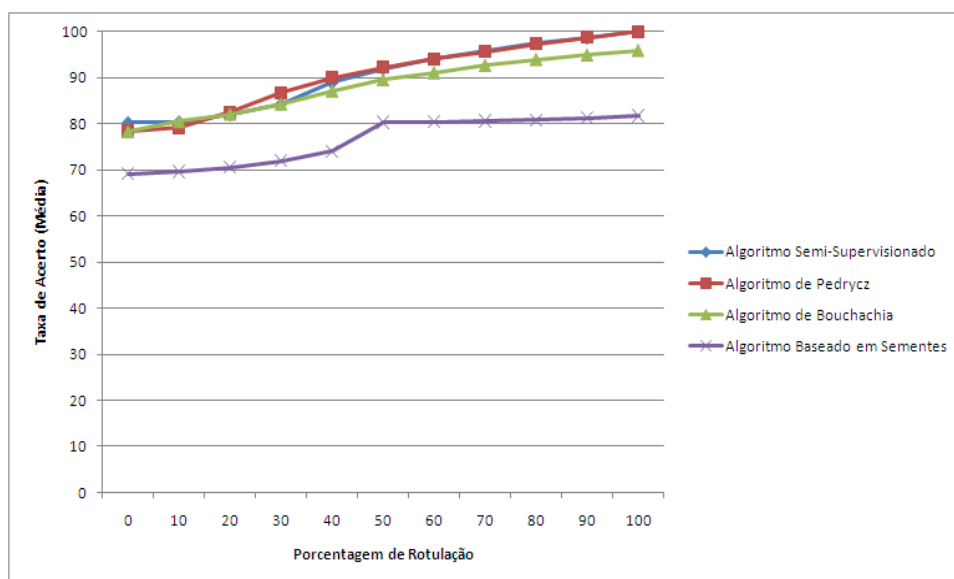


Figura 5.26 Estudo Comparativo: Base de Dados Sintética

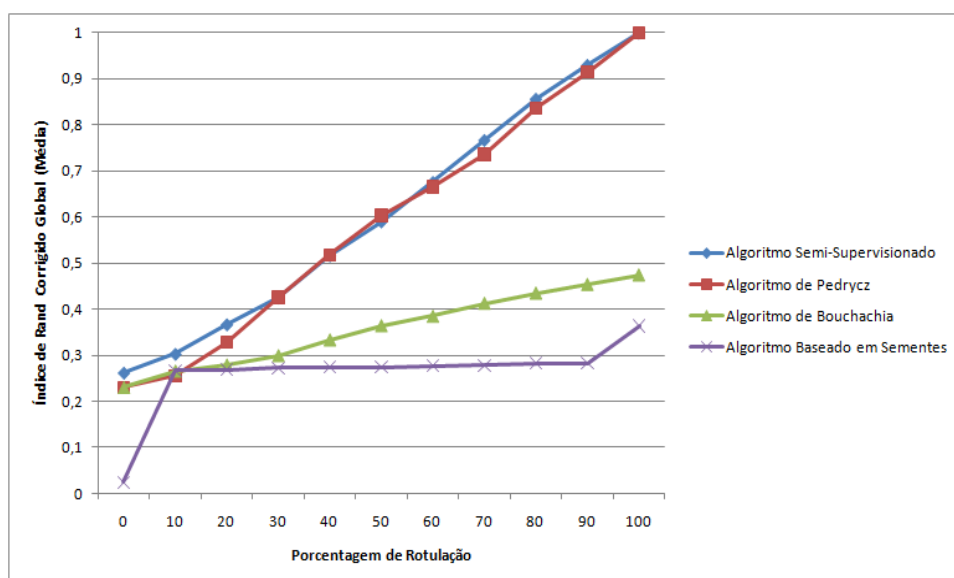


Figura 5.27 Estudo Comparativo do Índice de Rand Corrigido Global: Base de Dados Sintética

configurações do experimento. Com pouco dado rotulado, 10%, o algoritmo proposto obteve o melhor desempenho entre os algoritmos comparados. Os algoritmos de Bouchachia e Pedrycz tiveram desempenho equivalentes nessa mesma configuração. Nas configurações com 30% e 50% de dados rotulados, o algoritmo proposto e o algoritmo de Pedrycz obtêm os melhores desempenhos de forma equivalente. O algoritmo de Bouchachia obteve o terceiro melhor desempenho e o algoritmo baseado em sementes obteve o pior desempenho. Para 70% de dados

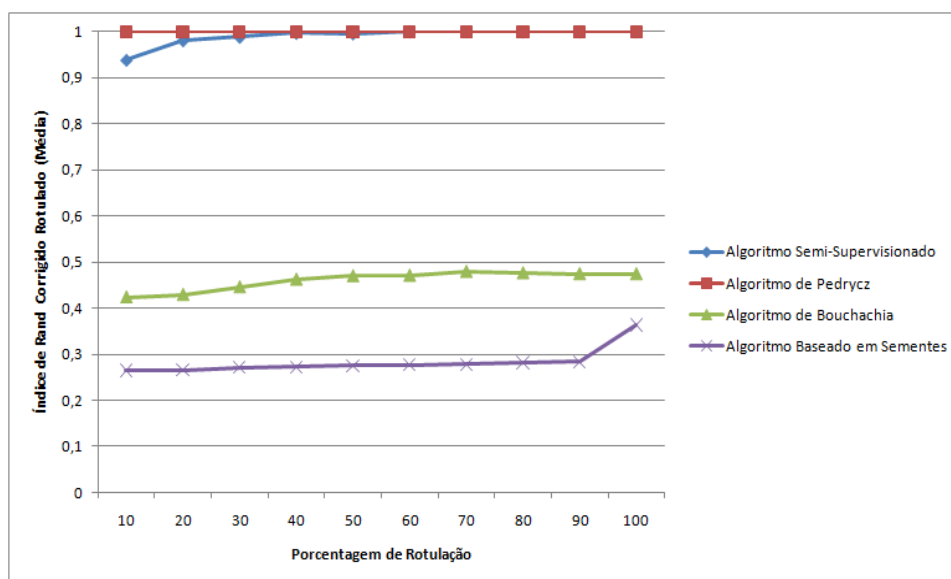


Figura 5.28 Estudo Comparativo do Índice de Rand Corrigido Rotulado: Base de Dados Sintética

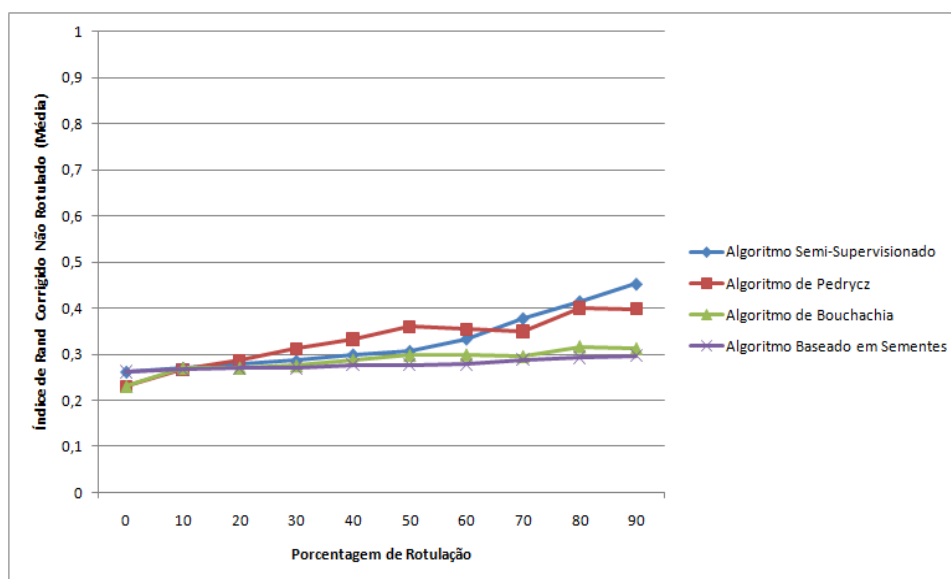


Figura 5.29 Estudo Comparativo do Índice de Rand Corrigido Não Rotulado: Base de Dados Sintética

rotulados, novamente o algoritmo proposto obteve o melhor desempenho, sendo seguido do algoritmo de Pedrycz, Bouchachia e o baseado em sementes, nessa ordem em termos de desempenho.

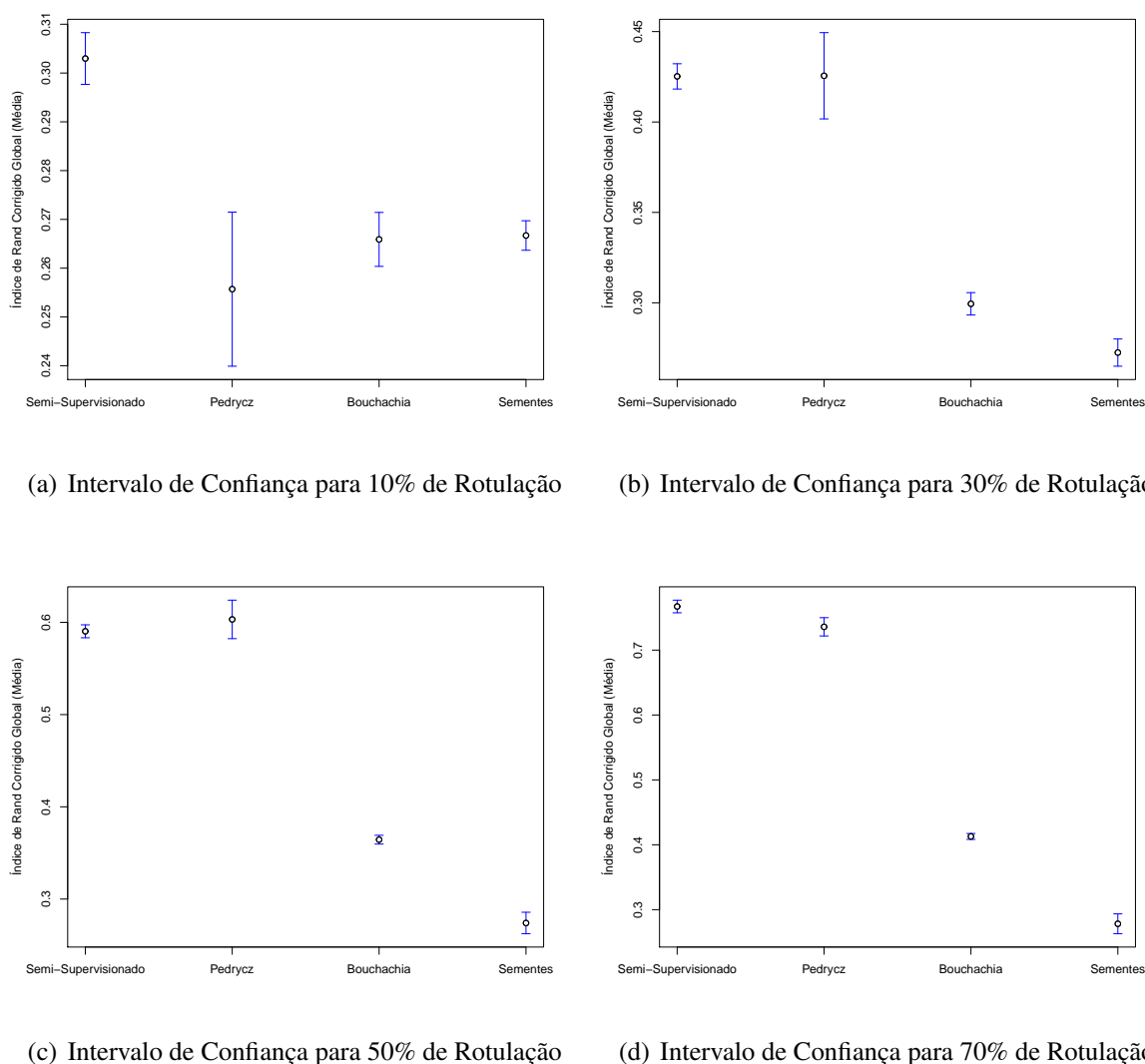


Figura 5.30 Intervalos de Confiança para a Base de Dados Sintética na Tarefa de Agrupamento

5.3.5 Resultados para a base de dados *Spam*

A base de dados *Spam* possui duas classes com um grande número de exemplos e com vários atributos na representação de cada padrão. Os resultados da taxa de acerto para essa base de dados são apresentados na Figura 5.31. Para poucos dados rotulados, 0% a 30%, o algoritmo proposto obteve os melhores resultados. Os algoritmos de Bouchachia e Pedrycz conseguem obter resultados equivalentes em todas as configurações nessa base de dados. Após 50% de dados rotulados, os dois algoritmos são os únicos que conseguem obter bons resultados e melhorar o desempenho com mais rótulos disponíveis no treinamento, alcançando o valor máximo quando todos os rótulos estão disponíveis. O algoritmo proposto não consegue melhorar seu desempenho quando aumenta o número de dados rotulados para o treinamento. O algoritmo

baseado em sementes aumenta seu desempenho com mais dados rotulados em seu treinamento, porém não alcança os outros algoritmos, obtendo o pior desempenho.

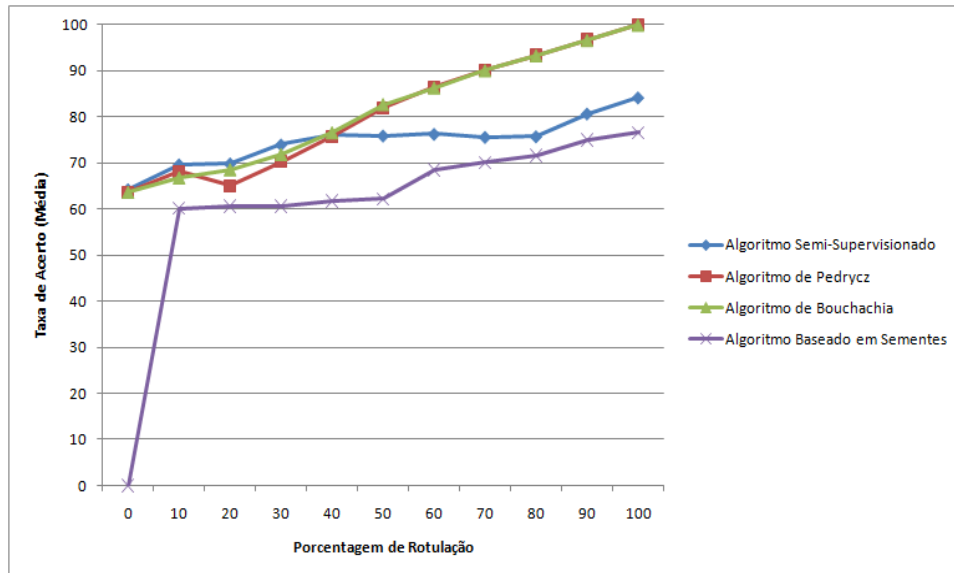


Figura 5.31 Estudo Comparativo: Base de Dados Spam

Os resultados para o índice de rand corrigido global nessa base de dados são apresentados na Figura 5.32. De 0% a 30% de dados rotulados, o algoritmo proposto obteve o melhor resultado para essa base de dados. O algoritmo de Bouchachia teve o segundo melhor desempenho, sendo seguido pelo desempenho do algoritmo de Pedrycz e do algoritmo baseado em sementes respectivamente. Na configuração de 40% de dados, os 3 algoritmos possuem um desempenho semelhante. De 50% a 100% de dados rotulados os algoritmos de Pedrycz e Bouchachia conseguem obter os melhores resultados, sendo equivalentes nessas configurações. Com todos os dados rotulados, os dois algoritmos conseguem obter o valor máximo do índice de rand. O algoritmo proposto consegue melhorar seu desempenho com mais dados rotulados, mas não consegue acompanhar a curva ascendente dos dois primeiros algoritmos, obtendo o terceiro melhor desempenho. O algoritmo baseado em sementes consegue aumentar seu desempenho com mais dados rotulados, mas obteve o pior desempenho dentre os algoritmos estudados.

Os resultados para o índice de rand corrigido do conjunto previamente rotulado é apresentado na Figura 5.33. Os algoritmos de Bouchachia e Pedrycz conseguem obter o índice máximo na maioria das configurações do experimento. O algoritmo proposto teve uma queda no desempenho na medida que mais dados eram rotulados. Essa queda explica o porque do algoritmo não conseguir bons resultados a partir de 50% de dados rotulados no índice de rand global. O algoritmo baseado em sementes possui um índice de valor 0 constante de 0% a 40% de dados rotulados. Após 50% de dados rotulados consegue aumentar seu desempenho, porém obteve o pior desempenho novamente.

Os resultados para o índice de rand corrigido para o conjunto de dados não rotulados é apresentado na Figura 5.34. Essa figura mostra que os algoritmos de Bouchachia e Pedrycz obtém

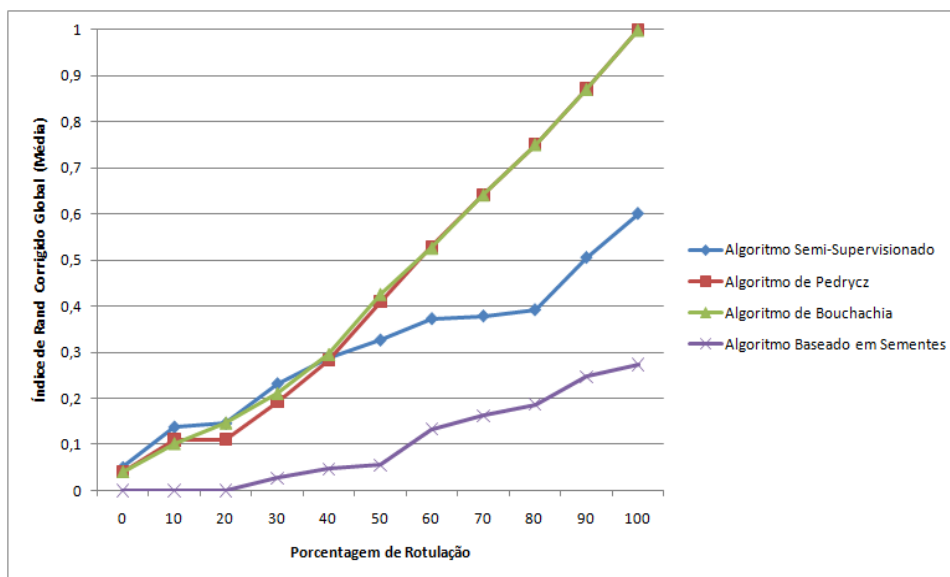


Figura 5.32 Estudo Comparativo do Índice de Rand Corrigido Global: Base de Dados Spam

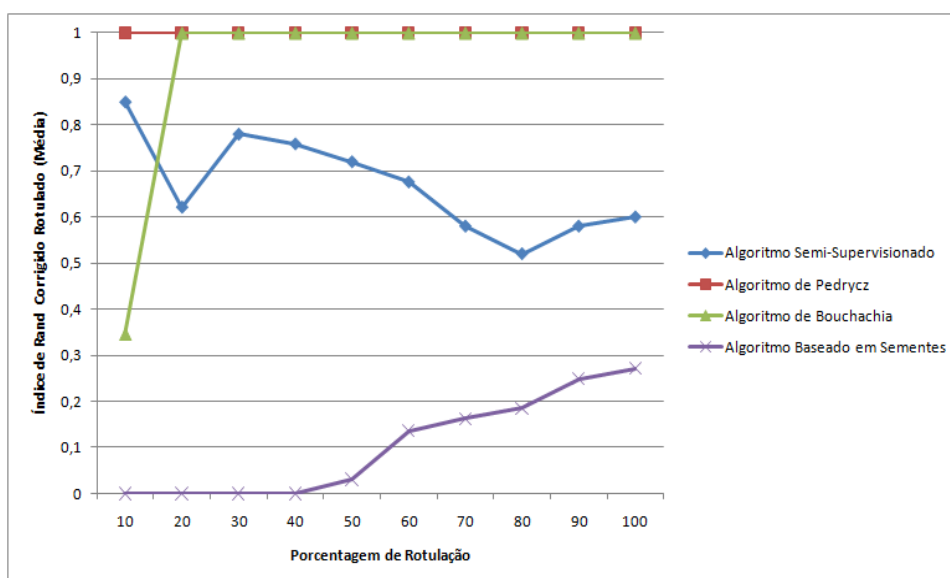


Figura 5.33 Estudo Comparativo do Índice de Rand Corrigido Rotulado: Base de Dados Spam

resultados equivalentes em todas as configurações do experimento. O algoritmo proposto teve um desempenho equivalente aos algoritmos de Pedrycz e Bouchachia quando havia poucos dados rotulados, de 0% a 20% de dados rotulados, com o algoritmo baseado em sementes obtendo o pior desempenho nessas configurações. De 30% a 100% de dados rotulados o algoritmo baseado em sementes consegue obter um índice maior que todos os algoritmos, tendo o melhor desempenho. O segundo melhor desempenho nessas configurações foi do algoritmo proposto.

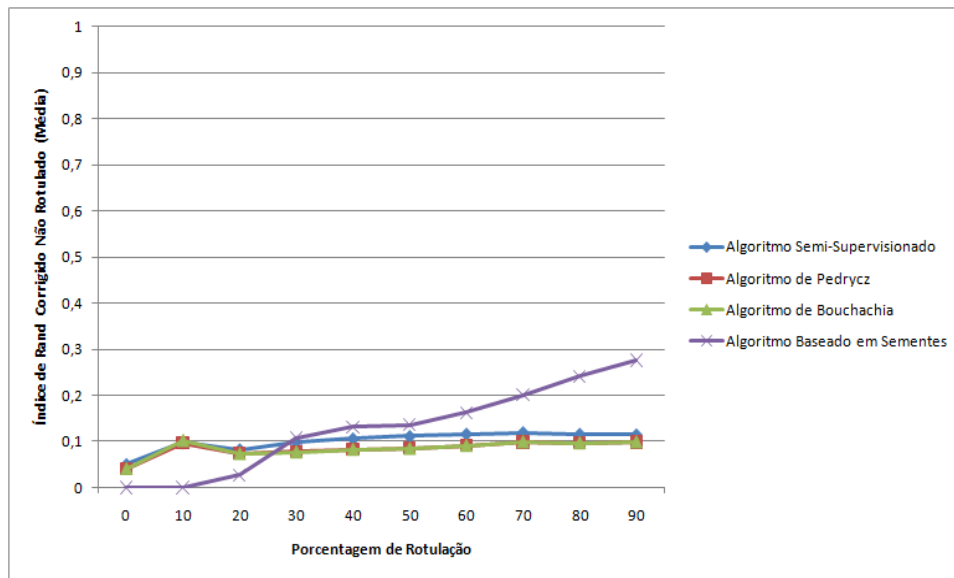


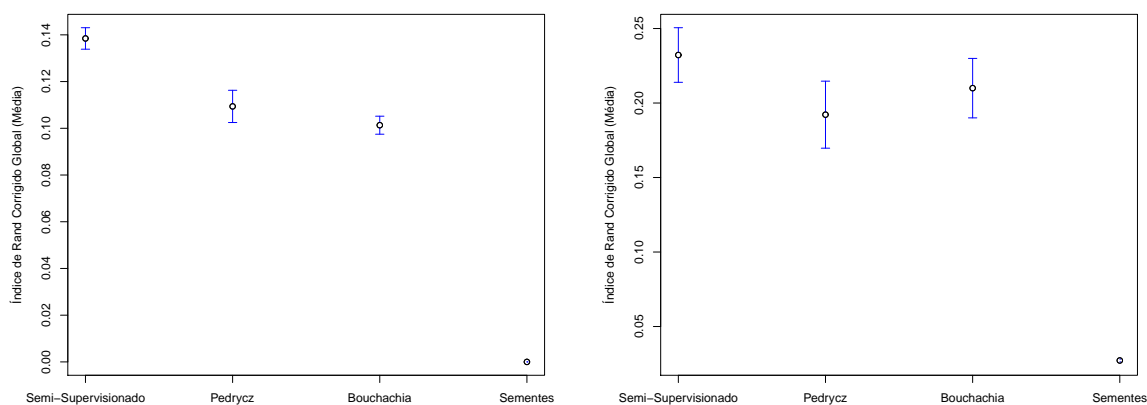
Figura 5.34 Estudo Comparativo do Índice de Rand Corrigido Não Rotulado: Base de Dados Spam

Os intervalos com 5% de confiança para as configurações com 10%, 30%, 50% e 70% de dados rotulados são apresentados na Figura 5.35. Com apenas 10% de dados rotulados, o algoritmo proposto se destaca dentre os demais algoritmos. Os algoritmos de Pedrycz e de Bouchachia possuem desempenho equivalentes para essa configuração e o algoritmo baseado em sementes obteve o pior desempenho. Para 30% de dados rotulados, os algoritmos possuem desempenho equivalentes, exceto o algoritmo baseado em sementes que obteve o pior desempenho novamente. Para 50% e 70% de dados rotulados, os algoritmos de Pedrycz e de Bouchachia obtêm o melhor desempenho de forma equivalente. O algoritmo proposto consegue o terceiro melhor desempenho seguido do desempenho do algoritmo baseado em sementes.

5.3.6 Discussão

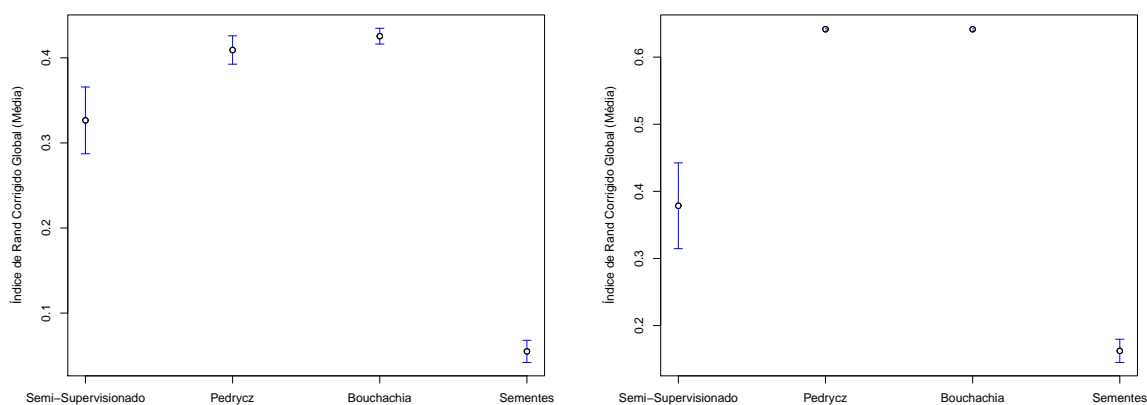
Podemos destacar nesse experimento que para algumas bases de dados (*Iris*, *Wine* e *Sintética*), o algoritmo proposto obteve os melhores resultados para a maioria das configurações de porcentagem de dados rotulados. Ainda, podemos observar que para a maioria das bases testadas, o algoritmo demonstrou aumento no desempenho da precisão na medida que mais dados rotulados eram acrescentados no treinamento do algoritmo. Desse modo, podemos afirmar que o desempenho do algoritmo proposto é compatível, e por vezes, melhor que algoritmos de agrupamento semi-supervisionados que minimizam uma função objetivo já consolidados na literatura.

O caso particular aconteceu para a base de dados *Spam*. O algoritmo proposto não conseguiu aumentar seu desempenho quando havia mais de 50% de dados disponíveis no treinamento. Outros pontos importantes que podem ser destacados é que algoritmo de Pedrycz obteve os melhores resultados dentre os outros algoritmos testados e chamou atenção pela simplicidade de sua implementação. O algoritmo de Bouchachia por sua vez é um algoritmo complexo que



(a) Intervalo de Confiança para 10% de Rotulação

(b) Intervalo de Confiança para 30% de Rotulação



(c) Intervalo de Confiança para 50% de Rotulação

(d) Intervalo de Confiança para 70% de Rotulação

Figura 5.35 Intervalos de Confiança para a Base de Dados *Spam* na Tarefa de Agrupamento

também obteve bons resultados e possui vários artigos mostrando que ele é melhor que muitos outros algoritmos semi-supervisionados. O que nos leva a crer que possuímos um ótimo algoritmo, melhor também que outros algoritmos não testados nesse trabalho. O algoritmo baseado em sementes obteve o pior resultado desse trabalho, apesar do aumento praticamente constante do seu desempenho com o aumento da porcentagem dos dados rotulados.

O índice de rand corrigido serviu para certificar que as partições formadas pelo algoritmo alcançam uma ótima qualidade. Seu gráfico muitas vezes acompanhava o gráfico da taxa de acerto, e também era capaz de perceber nuances que muitas vezes não eram refletidas nessa taxa. Isso aconteceu, por exemplo, na base de dados diabetes que o algoritmo baseado em sementes teve uma taxa de acerto constante mas conseguiu um pequeno aumento no índice de rand corrigido.

O índice de rand dos dados rotulados e não rotulados demonstraram o comportamento dos algoritmos nesses diferentes conjuntos de dados. Os resultados comprovaram que muitas vezes, o algoritmo proposto conseguiu obter melhores resultados no conjunto de dados não rotulados, que são os dados que mais importam no processo de agrupamento e de classificação. Às vezes, aconteceu de dados previamente rotulados serem rotulados numa classe diferente no processo de agrupamento. Ajustes, como não rotular os dados previamente rotulados, podem ser realizados para evitar que isso aconteça numa situação real.

Pra finalizar, podemos destacar que o novo algoritmo de agrupamento semi-supervisionado proposto obteve muito bons resultados para a maioria das base de dados avaliadas nesse trabalho. Também, podemos afirmar que o aumento de dados rotulados disponíveis no treinamento do algoritmo aumenta seu desempenho na grande maioria dos casos. Assim, enfatizamos mais uma vez que o novo algoritmo de agrupamento semi-supervisionado é uma alternativa viável e muitas vezes melhor que outros algoritmos já consolidados na literatura em aplicações que existam poucos ou muitos dados rotulados disponíveis.

Conclusões e Trabalhos Futuros

6.1 Conclusão

O aprendizado semi-supervisionado foi criado para atenuar limitações das duas aprendizagens mais clássicas, supervisionada e não supervisionada. A aprendizagem supervisionada precisa de dados rotulados suficientes para que o algoritmo possa alcançar um bom desempenho, enquanto a abordagem não supervisionada, geralmente, consegue desempenho menor que os algoritmos supervisionados. A aprendizagem semi-supervisionada precisa de poucos dados rotulados e consegue melhores resultados que as outras duas abordagens utilizando esses poucos dados rotulados em conjunto com dados não rotulados em seu treinamento. Esse trabalho propôs um novo algoritmo de agrupamento semi-supervisionado que alcança bons resultados com poucos dados rotulados.

O novo algoritmo de agrupamento semi-supervisionado minimiza uma função objetivo através de iterações sucessivas que otimizam os parâmetros dessa função através agrupamento de padrões. Esse algoritmo se mostrou bastante pertinente, já que apresentou desempenho melhor, ou pelo menos compatível, que algoritmos também de agrupamento semi-supervisionado que minimizam uma função objetivo já consolidados na literatura.

Para avaliar o desempenho do algoritmo, aplicamos uma validação cruzada adaptada para a aprendizagem semi-supervisionada. Além da validação cruzada, realizamos a tarefa de agrupamento, onde toda a base é treinada e avaliada. Os resultados desses experimentos foram bastante satisfatórios. Para poucos dados rotulados, o novo algoritmo de agrupamento semi-supervisionado obteve o melhor ou esteve entre os melhores desempenhos em relação aos outros algoritmos de agrupamento semi-supervisionado. À medida que mais dados eram disponibilizados para seu treinamento, o algoritmo melhorou seu desempenho na grande maioria dos casos, alcançando desempenho compatível com outros algoritmos semi-supervisionados.

O poder de classificação do novo algoritmo também foi avaliado, comparando-o com outros algoritmos totalmente supervisionados. Neste experimento, o algoritmo proposto conseguiu ser melhor que os algoritmos supervisionados em algumas bases de dados, conseguindo ser pelo menos compatível quando não alcançava o melhor desempenho.

Este trabalho contribuiu no domínio de algoritmos de agrupamento semi-supervisionados, abordando conceitos atuais e importantes. O novo algoritmo, fruto deste trabalho, alcança bom desempenho para poucos dados rotulados, podendo ser utilizado em aplicações reais onde rotular dados seja custoso. Além disso, pode ser mais uma excelente opção na possibilidade de se utilizar algoritmos supervisionados, em aplicações que existam dados rotulados suficientes para treinar um bom algoritmo.

6.2 Discussão

A utilização deste novo algoritmo conseguirá obter bons resultados em aplicações onde existam poucos dados rotulados. Além disso, também poderá obter bons resultados em casos onde existam dados suficientes para o treinamento de algoritmos supervisionados. Desse modo, o novo algoritmo alcança seu objetivo inicial que é ser mais uma opção em aplicações de classificação ou agrupamento de padrões.

Foram encontradas algumas dificuldades no desenvolvimento deste trabalho, como na aplicação da validação cruzada, pois alguns algoritmos precisavam de mais de uma iteração para o cálculo do grau de pertinência de um padrão nos grupos formados pelo algoritmo. Outra dificuldade está relacionada à padronização dos experimentos. Para uma avaliação justa, as inicializações dos algoritmos de agrupamento semi-supervisionados e seus critérios de parada foram padronizados e todos eles utilizaram as mesmas abordagens.

Por se tratar de equações matemáticas complexas, foi difícil transformá-las numa linguagem computacional. Dessa forma, a implementação dos experimentos foi pensada para se integrar a ferramenta de mineração de dados Weka. Desse modo, a entrada da ferramenta de testes é a mesma utilizada na ferramenta Weka, assim podemos realizar testes utilizando algoritmos implementados pelo Weka. Além disso, a ferramenta possui um *framework* para algoritmos de agrupamento. Desse modo, para adicionar um novo algoritmo desse tipo é preciso apenas estender a classe que contém as interfaces utilizadas nesse tipo de algoritmo e assim integrar um novo algoritmo a ferramenta.

6.3 Trabalhos Futuros

Neste trabalho foram utilizadas apenas bases de dados disponíveis em *sites* conhecidos da academia. O objetivo é estender os testes para bases de dados de aplicações reais. Duas aplicações serão testadas prioritariamente: uma base de dados esparsa de classificação de textos e outra de classificação de expressões gênicas, conhecidas por serem bastante extensas, sendo assim, difíceis de rotular por seres humanos.

A ferramenta ainda não possui uma interface amigável com o usuário. Um dos trabalhos futuros é implementar uma interface para que as configurações dos experimentos sejam realizadas.

Existem alguns algoritmos de agrupamento semi-supervisionado que utilizam *constraints* para otimizar o critério de agrupamento [BBBM06] em seu treinamento. Outra proposta é comparar o desempenho do novo algoritmo de agrupamento semi-supervisionado com algoritmos desse tipo.

O algoritmo proposto realiza uma minimização da função objetivo. A medida de similaridade utilizada nessa função objetivo é a distância básica de Mahalanobis, transformada em distância euclidiana. Como objetivo futuro, pretende-se realizar alterações nessa função de similaridade. As distâncias adaptativas permitem a construção de partições em diversos formatos, além do padrão circular, padrão formado pela distância euclidiana, e desse modo pode aprender estruturas mais complexas de distribuições de dados. Assim, pretende-se utilizar distâncias adaptativas com a finalidade de melhorar o desempenho do algoritmo proposto.

Referências Bibliográficas

- [AG05] M. R. Amini and P. Gallinari. Semi-supervised learning with an imperfect supervisor. *Knowledge and Information Systems*, 8:385–413, 2005.
- [AYM⁺02] M. Ahmed, S. Yamany, N. Mohamed, A. Farag A, and T. Moriarty. A modified fuzzy c-means algorithm for bias field estimation and segmentation of mri data. *IEEE Trans Med Imaging*, 21(3):193–199, 2002.
- [AZ07] R. K. Ando and T. Zhang. Two-view feature generation model for semi-supervised learning. *International Conference on Machine Learning*, pages 25–32, 2007.
- [BB06] M. F. Balcan and A. Blum. *Semi-supervised learning*, chapter An augmented pac model for semi-supervised learning. Mit Press, 2006.
- [BBBM06] S. Basu, M. Bilenko, A. Banerjee, and R. Mooney. *Semi-Supervised Learning*, chapter Probabilistic Semi-Supervised Clustering with Constraints, pages 7–39. Mit Press, 2006.
- [BC01] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. *International Conference on Machine Learning*, 2001.
- [Bez81] J.C. Bezdek. *Pattern Recognition With Fuzzy Objective Function Algorithms*. Plenum, 1981.
- [BHBC96] A. M. Bensaid, L. O. Hall, J. C. Bezdek, and L. P. Clarke. Partially supervised clustering for image segmentation. *Pattern Recognition*, 29(5):859–871, 1996.
- [Bis95] C. Bishop. *Neural networks for pattern recognition*. Oxford press, New York., 1995.
- [BLCS07] A. Benczúr, L. Lukács, K. Csalogány, and D. Siklósi. Semi-supervised learning: A comparative study for web spam and telephone user churn. *Graph Labelling Workshop and Web Spam Challenge*, 2007.
- [BM98] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. *Proceedings of the Workshop on Computational Learning Theory*, 1998.
- [BMN04] M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised large graphs. *Learning Theory*, pages 624–638, 2004.

- [Bou07] A. Bouchachia. Learning with partly data. *Neural Computing and application*, (16):267–293, 2007.
- [BP06] A. Bouchachia and W. Pedrycz. Data clustering with partial supervision. *Data Mining and Knowledge Discovery*, (12):47–78, 2006.
- [BS01] H. Bolfarine and C. Sandoval. *Introdução à Inferência Estatística*. Sociedade Brasileira de Matemática, 2001.
- [CC06] F. G. Cozman and I. Cohen. *Semi-supervised learning*, chapter Risks of Semi-Supervised Learning: How Unlabeled Data Can Degrade Performance of Generative Classifiers, pages 55–70. Mit Press, 2006.
- [CCC03] F. Cozman, I. Cohen, and M. Cirelo. Semi-supervised learning of mixture models. *International Conference on Machine Learning*, 2003.
- [CCdS03] I. G. Costa, F. A.T. De Carvalho, and M. C.P. de Souto. Comparative study on proximity indices for cluster analysis of gene expression time series. *Journal of Intelligent & Fuzzy Systems*, 13:133–142, 2003.
- [CCZ06] O. Chapelle, M. Chi, and A. Zien. A continuation method for semisupervised svms. *International Conference on Machine Learning*, 2006.
- [CK05] N. V. Chawla and G. Karakoulas. Learning from labeled and unlabeled data: An empirical study across techniques and domains. *Journal of Artificial Intelligence Research*, 23:331–366, 2005.
- [CL06] V. Cheng and C.H. Li. Personalized spam filtering with semi-supervised classifier ensemble. *International Conference on Web Intelligence*, 2006.
- [CS99] M. Collins and Y. Singer. Unsupervised models for named entity classification. *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 189–196, 1999.
- [CSK06] O. Chapelle, V. Sindhwani, and Keerthi. Branch and bound for semisupervised support vector machines. *Advances in Neural Information Processing Systems*, 2006.
- [CWB06] R. Collobert, J. Weston, and L. Bottou. Trading convexity for scalability. *International Conference on Machine Learning*, 2006.
- [CWS02] O. Chapelle, J. Weston, and B. Schoelkopf. Cluster kernels for semi-supervised learning. *NIPS*, 2002.
- [CZ05] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. *Workshop on Artificial Intelligence and Statistics*, 2005.

- [CZS06] O. Chapelle, A. Zien, and B. Scholkopf. *Semi-supervised learning*. MIT Press, 2006.
- [DHS01] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.
- [DKS02] R. Dara, S. Kremer, and D. Stacey. Clustering unlabeled data with soms improves classification of labeled real-world data. *World Congress on Computational Intelligence*, 2002.
- [DLR77] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.
- [Dub87] R. Dubes. How many clusters are best? an experiment. *Pattern Recognition*, 20(6):645–663, 1987.
- [EEAS06] K. G. Estes, J. L. Evans, M. W. Alibali, and J. R. Saffran. Can infants map meaning to newly segmented words? statistical segmentation and word learning. *Psychological Science*, 18(3):254–260, 2006.
- [Eve80] B. Everitt. *Cluster Analysis*. New York: Academic Press, 1980.
- [FEP95] M. Fahle, S. Edelman, and T. Poggio. Fast perceptual learning in hyperacuity. *Vision Research*, 35:3003–3013, 1995.
- [For73] G.D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.
- [Fra67] C.F. Fralick. Learning to recognize pattern without a teacher. *IEEE Transactions on Information theory*, 1(1):57–64, 1967.
- [FUS05] A. Fujino, N. Ueda, and K. Saito. Semi-supervised learning based on a hybrid of generative and discriminative models. *Information Technology Letters*, (4):161–164, 2005.
- [FW98] E. Frank and I. H. Witten. Generating accurate rule sets without global optimization. *Conference on Machine Learning*, pages 144–151, 1998.
- [GCB04] N. Grira, M. Crucianu, and N. Boujemaa. Unsupervised and semisupervised clustering: a brief survey. In *A Review of Machine Learning Techniques for Processing Multimedia Content*. Report of the MUSCLE European Network of Excellence, 2004.
- [GFL04] L. Grady and G. Funka-Lea. Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. *European Conference on Computer Vision*, 2004.

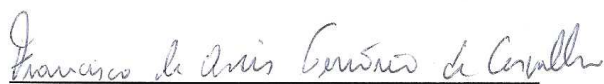
- [GL03] T. M. Gureckis and B. C. Love. Human unsupervised and supervised learning as a quantitative distinction. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(5):885–901, 2003.
- [GMW07] G. Gan, C. Ma, and J. Wu. *Data Clustering: Theory, Algorithms, and Applications*. SIAM, Society for Industrial and Applied Mathematics, 2007.
- [GZ00] S. Goldman and Y. Zhou. Enhancing supervised learning with unlabeled data. *International Conference on Machine Learning*, pages 327–334, 2000.
- [GZ06] A. Goldberg and X. Zhu. Seeing stars when there aren’t many stars: Graphbased semi-supervised learning for sentiment categorization. *Workshop on Textgraphs: Graph-based Algorithms for Natural Language Processing*, 2006.
- [HA85] L. J. Hubbert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:63–76, 1985.
- [HBH00] R. J. Hathaway, J. Bezdek, and Y. Hu. Generalized fuzzy c-means clustering strategies using lp-norm distances. *IEEE Transaction on Fuzzy Systems*, 8(5):576–582, 2000.
- [Hos73] D. W. J. Hosmer. On mle of the parameter of a mixture of two normal distribution when the sample size is small. *Communications in statistics*, (1):217–227, 1973.
- [HR68] H. O. Hartley and J. N. K. Rao. Classification and estimation in analysis of variance problems. *Review of International Statistical Institute*, 36:141–147, 1968.
- [HWP05] A. Holub, M. Welling, and P. Perona. Exploiting unlabelled data for hybrid object classification. In *Workshop in Inter-Class Transfer*, 2005.
- [JD88] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice Hall, New Jersey, 1988.
- [JMF99] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.
- [Joa99] T. Joachims. Transductive inference for text classification using support vector machines. *International Conference on Machine Learning*, Morgan Kaufmann:200–209, 1999.
- [Joa03] T. Joachims. Transductive learning via spectral graph partitioning. *International Conference on Machine Learning*, pages 290–297, 2003.
- [Jon05] R. Jones. *Learning to extract entities from labeled and unlabeled text*. PhD thesis, Carnegie Mellon University, 2005.
- [Lan06] P. Langley. Intelligent behavior in humans and machines. Technical report, Computational Learning Laboratory, CSLI, Stanford University, 2006.

- [LLW04] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transactions on Graphics*, 2004.
- [Lov02] B. C. Love. Comparing supervised and unsupervised category learning. *Psychonomic Bulletin & Review*, 9(4):829–835, 2002.
- [Mac67] J. Macqueen. Some methods for classification and analysis of multi-variate observation. *Berkley Symposium on mathematic statistics and probability*, 1:281–297, 1967.
- [MC86] G. W. Milligan and M. C. Cooper. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research*, 21:441–458, 1986.
- [Mit97] T. Mitchel. *Machine Learning*. McGraw Hill, 1997.
- [Mit99] T. Mitchell. The role of unlabeled data in supervised learning. *Proceedings of the Sixth International Colloquium on Cognitive Science*, 1999.
- [MLH04] B. Maeireizo, D. Litman, and R. Hwa. Co-training for predicting emotions with spoken dialogue data. *Annual Meeting of the Association for Computational Linguistics*, 2004.
- [MPC⁺08] V. Macario, R. B. C. Prudêncio, F. A. T. De Carvalho, L. Rodrigues L. R. Torres, and M. G. Lima. Automatic information extraction in semi-structured official journals. *Brazilian Symposium on Neural Networks*, 2008.
- [NJT05] Z. Y. Niu, D. H. Ji, and C. L. Tan. Word sense disambiguation using label propagation based semi-supervised learning. *Proceedings of the ACL*, 2005.
- [NMTM00] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39:103–134, 2000.
- [O’N78] T. O’Neill. Normal discrimination with unclassified observations. *Journal of the American Statistical Association*, 73(364):821–826, 1978.
- [Ped85] W. Pedrycz. Algorithms of fuzzy clustering with partial supervision. *Pattern Recognition*, 3:13–20, 1985.
- [PK08] N. N. Pise and P. Kulkarni. A survey of semi-supervised learning methods. *International Conference on Computational Intelligence and Security*, pages 30–34, 2008.
- [PW97] W. Pedrycz and J. Waletzky. Fuzzy clustering with partial supervision. *IEEE transactions on system, man and cybernetics*, 27(5), 1997.

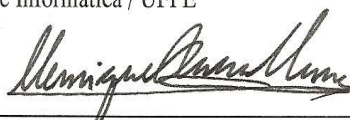
- [Qui93] R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.
- [RH05] C. Rosenberg, M. Hebert, and H. Schneiderman . Semi-supervised selftraining of object detection models. *Workshop on Applications of Computer Vision*, 2005.
- [RV95] J. Ratsaby and S. Venkatesh. Learning from a mixture of labeled and unlabeled examples with parametric side information. *Conference on Computational Learning Theory*, pages 412–417, 1995.
- [RWW03] E. Riloff, J. Wiebe, and T. Wilson. Learning subjective nouns using extraction pattern bootstrapping. *Conference on Natural Language Learning*, 2003.
- [San03] M. K. Sanches. Aprendizado de máquina semi-supervisionado: proposta e um algoritmo para rotular exemplos a partir de poucos exemplos rotulados. Master’s thesis, ICMC-USP, 2003.
- [SB88] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 5:513–523, 1988.
- [Scu65] H. J. Scudder. Probability of error of some adaptive pattern recognition machines. *IEEE Transactions on Information theory*, pages 363–371, 1965.
- [See00] M. Seeger. Learning with labeled and unlabeled data. Technical report, Institute for ANC, Edinburgh, UK, 2000. See <http://cwww.dai.ed.ac.uk/seeger/papers.html>.
- [See06] M. Seeger. *Semi-supervised learning*, chapter A taxonomy of semi-supervised learning methods. MIT Press, 2006.
- [SK06] V. Sindhwani and S. S. Keerthi. Large scale semisupervised linear svms. *Annual ACM Conference on Research and Development in Information Retrieval archive*, 2006.
- [SKC06] V. Sindhwani, S. Keerthi, and O. Chapelle. Deterministic annealing for semi-supervised kernel machines. *International Conference on Machine Learning*, 2006.
- [SM86] R. E. Stepp and R. S. Michalski. *Machine Learning: An Artificial Intelligence Approach*, volume 2, chapter Conceptual Clustering: Inventing Goal-Oriented Classifications of Structured Objects, pages 471–478. Morgan Kaufmann, 1986.
- [TK06] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Elsevier, 3 edition, 2006.
- [Vap95] V. N. Vapnik. The nature of statistical learning theory. *Springer-Verlag*, 1995.
- [Vap98] V. Vapnik. *Statistical learning theory*. Wiley-Interscience, 1998.

- [VC74] V. N. Vapnik and A. Y. Chervonenkis. *Theory of Pattern Recognition*. Verlag, 1974.
- [WCRS01a] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. *International Conference on Machine Learning*, pages 577–584, 2001.
- [WCRS01b] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, 2001.
- [Yar95] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. *Annual Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.
- [ZG04] Y. Zhou and S. Goldman. Democratic co-learning. *IEEE International Conference on Tools with Artificial Intelligence*, 2004.
- [ZGL03] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. *International Conference on Machine Learning*, 2003.
- [Zhu08] X. Zhu. *Semi-Supervised Learning Literature Survey*. Carnegie Mellon University, 2008.
- [ZL05a] Z. H. Zhou and M. Li. Semi-supervised regression with co-training. *International Joint Conference on Artificial Intelligence*, 2005.
- [ZL05b] Z. H. Zhou and M. Li. Tri-training: exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17:1529–1541, 2005.
- [ZRQK07] X. Zhu, T. Rogers, R. Qian, and C. Kalish. Humans perform semisupervised classification too. *AAAI Conference on Artificial Intelligence*, 2007.

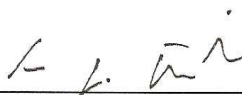
Dissertação de Mestrado apresentada por **Valmir Macário Filho** Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título “**Um Novo Algoritmo de Agrupamento Semi-Supervisionado Baseado no Fuzzy C-Means**”, orientada pelo **Prof. Francisco de Assis Tenório de Carvalho** e aprovada pela Banca Examinadora formada pelos professores:



Prof. Francisco de Assis Tenório de Carvalho
Centro de Informática / UFPE



Prof. Henrique Pacca Loureiro Luna
Instituto de Computação / UFAL



Prof. Ivan Gesteira Costa Filho
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 28 de agosto de 2009.



Prof. Nelson Souto Rosa
Vice-Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.