UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Felipe Leite Fagundes

Aprendizado de Métricas Utilizando uma Função de Distância Parametrizada e o Algoritmo *k-means* com Aplicação na Solução de Problemas de Classificação

Juiz de Fora

2017

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Felipe Leite Fagundes

Aprendizado de Métricas Utilizando uma Função de Distância Parametrizada e o Algoritmo *k-means* com Aplicação na Solução de Problemas de Classificação

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Raul Fonseca Neto
Coorientador: Carlos Cristiano H. Borges

Juiz de Fora
2017

Ficha catalográfica elaborada através do programa de geração
automática da Biblioteca Universitária da UFJF,
com os dados fornecidos pelo(a) autor(a)

Fagundes, Felipe Leite.
Aprendizado de Métricas Utilizando uma Função de Distância Parametrizada e o Algoritmo k-means com Aplicação na Solução de Problemas de Classificação / Felipe Leite Fagundes. -- 2017.
45 f.

Orientador: Raul Fonseca Neto
Coorientador: Carlos Cristiano Hasenclever Borges
Dissertação (mestrado acadêmico) - Universidade Federal de Juiz de Fora, ICE/Engenharia. Programa de Pós-Graduação em Ciência da Computação, 2017.

1. Aprendizado de Máquinas. 2. Classificação. 3. Análise de Dados. I. Fonseca Neto, Raul, orient. II. Borges, Carlos Cristiano Hasenclever, coorient. III. Título.

Felipe Leite Fagundes

Aprendizado de Métricas Utilizando uma Função de Distância Parametrizada e o Algoritmo *k-means* com Aplicação na Solução de Problemas de Classificação

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Aprovada em 31 de Agosto de 2017.

BANCA EXAMINADORA

Prof. Dr. Raul Fonseca Neto - Orientador
Universidade Federal de Juiz de Fora

Prof. Dr. Carlos Cristiano H. Borges - Coorientador
Universidade Federal de Juiz de Fora

Prof. Dr. Saulo Moraes Villela
Universidade Federal de Juiz de Fora

Prof. Dr. Antônio de Pádua Braga
Universidade Federal de Minas Gerais

AGRADECIMENTOS

Sob a óptica de que *"a mente que se abre a uma nova ideia, jamais voltará ao seu tamanho original"* (Albert Einstein), embarquei e naveguei em novos mares, até então, completamente desconhecidos por mim. Os desafios foram enormes. Mas todo o suporte que recebi, seja técnico ou emocional, das pessoas engajadas com a minha causa, foi extremamente necessário para que eu conseguisse superar todos esses desafios com êxito. Meras palavras escritas aqui não são capazes de expressar todo o sentimento de gratidão que tenho por essas pessoas! Mas o registro faz-se necessário e é a mínima forma de reconhecimento, que ficará gravado para a posteridade.

A todos os meus familiares, alicerce para minha vida, em especial, meu pai (Pedro), minha mãe (Idalina), minha irmã (Letícia) e a minha amada esposa (Laura), agradeço por todo o carinho, força e amor que sempre dispuseram. Agradeço, também, pela compreensão da perene escassez do tempo durante o curso e pela paciência ao me ouvirem falar tantas vezes sobre IA! Sem o apoio de vocês, não teria conseguido!

Ao Raul, agradeço imensamente por acreditar em meu potencial desde o início, por ter sido meu mentor e um grande propulsor de ideias, fundamentais para meu desenvolvimento acadêmico. Ao Cristiano, agradeço por todas as contribuições técnicas desde os primórdios do desenvolvimento desta dissertação, quando ainda navegávamos sem um rumo definido. Da mesma forma, agradeço aos demais professores das disciplinas cursadas durante o Mestrado, fundamentais para a abertura da minha mente às novas ideias. São eles: Alex, Heder, Henrique Hippert, Jairo, Marcelo, Saul, Saulo e Wagner.

À Julieta, gostaria de agradecer por ter agido não só como chefe para mim em seu cargo, mas como uma verdadeira líder, incentivando-me a sempre buscar mais, mantendo a motivação para conciliar trabalho e estudos com afinco. Agradeço, ainda, a todos os funcionários do ICE, sobretudo, à Sarah, pela seriedade de seu trabalho no PGCC.

Sem dúvida, não poderia ficar de fora dessa lista de agradecimentos os colegas que fiz durante esse tempo no PGCC. Dentre esses, os amigos do "Linbo": João, Karen e Marcelo. Além de todos os demais com quem tive a grata oportunidade de conviver no Programa.

Foram apenas dois anos e meio, mas um turbilhão de mudanças na vida pessoal que me colocaram em xeque mais de uma vez. E se cheguei até aqui, podem ter certeza que a contribuição de vocês foi fundamental. OBRIGADO A TODOS!!!

"As invenções são, sobretudo,
resultado de um trabalho
teimoso."

Santos Dumont

RESUMO

A utilização de diferentes métricas em algoritmos de aprendizado de máquinas pode mudar completamente os resultados de análises realizadas em bases de dados. Variar as maneiras de medir distâncias ou similaridades dos dados pode gerar reflexos para a captura de informações dessas bases e, com isso, influenciar diretamente a tomada de decisões. Neste sentido, métodos de aprendizagem de métricas têm sido abordados e aplicados em diversos ramos das pesquisas que manipulam bases de dados, com a finalidade de encontrar métricas mais adequadas para soluções de problemas de análise de *cluster*, classificação, mineração de dados, dentre outros relacionados ao reconhecimento de padrões de dados. O método de aprendizado de métricas utilizado como base deste trabalho foi originalmente formulado como um problema de otimização, com o objetivo de minimizar um conjunto parametrizado de distâncias de Mahalanobis. No método original é necessário estabelecer uma lista com pares de vetores similares ou dissimilares, que possibilitam a correção dos parâmetros para medição das distâncias. Já neste trabalho é proposto um novo método, que não necessita da comparação par a par entre vetores, mas apenas da comparação de distâncias de cada vetor do conjunto de treinamento com dois centroides: o definido pelo algoritmo *Seeded k-means* e o definido por um especialista como sendo um centroide esperado. A distância entre o vetor e os dois centroides é usada como fator global de correção dos parâmetros para medição das distâncias. Os novos parâmetros para medição de distâncias alteram a forma como os vetores são agrupados, melhorando sensivelmente os resultados em relação à métrica Euclideana. A maior contribuição deste estudo foi a formulação de um método para aprendizado desses parâmetros que reduzisse a complexidade em tempo em relação a outros métodos de aprendizado propostos na literatura, denominado MAP – Método de Aprendizado de Parâmetros. O MAP demonstrou melhoras significativas para problemas de classificação em diversas bases de dados do *UCI Machine Learning Repository* com métricas aprendidas em conjuntos de treinamento.

Palavras-chave: Aprendizado de Máquinas. Classificação. Análise de dados.

ABSTRACT

The use of different metrics in machine learning algorithms is able to change the results of analyzes carried out in databases. By varying how to measure distances or data similarities we can generate reflexes for information capture, which can influence the decision-making. In this sense, metric learning methods have been approached and applied in several branches of the research in the world, in order to find better metrics for problems of cluster analysis, classification, data mining, among others related data pattern recognition. The metric learning method used as the basis of this work was originally formulated as an optimization problem, in order to minimize a parameter set of Mahalanobis distances. In the original method, it is necessary to define a list of similar or dissimilar vectors pairs, which allow the correction of the distance measurement parameters. In this work, a new method is proposed, which does not require the pairwise comparison, but only the distance comparison from each vector of a training set to two points: one defined by the Seeded k-means and other defined by an expert as being an expected centroid. The distance between the vector and the two centroids is used as correction factor of the parameters for measuring distances. The new learned parameters for distances measurement can change the clusters improving the results compared to the Euclidean metric. The major contribution of this study was the formulation of a method to learn these parameters that reduces the complexity in time if compared to other methods proposed in the literature. The proposal of PLM – Parameter Learning Method – have been demonstrated significant improvements in classification problems for several UCI Machine Learning Repository databases.

Keywords: Machine Learning. Classification. Data Analysis.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

A necessidade de obtenção de formas adequadas para medir a distância ou a semelhança entre dados é um tema presente em problemas de aprendizagem de máquinas, reconhecimento de padrões, recuperação da informação e até bioinformática, mas a definição de métricas ideais para uso cada problema é, geralmente, complexa. Isso motivou o surgimento de estudos sobre o aprendizado de métricas e o tema tem despertado cada vez mais interesse de pesquisadores nos últimos anos, sobretudo na área de aprendizado de máquinas (BELLET et al., 2013).

A aprendizagem métrica pode trazer vantagens sempre que a noção de distância entre instâncias desempenha um papel importante. Neste sentido, verifica-se sua aplicação em problemas muito distintos, tais como na previsão de conexões em redes (SHAW et al., 2011), na representação de estados na aprendizagem por reforço (TAYLOR et al., 2011), na recomendação de músicas (MCFEE et al., 2012), em problemas de partição (LAJUGIE et al., 2014), na verificação de identidades (BEN et al., 2012), na identificação de semelhanças entre páginas da Web (LAW et al., 2012), dentre outros. Nota-se, portanto, a atualidade e importância do tema no meio científico.

O sucesso de um algoritmo de aprendizado de máquinas depende, criticamente, do tipo de métrica utilizada para a avaliação da distância ou da medida de similaridade entre vetores no espaço de entrada (FAGUNDES et al., 2016). No algoritmo *k-means* (MACQUEEN, 1967), por exemplo, foi demonstrado que a forma como as instâncias são agrupadas usando métricas aprendidas varia consideravelmente em comparação com o agrupamento feito utilizando a tradicional métrica Euclidiana. Nos algoritmos de classificação, como o k-NN (COVER; HART, 1967), por exemplo, a alteração da métrica também pode provocar variações nos resultados. Especificamente no k-NN, as novas instâncias são classificadas com base nas distâncias dos k vizinhos mais próximos e essa definição de proximidade depende completamente da métrica adotada.

Nesta dissertação é apresentado um novo método para aprendizagem de métricas denominado Método de Aprendizado de Parâmetros (MAP). O desenvolvimento do MAP foi inspirado por um problema de otimização dos parâmetros da matriz de distâncias de Mahalanobis, proposto por Xing et al. (2002). No modelo proposto pelos autores, o

somatório do conjunto das distâncias entre pares de vetores similares, que eram predefinidos por um especialista, deveria ser minimizado a partir da correção dos parâmetros da matriz de Mahalanobis. Já neste trabalho, abordou-se a possibilidade de aprendizado de parâmetros para alterações na métrica Euclideana por meio da comparação entre os dados de uma base, os centroides de seus prováveis *clusters* e os centros predefinidos por um especialista como os centros esperados. Desta forma, no MAP, o número de comparações necessárias cresce de maneira linear em relação ao número de instâncias da base de dados, enquanto que nos demais métodos encontrados na literatura durante o desenvolvimento deste trabalho, praticamente todos inspirados no método proposto por Xing et al. (2002), esse número de comparações cresce de maneira quadrática, devido à necessidade de comparação par a par entre das instâncias.

1.1 DEFINIÇÃO DO PROBLEMA E OBJETIVOS

Considerando a dificuldade para a definição de métricas específicas em cada tipo base de dados ocorrer de forma escalável em relação ao número de instâncias e de dimensões, neste trabalho buscou-se a implementação de um método de aprendizado de métricas que fosse capaz de ser aplicado não só a pequenos conjuntos de dados ou a bases com baixa dimensionalidade, mas também a conjuntos com alta dimensionalidade ou grande número de instâncias, sendo essas as maiores vantagens do método em relação aos trabalhos propostos na literatura dessa área.

O MAP então é proposto com a finalidade de aprender métricas, as quais seriam capazes de aprimorar as soluções encontradas pelo algoritmo *k-means* (MACQUEEN, 1967). Adicionalmente, pretende-se a extensão do uso das métricas aprendidas pelo MAP para aplicação em outros algoritmos de aprendizado de máquinas que, tradicionalmente, utilizam a métrica Euclideana.

1.2 CONTRIBUIÇÕES

A maior contribuição deste trabalho foi o desenvolvimento do MAP, que permite o aprendizado de métricas de forma linear em relação ao número de instâncias e atributos de uma base de dados. Diferente dos demais métodos de aprendizado de métricas encontrados na literatura, no MAP não há necessidade de comparação par a par entre todos os vetores

uma base de dados ou de seu conjunto de treinamento, o que aumentaria a complexidade do problema. Dessa forma, o método torna-se escalável em relação ao número de instâncias. Além disso, a opção de utilizar uma variação do algoritmo *k-means* – o *Seeded k-means* (BASU et al., 2002) – como o direcionador para o aprendizado foi fundamental, pois é considerado um algoritmo de complexidade esperada linear em relação ao número de instâncias e atributos, e que tende a convergir com maior velocidade, comparando-se com o *k-means* tradicional.

1.3 ORGANIZAÇÃO

Este trabalho foi organizado da seguinte maneira: no Capítulo 2 são apresentados trabalhos de destaque na área de aprendizado de métricas e o mecanismo de funcionamento dos algoritmos utilizados nos experimentos. No Capítulo 3 é apresentado o MAP, suas características e limitações. O Capítulo 4 aborda os experimentos e resultados obtidos com a aplicação do MAP. Finalizando, no Capítulo 5, apresentam-se algumas conclusões sobre a abordagem desenvolvida, bem como algumas ideias sobre trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 APRENDIZADO DE MÉTRICAS

Os algoritmos de aprendizagem métricas têm o objetivo de encontrar parâmetros para medição de distâncias entre vetores (distribuídos em $\mathbb{R}^n$) que sejam capazes de melhorar o desempenho de preditores, como, por exemplo, algoritmos de agrupamento, classificadores e regressores. Uma nova métrica é aprendida a partir de um conjunto de dados e faz com que o preditor obtenha melhores resultados, comparando-se com os resultados obtidos com a métrica original para aquele conjunto. Esse processo está resumido na Figura 2.1, adaptada a partir do trabalho de Bellet et al. (2013).

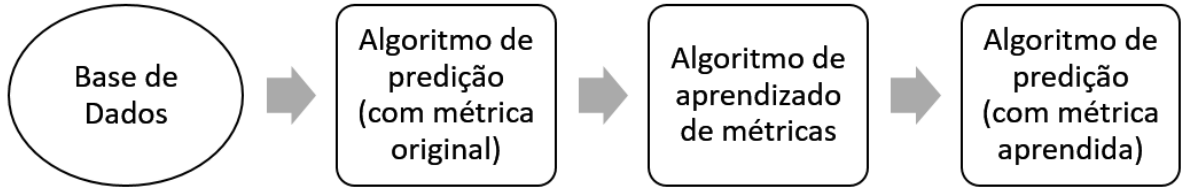


Figura 2.1: Processo de Aprendizado de Métricas

Embora possam ser identificados estudos da década de 1980 que abordam a aprendizagem de métricas, como o trabalho de Short and Fukunaga (1981), e na década de 1990, com trabalho de Hastie and Tibshirani (1996), o tema ganhou relevância na comunidade científica a partir do ano de 2002, com a publicação de Xing et al. (2002), que o formulou a aprendizagem como um problema de otimização convexa.

Pouco antes do trabalho de Xing et al. (2002), Wagstaff et al. (2001) propuseram o uso de informações adicionais no algoritmo *k-means* (MACQUEEN, 1967), que consistiam em inserir restrições de similaridade ou dissimilaridade entre pares de vetores de um subconjunto dos dados. Essas restrições foram denominadas "*must-link*", significando que dois vetores devem pertencer ao mesmo *cluster*, e "*cannot-link*", significando que dois vetores devem pertencer a *clusters* diferentes. Porém esse método, denominado *COP k-means*, não garantia a convergência para uma solução que atendesse a todas as restrições de similaridade e dissimilaridade estabelecidas.

Considerando a não garantia de convergência do *COP k-means*, o trabalho de Xing et

al. (2002) abordou a possibilidade do uso de aprendizado de métricas, de forma que as restrições "*must-link*" e "*cannot-link*" pudessem ser atendidas na formação dos *clusters* do *COP k-means*.

Para entender o método de aprendizagem proposto por Xing et al. (2002), é fundamental ter em mente o conceito da distância de Mahalanobis (d_M) (MAHALANOBIS, 1936), retratada na Equação 2.1, sendo M a matriz de covariância entre os vetores x_i e x_j.

$$d_M(x_i, x_j) = ||x_i - x_j||_M = \sqrt{(x_i - x_j)^T M^{-1}(x_i - x_j)} \tag{2.1}$$

Genericamente, a distância parametrizada (d_A) entre dois vetores x_i e x_j pode ser representada conforme a Equação 2.2, que substitui a matriz inversa de covariâncias M^{-1} por uma matriz de parâmetros A.

$$d_A(x_i, x_j) = ||x_i - x_j||_A = \sqrt{(x_i - x_j)^T A(x_i - x_j)} \tag{2.2}$$

Sendo que as seguintes propriedades devem ser atendidas:

- $d_A(x_i, x_j) \geq 0$ (não negatividade)
- $d_A(x_i, x_i) = 0$ (identidade)
- $d_A(x_i, x_j) = d_A(x_j, x_i)$ (simetria)
- $d_A(x_i, x_j) \leq d_A(x_i, x_k) + d_A(x_k, x_j)$ (desigualdade triangular)

Para isso, faz-se necessário que a matriz de parâmetros A seja semidefinida-positiva ($A \succeq 0$), não nula e simétrica. Nota-se aqui que, se a matriz A for a matriz identidade, obtém-se a tradicional distância Euclideana.

A modelagem da proposta de Xing et al. (2002), então, ocorre da seguinte forma: utiliza-se como entrada um conjunto de relações de similaridade S, envolvendo pares de vetores (x_i, x_j) que pertencem a um mesmo *cluster*, e um conjunto de relações de dissimilaridade D, envolvendo pares (x_i, x_j) que pertencem a *clusters* distintos. Então é computado o somatório das distâncias ao quadrado $d_A^2(x_i, x_j)$ de todos os pares de vetores do conjunto S, considerando que a matriz A é definida inicialmente como a matriz inversa da matriz de covariância (M^{-1}). Ou seja, é a distância de Mahalanobis ao quadrado.

Após isso, a matriz A tem seus parâmetros (autovalores) corrigidos iterativamente, de forma que seja minimizado esse somatório. Contudo, tornam-se necessários mecanismos para que a matriz A se mantenha semidefinida-positiva, bem como, não zere todos os seus parâmetros. Para isso, foi formulada uma restrição de dissimilaridade que mantivesse a característica de convexidade. A formulação do problema é resumida a seguir.

$$min \sum_{(x_i,x_j)\in S} ||x_i - x_j||_A^2$$

Sujeito a:

$$\sum_{(x_i,x_j)\in D} ||x_i - x_j||_A \geq 1, \tag{2.3}$$

$$A \succeq 0.$$

$S : (x_i, x_j) \in S$ se x_i e x_j são similares

$D : (x_i, x_j) \in D$ se x_i e x_j são dissimilares

Tendo em vista a alta complexidade para resolução de um problema de programação quadrática para problemas com muitas dimensões, Xing et al. (2002) preveem o caso mais simples de aprendizado de métricas utilizando apenas da diagonal da matriz A, de forma que a necessidade de aprendizado desses parâmetros crescesse linearmente em relação ao número de dimensões, reduzindo a complexidade do problema. Nota-se que a utilização e aprendizado apenas da diagonal da matriz A é equivalente aprender pesos para cada uma das dimensões do problema.

Ainda, como alternativa para reduzir a complexidade para o caso de uso de todos os parâmetros da matriz A, os autores propõem a formulação dual, conforme a seguir (2.4).

$$g(A) = max \sum_{(x_i,x_j)\in D} ||x_i - x_j||_A \tag{2.4}$$

Sujeito a:

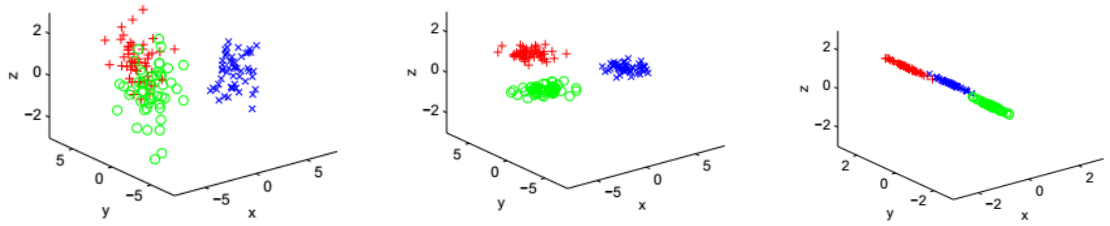
$$\sum_{(x_i,x_j)\in S} ||x_i - x_j||_A^2 \leq 1,$$

$$A \succeq 0.$$

Com a função objetivo sendo linear, torna-se possível a utilização do método iterativo do gradiente ascendente para correções dos parâmetros da matriz A, conforme 2.5.

$$A := A + \alpha \nabla_A g(A) \tag{2.5}$$

Visualmente, os efeitos do aprendizado de métricas podem ser notados na Figura 2.2, retirada do trabalho de Xing et al. (2002). Em (a) são apresentados os dados originais. Em (b) os dados com as métricas aprendidas usando apenas a matriz diagonal. Em (c) estão os dados agrupados de acordo com todos os parâmetros aprendidos na matriz A.



(a) (b) (c)

Figura 2.2: Demonstração dos Efeitos do Aprendizado de Métricas

Todavia, mesmo com a simplificação de uso da diagonal da matriz A e com a formulação dual, destaca-se que o método ainda não é facilmente escalável, tendo em vista a necessidade de comparação par a par dos vetores no conjunto de dados utilizado para o aprendizado da métrica ideal.

Schultz and Joachims (2004) apresentam um método para aprender métricas por meio de comparações relativas entre subconjuntos de triplas de instâncias de uma base de dados. As restrições são definidas por $(x_i, x_j, x_k) \in P$, de forma que uma instância x_i deve ser

mais próxima de x_j do que de x_k. Além da nova maneira de formularem as restrições, os autores propõem uma generalização da matriz de Mahalanobis, que é reescrita como $M = AW^TA^T$, em função de uma matriz de parâmetros não nula A e de um vetor de pesos não negativos W, que seja não nulo. Desse maneira, a parametrização torna-se ainda mais flexível. A formulação para a distância parametrizada $d_{A,W}$ é apresentada em 2.6.

$$d_{A,W}(x_i, x_j) = \sqrt{((x_i - x_j)^T A)W^T(A^T(x_i - x_j))} \tag{2.6}$$

No caso simplificado em que A é a matriz de identidade, gera-se o modelo de distância Euclideana parametrizada pelo vetor de pesos W, conforme 2.7.

$$d_{I,W}(x_i, x_j) = \sqrt{((x_i - x_j)^T I)W^T(I^T(x_i - x_j))} = \sqrt{(x_i - x_j)^T W^T(x_i - x_j)} \tag{2.7}$$

Com a nova formulação para o conjunto de restrições (P) e para cálculo das distâncias, Schultz and Joachims (2004) propõem o aprendizado de métricas resolvendo um problema de otimização convexa semelhante ao SVM (Máquina de Vetores Suporte - Cortes and Vapnik (1995)) para encontrar o vetor de pesos de máxima margem, conforme 2.8.

$$min_W ||AW^TA^T||_F^2 + C\sum_{i,j,k}\xi_{i,j,k} \tag{2.8}$$

Sujeito a:

$$d_{A,W}^2(x_i, x_k) - d_{A,W}^2(x_i, x_j) \geq 1 - \xi_{i,j,k}, \forall(x_i, x_j, x_k) \in P$$

onde $||.||_F^2$ é o quadrado da norma Frobenius, ξ_{ijk} são variáveis de folga e $C \geq 0$ é um parâmetro de regularização. Essa abordagem fica, portanto, restrita ao aprendizado do vetor de parâmetros W, enquanto a matriz A deve ser definida manualmente.

Weinberger et al. (2006) já propõem o aprendizado de métricas com foco em classificação pelo algoritmo k-NN. Essa abordagem tem o objetivo de maximização da margem por meio de uma função objetiva convexa. O método, denominado LMNN (*Large Margin Nearest Neighbor* - larga margem do vizinho mais próximo), tem sua formulação semelhante ao método de Schultz and Joachims (2004) (2.8), utilizando as restrições de comparações relativas P, mas também fazendo uso das comparações par a par S. Além disso, ainda há

o parâmetro μ, que pondera a atração entre pares de instâncias semelhantes (x_i, x_j) e a repulsão de x_i e x_k conforme descrito em 2.9.

$$min_M(1-\mu)\sum_{(x_i,x_j)\in S} d_M^2(x_i,x_j) + \mu\sum_{i,j,k}\xi_{i,j,k} \tag{2.9}$$

Sujeito a:

$$d_M^2(x_i,x_k) - d_M^2(x_i,x_j) \geq 1-\xi_{i,j,k}, \forall(x_i,x_j,x_k)\in P$$

onde $\mu \in [0,1]$

Um *survey* é apresentado por Yang and Jin (2006) com um estudo abrangente sobre os problemas de aprendizado de métricas com a citação de diversos trabalhos neste campo, todos com a mesma lógica de uso das restrições de similaridade entre pares de dados $(x_i, x_j) \in S$, dissimilaridades $(x_i, x_k) \in D$ ou distâncias relativas de triplas $(x_i, x_j, x_k) \in P$.

No trabalho de Jain et al. (2009), é apresentado um método de aprendizado de métricas *online*. A solução proposta é baseada em consecutivas predições da similaridade, também a partir da apresentação de pares de vetores. Ao receber um novo par de vetores, o algoritmo decide, com base na matriz de Mahalanobis com parâmetros atualizados *online*, se a distância computada confirma se os vetores são similares ou não. Caso haja discordância da informação de similaridade com o indicado pelo algoritmo, uma perda é imputada na matriz de parâmetros. O objetivo do aprendizado é a minimização desta perda ao longo de todo período de observação. Os autores ressaltam que a solução deste tipo de problema é muito importante para tarefas de aprendizado *online*, relacionadas, sobretudo, ao reconhecimento de objetos em cenas com movimento.

Bellet et al. (2013) e Kulis et al. (2013) apresentam revisões recentes dos trabalhos mais relevantes sobre aprendizado de métricas e nota-se que persiste nos principais trabalhos desde o início dos anos 2000 a aprendizagem utilizando as restrições *must link* e *cannot link*, tais como Davis et al. (2007), Weinberger and Saul (2009) e Yang et al. (2013).

Em resumo, Bellet et al. (2013) apresenta as principais características dos algorigmos de aprendizado de métricas de acordo com a Figura 2.3.

Descrevendo cada

- Paradigma de aprendizado:

  Supervisionado - o algoritmo de aprendizagem métrica tem acesso a um conjunto de instâncias de treinamento rotuladas, onde cada exemplo de treinamento é

Figura 2.3: Propriedades dos Algoritmos de Aprendizado de Métrica

composto de uma instância $x_i \in X$ e uma etiqueta (ou classe) $y_i \in Y$. Y é um conjunto discreto e finito de rótulos que, na prática, é frequentemente usado para gerar conjuntos específicos de restrições de pares de instâncias similares S, dissimilares D ou triplas P, que possam estabelecer relações de proximidade entre as instâncias.

Fracamente supervisionado - o algoritmo de aprendizado de métricas não tem acesso aos rótulos de instâncias de treinamento individuais. Apenas são fornecidas a ele informações sob a forma de conjuntos de restrições S, D, P. Esta é uma configuração significativa em uma variedade de aplicações onde os dados rotulados são de difícil obtenção, enquanto as informações laterais são baratas. Por exemplo, cliques nos resultados do mecanismo de pesquisa, citações entre artigos ou *links* em uma rede. Todavia, por meio da computação dos fechos transitivos, determinar todos os subconjuntos fechados ou *clusters*.

Semissupervisionado - além da supervisão (total ou fraca), o algoritmo tem acesso a uma amostra (normalmente grande) de instâncias não marcadas para as quais nenhuma informação está disponível. Podem ser usados para evitar *overfitting* quando os dados ou informações laterais rotulados são escassos.

- Tipo de métrica:

  Linear - seu poder é limitado, mas elas são mais fáceis de otimizar, geralmente levam a formulações convexas que permitem a otimização global da solução, mas evitando *overfitting*.

  Não linear - muitas vezes dão origem a formulações não convencionais, sujeitas à otimalidade local. Mas têm capacidade de capturar variações não-lineares nos dados.

- Escalabilidade:

  Em dimensões: capacidade do algoritmo ser executado em problemas em que as instâncias possuam um número elevado de dimensões. Todavia, como muitas vezes os algoritmos de aprendizado de métricas são formulados com o objetivo de aprendizagem de matrizes d x d, projetar algoritmos que escalem bem em número de dimensões é um desafio considerável.

  Em instâncias: capacidade do algoritmo ser executado em problemas em que haja um numero elevado de instâncias. Como muitas vezes os algoritmos de aprendizado de métricas são formulados para satisfazer as restrições entre pares ou triplas de instâncias, obter escala em relação ao número de instâncias também se torna um desafio.

- Otimalidade da Solução:

  Global - o algoritmo garante que a métrica aprendida é a melhor possível, considerando as restrições daquele problema. Este é essencialmente o caso das formulações convexas.

  Local - o algoritmo não garante que a solução encontrada é o ótimo global.

- Seleção de Características:

  Ocorre quando o algoritmo permite a anulação de parâmetros que definem o peso de uma ou mais dimensões.

2.2 ANÁLISE DE AGRUPAMENTOS

A análise de agrupamentos, ou análise de *cluster*, é um tipo de problema que propõe o agrupamento de instâncias de uma base de dados de acordo com alguma regra que defina a similaridade entre essas instâncias, separando-as em dois ou mais grupos distintos, também denominados *clusters*. Para o entendimento do funcionamento do MAP, é necessário primeiro entender um dos algoritmos de agrupamento mais tradicionais da literatura: o *k-means*.

2.2.1 *K-MEANS*

O algoritmo *k-means* (MACQUEEN, 1967), originariamente não supervisionado, tem por objetivo agrupar os vetores de uma base de dados em k diferentes *clusters*, minimizando o somatório J dos quadrados das distâncias Euclideanas entre todos os pares (x_i, x_j) de vetores que estejam alocados um mesmo *cluster*. Contudo, verifica-se que J equivale à soma dos quadrados das distâncias Euclideanas de cada vetor do espaço de entrada ao centroide c_l do respectivo *cluster* S_l. O cálculo do centroide c_l é dado pela média dos vetores que compõem o *cluster* S_l. Sendo assim, obtém-se J conforme Equação 2.10.

$$J = \sum_{l} \sum_{(x_i,x_j)\in S_l} ||x_i - x_j||_2^2 = \sum_{l} n_l \sum_{(x_i,x_j)\in S_l} ||x_i - c_l||_2^2 = \tag{2.10}$$

$$\sum_{l} n_l \sum_{i|x_i \in S_l} (x_i - c_l)^T (x_i - c_l)$$

sendo $||.||_2^2$ o quadrado da distância Euclideana, l o identificador do *cluster* e n_l o número de instâncias do *cluster* S_l.

Como o objetivo do algoritmo *k-means* é a minimização da distância *intracluster*, obtém-se a função objetivo 2.11.

$$min(J) = min \sum_{l} \sum_{(i|x_i \in S_l)} ||x_i - c_l||_2^2 \tag{2.11}$$

$$\text{para } l \in \{1, \ldots, k\} \text{ e } i \in \{1, \ldots, n\},$$

sendo n o número total de instâncias da base de dados do problema.

Esta função pode ser minimizada de duas formas. A primeira, conhecida como modo *online*, utiliza o método do gradiente estocástico descendente. Neste caso, computa-se a

derivada parcial da função de distância entre os vetores do conjunto de dados em relação ao seus respectivos centroides, o que determina a direção do gradiente. Ou seja:

$$\frac{\partial J(x_i)}{\partial c_l} = -2(x_i - c_l) \tag{2.12}$$

O centroide vencedor é definido considerando a menor distância Euclidiana do vetor em relação a todos os centroides, ou seja:

$$l = arg\ min||x_i - c_l||_2, \forall i \in \{1, \ldots, n\}. \tag{2.13}$$

Em seguida, corrige-se a posição do centroide vencedor em direção ao vetor x_i utilizando-se a seguinte equação de correção:

$$c_{l(t+1)} := c_{l(t)} + \eta(x_i - c_{l(t)}) \tag{2.14}$$

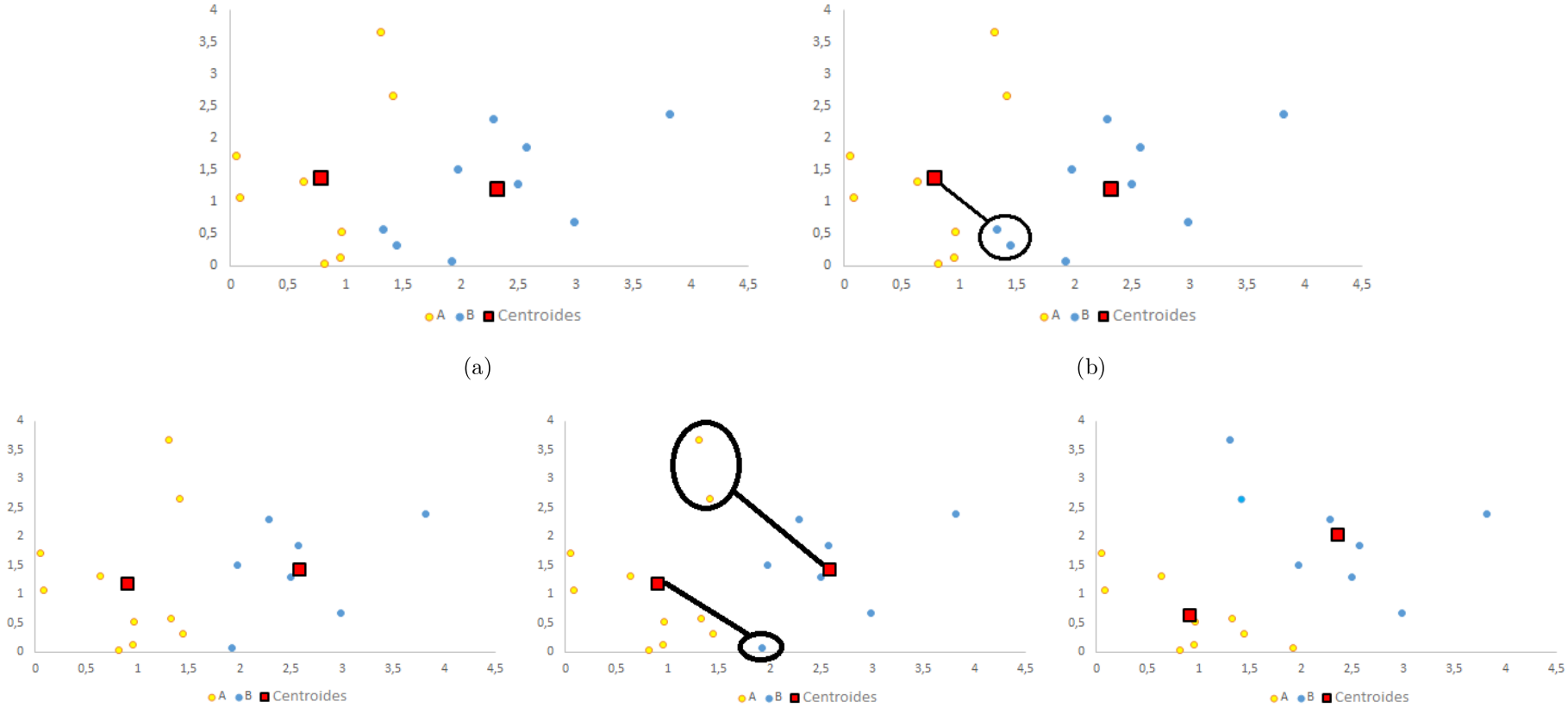
$$\forall l \in \{1, \ldots, k\}, \forall i \in \{1, \ldots, n\}, \text{ com } 0 < \eta < 1.$$

O algoritmo converge após um número finito de iterações para uma taxa de aprendizado apropriada, ou pode-se definir como critério de parada um J aceitável.

Outra forma de minimização da função de distância do algoritmo *k-means* é conhecida como *offline*, sendo mais utilizada devido a sua implementação mais simples e a ausência da necessidade da taxa de aprendizado. Nesta versão, cada centroide é recalculado como a média dos vetores que pertencem àquele *cluster*:

$$c_{l(t+1)} = \frac{1}{n_l} \sum_{i|x_i \in S_l} x_i, \forall l \in \{1, \ldots, k\}, \forall i \in \{1, \ldots, n\}. \tag{2.15}$$

Os valores dos centroides são computados de forma iterativa, sempre que houver uma mudança em algum rótulo do vetor de dados. Assim, a cada iteração, todos os subconjuntos de *clusters* S_l devem ser atualizados considerando um novo esquema de rótulos dos dados com base nos centroides atualizados. A convergência é alcançada quando não ocorrerem mais modificações no esquema de rótulos ou nos subconjuntos. Em ambos os algoritmos os valores iniciais dos centros são estabelecidos de forma randômica (HAMERLY; ELKAN, 2002).



(a) (b)

(c) (d) (e)

Figura 2.4: Iterações do *k-means offline*

Na Figura 2.4 é demonstrado um exemplo da realização das iterações do *k-means offline* de (a) até a convergência em (e). Dois centros aleatórios são escolhidos e os pontos mais próximos ao primeiro centro são marcados em amarelo (grupo A), enquanto os mais próximos do segundo são marcados em azul (grupo B). Então são calculados os respectivos centroides desses grupos (em vermelho), conforme (a). Nota-se em (b) que, há dois pontos em azul mais próximos do centroide do grupo A do que do centroide do grupo B. Logo, esses dois pontos são realocados no grupo A e é recalculado o centroide de cada grupo, agora com a nova formação, conforme (c). O procedimento segue até que não façam mais realocações, conforme ocorre em (e).

2.2.2 *SEEDED K-MEANS*

Basu et al. (2002) propuseram uma variante do algoritmo *k-means* considerando a possibilidade de utilização de vetores com *clusters* inicialmente conhecidos, que seriam as "sementes" para definição dos centros de inicialização do *k-means*. Para isso, é necessário que, pelo menos para cada *cluster*, exista uma semente. O centro de inicialização é definido pela média das sementes de cada *cluster*. Resultados experimentais demonstraram que o *Seeded k-means* converge mais rápido e tem a capacidade de aumentar a acurácia do *k-means* tradicional. Esse fato seria esperado, considerando-se que o *Seeded k-means* utiliza mais informações que o *k-means* tradicional, sendo um algoritmo semissupervisionado. Nota-se no exemplo da Figura 2.5 que os centroides das sementes são mais próximos dos centroides finais do que os dois centros iniciais aleatórios.

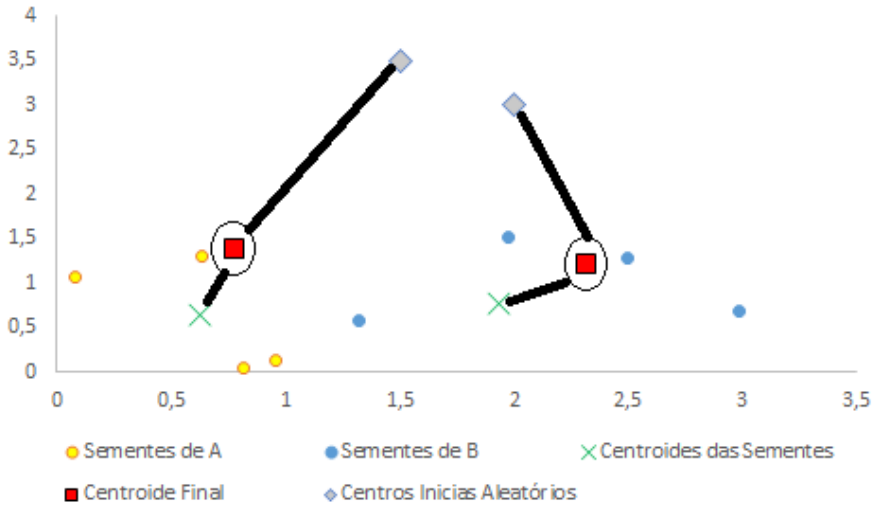


Figura 2.5: Convergência do *Seeded k-means*

3 DEFINIÇÃO DO MAP

Para formulação do Método de Aprendizado de Parâmetros (MAP), foram considerados os seguintes desafios:

- Na prática, pode ser ineficiente inserir restrições de similaridade entre todos os pares de um conjunto de dados ou triplas. Essas restrições estão propostas em praticamente todos os métodos de aprendizado de métricas citados nos *surveys* mais atuais sobre o tema (Bellet et al. (2013) e Kulis et al. (2013)). Contudo, podem tornar inviável o aprendizado de métricas em bases de dados com número elevado de instâncias, uma vez que o aprendizado acarretaria, no mínimo, em uma complexidade $O(n^2)$, considerando a necessidade de combinações de todos os n pares de vetores;

- O uso de todos os parâmetros da matriz A, demonstrada na Equação 2.2, pode tornar o aprendizado de métricas inviável em bases de dados de alta dimensionalidade, uma vez que é uma matriz quadrada bidimensional. Ou seja, no mínimo o aprendizado seria $O(m^2)$.

A seguinte observação, feita por Edwards and Cavalli-Sforza (1965), tornou-se preponderante para resolução do primeiro desafio: seja $C_l = \{x_1, x_2, ..., x_n\}$ um conjunto de vetores. A soma do quadrado das distâncias entre todos os pares de vetores é igual a soma do quadrado das distâncias de cada vetor ao centroide desse conjunto, multiplicado pela quantidade de vetores. Sendo c_l o centroide de um conjunto de pontos C_l, temos:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} ||x_i - x_j||_2^2 = n\sum_{i=1}^{n} ||x_i - c_l||_2^2 \tag{3.1}$$

A função de distância do algoritmo *k-means* é construída com base na equivalência da soma dos quadrados das distâncias entre todos os pares vetores de um mesmo *cluster* e a soma dos quadrados das distâncias desses vetores em relação aos seus respectivos centroides, multiplicada pelo número de vetores do respectivo *cluster*, conforme visto na Equação 2.10. Ao utilizar um método análogo de correção do centroide proposto pelo algoritmo *k-means*, descarta-se a necessidade de comparação par a par dos vetores, sendo apenas necessária a comparação destes com os centroides.

Com relação ao segundo desafio, para evitar a necessidade de correção de todos os parâmetros na matriz A_{m*m}, optou-se pela utilização do vetor de parâmetros $W = [w_1, w_2, ..., w_m]$ de dimensão m, que equivale à diagonal da matriz A_{m*m}. Dessa maneira, somente será necessária a correção de m parâmetros para o aprendizado de uma métrica que se adapte melhor a cada base de dados, ao invés de m^2, como seria ao se utilizar a matriz A_{m*m}.

Considerando-se os dois desafios, obtém-se o seguinte modelo para a formulação do MAP para o *k-means*:

$$J = \sum_l \sum_{(x_i,x_j)\in S_l} ||x_i - x_j||^2_W =$$

$$= \sum_l n_l \sum_{i|x_i \in S_l} (x_i - c_l)^T W (x_i - c_l) = \tag{3.2}$$

$$= \sum_l n_l \sum_{i|x_i \in S_l} W(x_i - c_l)^2 =$$

$$= \sum_l n_l \sum_{i|x_i \in S_l} w_1 * (x_{i(1)} - c_{l(1)})^2 + w_2 * (x_{i(2)} - c_{l(2)})^2 + ... + w_m * (x_{i(m)} - c_{l(m)})^2$$

Portanto, a única alteração necessária no algoritmo *k-means* está na determinação do centro vencedor $l = arg\, min_j ||x_i - c_l||_w, \forall i \in \{1, \ldots, n\}$, onde agora devem ser computadas as distâncias parametrizadas pelo vetor W.

A estratégia do MAP para correção do W tem como base a sucessiva redução das diferenças da função de distância entre os centroides inicialmente calculados pela média das sementes de entrada do algoritmo *Seeded k-means*, que são utilizados neste trabalho como os centro que um especialista espera obter, e os centroides obtidos ao término das iterações do algoritmo. Para isso, até que a diferença entre os centroides se iguale ou até que seja alcançado o número de iterações t_{max}, formulou-se a seguinte função de correção δ, a partir das derivadas parciais em relação a W:

$$\delta = \frac{J'_{exp} - J'_{kme}}{J'_{exp} + J'_{kme}} \tag{3.3}$$

$$J'_{exp} = \frac{\partial J_{exp}}{\partial W} = \sum_{l} n_l \sum_{i|x_i \in S_l} ||x_i - c_{l(exp)}||^2_W \tag{3.4}$$

$$J'_{kme} = \frac{\partial J_{kme}}{\partial W} = \sum_{l} n_l \sum_{i|x_i \in S_l} ||x_i - c_{l(kme)t}||^2_W \tag{3.5}$$

$$W_{t+1} := W_t - \eta * \delta_t \tag{3.6}$$

sendo $c_{l(exp)}$ o centroide do *cluster* l definido pelas semente do *Seeded k-means* e $c_{l(kme)t}$ o centroide encontrado na rodada t do *Seeded k-means*.

Ou seja, a cada iteração t, o vetor de parâmetros W é corrigido pela função de correção normalizada δ a uma taxa arbitrada η. Após isso, um novo *Seeded k-means* é executado com as mesmas sementes fornecidas inicialmente, mas utilizando o vetor de parâmetros W aprendido para o cálculo da distância entre os vetores e seus candidatos a centroides. Dessa forma, espera-se que os novos agrupamentos obtidos possam ter seus centroides mais próximos dos centroides do especialista, definidos inicialmente pelas sementes do *Seeded k-means*. Quando J_{exp} iguala-se ao J_{kme}, entende-se que o melhor vetor de parâmetros W foi obtido para aquela base de dados, de acordo com as informações inseridas inicialmente pelo especialista.

O uso do vetor de parâmetros equivale à atribuição de pesos para cada dimensão. Dessa forma, o problema passa também a ter forte relação com métodos de seleção de características.

3.1 TAXA DE APRENDIZADO

A taxa de aprendizado η é a razão em que o vetor de parâmetros W é corrigido ao longo das iterações do algoritmo. O uso de taxas elevadas ou muito reduzidas pode comprometer a varredura em busca das melhores soluções para W. Neste trabalho foram feitos experimentos com um amplo espectro de taxas, na tentativa de definição de uma taxa "ideal". Contudo, essa taxa ideal depende também do número máximo de iterações desejado de características de cada base de dados. Logo, com a finalidade de estabelecer uma metodo-

logia de testes, a taxa foi arbitrada em $\eta = 0,01$ para todos os testes realizados. Todavia, para impedir o descumprimento da restrição de não negatividade de W, foi necessário o acréscimo de uma etapa de revisão da taxa em caso de geração de algum w_i negativo. Ao identificar essa situação, a taxa η era corrigida para que ocorresse no máximo $w_i = 0$. Dessa forma, aquela dimensão seria, então, eliminada.

3.2 TAXA DE REFINAMENTO DE APRENDIZADO

Uma das restrições do MAP é a não garantia da sua convergência, estando sujeito a mínimos locais. A taxa de aprendizado η então passa a ter importância fundamental na busca de melhores métricas, contudo ela é definida arbitrariamente ou por meio de experimentos para verificação de uma taxa ideal. Tendo em vista este cenário, foi considerada a possibilidade de utilização de uma taxa de refinamento ϕ para correção da taxa de aprendizado η. Essa taxa tem o objetivo de aumentar ou diminuir η de acordo com o resultado obtido na rodada anterior. Consiste num acréscimo no tamanho do passo em caso de piora em relação à solução anterior ou de redução do tamanho do passo, em caso de melhora. Também com a finalidade de estabelecer uma metodologia de testes, foi introduzida uma taxa de refinamento $\phi = 1\%$ na tentativa de busca dos melhores parâmetros para o vetor W.

3.3 CRITÉRIO PARA DEFINIÇÃO DO W_{OPT}

O objetivo do MAP é possibilitar o aumento da acurácia de algoritmos de análise de *cluster* ou de classificação em comparação com a tradicional métrica Euclideana. Para definição do melhor vetor de parâmetros W_{opt}, é considerada a acurácia de cada *Seeded k-means* executado no conjunto de treinamento Z, no qual todas as sementes possuem os rótulos R definidos por um especialista, conforme Algoritmo 1.

3.4 PSEUDOCÓDIGO

Algoritmo 1: Método de Aprendizado de Parâmetros (MAP)

Entrada:

conjunto de treinamento (sementes): $Z = \{(x_i)\}$, com cardinalidade m e $i = 1, ..., n$;

rótulos do conjunto de treinamento: $R = \{(r_i)\}$, com k diferentes rótulos;

taxa de aprendizado: η;

taxa de refinamento: ϕ;

número máximo de iterações: max .

Saída:

melhor vetor de parâmetros: W_{opt}

início

$W_1 \leftarrow 1$;

$Acuracia_{opt} \leftarrow 0$;

$C_{exp} \leftarrow CalculaCentroides(Z, R)$;

para *t variando de 1 até max* **faça**

$\{C_{(kme)t}, Acurcia_t\} \leftarrow$ *Seeded k-means*(Z, R, C_{exp}, W_t);

$J'_{kme} = \sum_l n_l \sum_{i|x_i \in S_l} ||x_i - c_{l(kme)t}||^2_W; \forall l \in \{1, \ldots, k\}, \forall i \in \{1, \ldots, n\}$;

$J'_{exp} = \sum_l n_l \sum_{i|x_i \in S_l} ||x_i - c_{l(exp)}||^2_W \forall l \in \{1, \ldots, k\}, \forall i \in \{1, \ldots, n\}$;

$\delta = \frac{J'_{exp} - J'_{kme}}{J'_{exp} + J'_{kme}}$

$W_{t+1} = W_t - \eta * \delta$

se $[Acurcia_t > Acurcia_{opt}]$ **então**

$Acurcia_{opt} \leftarrow Acurcia_t$

$W_{opt} \leftarrow W_t$

$\eta \leftarrow \eta * (1 - \phi)$

fim se

senão

$\eta \leftarrow \eta * (1 + \phi)$

fim se

fim para

fim

retorna W_{opt}

4 EXPERIMENTOS E RESULTADOS

4.1 BASES DE DADOS TESTADAS

Para demonstrar a amplitude da aplicabilidade do MAP, foram testadas diferentes bases de dados com características heterogêneas. Todas elas foram extraídas do repositório de dados *UCI-Machine Learning*, exceto uma (Artificial), gerada artificialmente com a finalidade de demonstrar o comportamento do método do MAP visualmente em duas dimensões. A seguir há um pequeno descritivo das origens das bases de dados do *UCI-Machine Learning*:

- *Arcene*: dados obtidos pela fusão de três conjuntos de dados de espectrometria de massa, que indicam a quantidade de cada tipo de proteínas em soros humanos. Com base nessas características, pacientes com câncer devem ser diferenciados de pacientes saudáveis.

- *Breast cancer Wisconsin (Diagnostic)*: dados com características de células cancerosas e a classificação clínica em maligno ou benigno.

- *Glass*: dados sobre as características de diferentes tipos de vidros. O estudo desse tipo de problema é motivado principalmente por investigações criminalísticas.

- *Ionosphere*: a base de dados contempla características de sinais de radar, com a finalidade de definir a qualidade dos sinais.

- *Iris*: base de dados de três tipos de planta Iris que são classificadas de acordo com as suas características.

- *LSVT*: cada atributo dessa base corresponde à aplicação de um algoritmo de processamento de sinal de fala que visa caracterizar objetivamente o sinal. As vozes são de pacientes portadores da doença de Parkinson que estão em tratamento. São definidas quais vozes estão com a comunicação aceitável e quais não estão.

- *Parkinsons*: conjunto de dados composto por uma série de medidas biomédicas de voz de 31 pessoas, 23 com doença de Parkinson. O objetivo principal é discriminar pessoas saudáveis daqueles com doença de Parkinson.

- *Pima Indians diabetes*: base de dados com diversas características sobre mulheres índias da etnia Pima. A finalidade desses dados é auxiliar na classificação entre índias diabéticas ou não.

- *Sonar*: dados obtidos pelo sinal de sonares, com o objetivo de identificar se há metais em meio a rochas.

- *Wine*: dados com o resultados de uma análise química dos vinhos cultivados na mesma região da Itália, mas derivados de três cultivares diferentes. A análise determinou as quantidades de 13 constituintes encontrados em cada um dos três tipos de vinhos.

A Tabela 4.1 traz outros detalhes de cada base testada, como número total de instâncias, número de dimensões e número de classes.

Tabela 4.1: Bases de Dados Testadas

BASE	INSTÂNCIAS	DIMENSÕES	CLASSES
Artificial	400	2	3
Arcene	100	10000	2
Breast Cancer Wisconsin	569	32	2
Glass	214	10	6
Ionosphere	351	34	2
Iris	150	4	3
LSVT	126	309	2
Parkinsons	197	23	2
Pima Indians Diabetes	768	8	2
Sonar	208	60	2
Wine	178	13	3

4.2 PARÂMETROS DE TESTE

Com a finalidade de demonstrar a capacidade de generalização do MAP, foram realizados 30 experimentos com cada uma das bases citadas. Para cada um dos 30 experimentos, foram selecionadas diferentes amostras aleatórias, contendo 50% das instâncias, que eram as sementes do *Seeded k-means*. Em cada experimento, após o aprendizado do vetor de parâmetros W_{opt}, a nova métrica era testada nos outros 50% dos dados com a finalidade de realizar o agrupamento, juntamente com o *Seeded k-means*, que usava como centros de

inicialização os centroides das sementes usadas no aprendizado, e o *k-means* tradicional, com um centro inicial aleatório.

O valor de η foi definido em 0,01, assim como o valor de ϕ. O número máximo de iterações foi definido em 1000, tendo em vista os experimentos dispostos no Apêndice A, que demonstram saturação na melhoria da acurácia.

Todas as bases de dados tiveram cada uma de suas dimensões normalizadas em antes da realização dos experimentos, com a finalidade de melhorar o desempenho do *k-means*.

4.3 ANÁLISE DOS EXPERIMENTOS

Nos resultados dos experimentos realizados, foi possível perceber que o MAP apresentou melhoria significativa da acurácia do *k-means*, fato que era esperado, tendo em vista que MAP utiliza informações dos rótulos das bases de dados, disponibilizadas no conjunto de sementes, enquanto o *k-means* é um algoritmo não supervisionado. Todavia, os resultados de maior relevância foram os que o MAP *k-means* apresentou em relação ao *Seeded k-means*, que utiliza a mesma quantidade de informações preliminares.

A seguir será demonstrada uma breve descrição dos experimentos realizados com a base de dados gerada artificialmente e, posteriormente, os resultados obtidos com esta e as demais bases testadas.

4.4 BASE ARTIFICIAL

A base de dados artificial gerada para este experimento tem a finalidade de demonstrar a capacidade de aprendizado do vetor de parâmetros W e a possível necessidade de sua aplicação. Esta base foi gerada para que cada classe, mesmo após a normalização, apresentasse propositalmente formato elíptico. Na Figura 4.1 é possível perceber que o algoritmo *k-means* e o *Seeded k-means* têm tendência a realizar os agrupamentos em formato radial. O *Seeded k-means*, apesar de ter as mesmas informações prévias obtidas pelo MAP *k-means* por meio das sementes, essas informações não são utilizadas para caracterizar a base de dados.

É possível perceber na Figura 4.2, com o gráfico que os erros são significativamente minimizados com a utilização do MAP *k-means*, passando a ocorrer praticamente apenas nas fronteiras de cada classe.

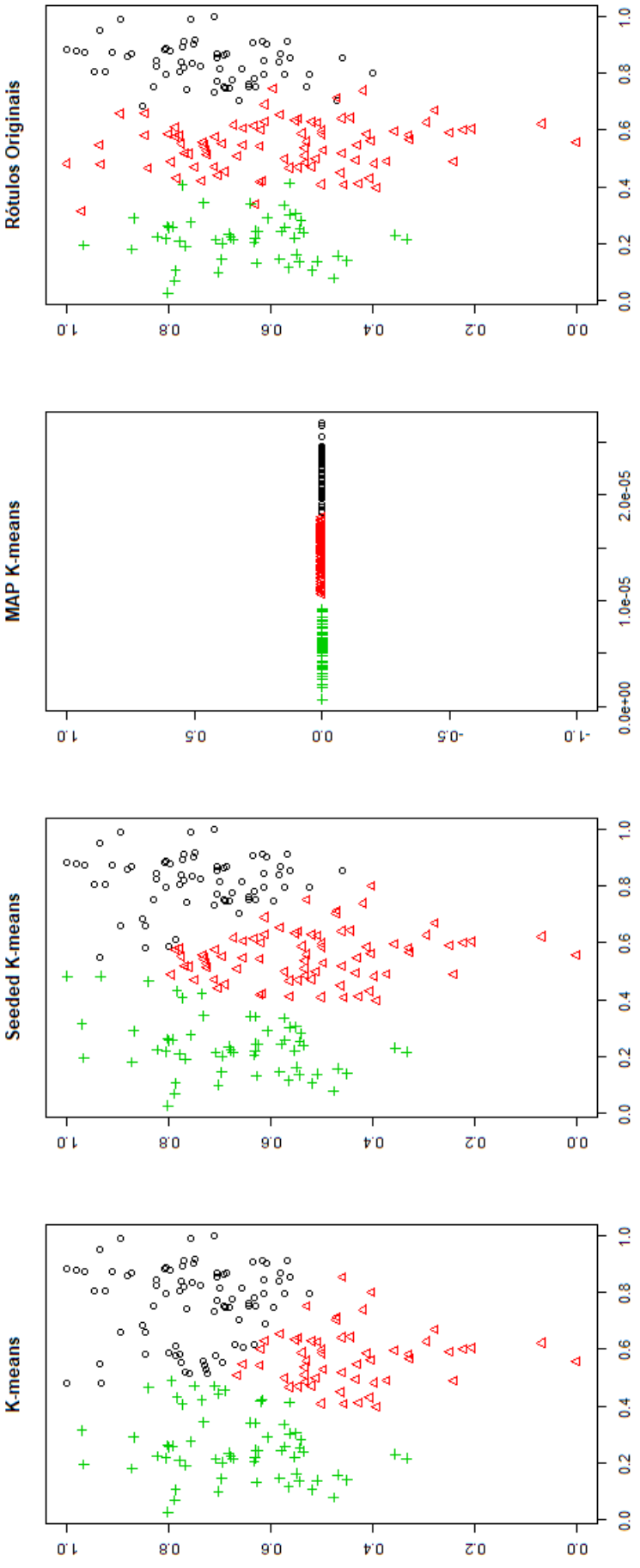
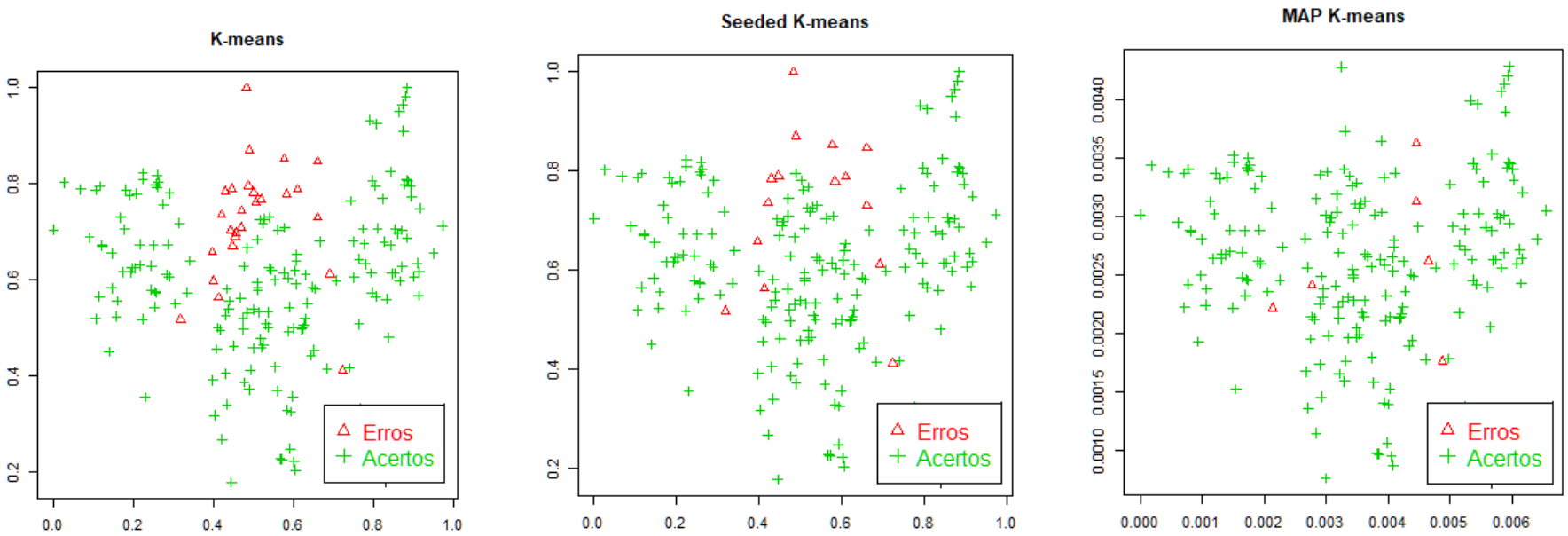


Figura 4.1: Comparação dos Resultados de Agrupamento



(a) (b) (c)

Figura 4.2: Comparação da Acurácia no Agrupamento

4.5 ANÁLISE DOS RESULTADOS

Nos experimentos realizados com as bases de dados da *UCI-Machine Learning*, considerando as médias de acurácia obtidas a partir dos 30 experimentos para cada base e os respectivos desvios padrões, demonstrados na Tabela 4.2, foi possível notar que o MAP contribuiu significativamente para melhoria do *Seeded k-means*, sobretudo nas seguintes bases: *Ionosphere*, *Iris*, *Pima Indians Diabetes* e *Sonar*.

Tendo em vista as interposições dos desvios padrões demonstrados na Tabela 4.2 entre o *Seeded k-means* e MAP *k-means*, foi realizado o teste T de diferença de médias entre esses dois resultados. Para duas bases de dados testadas (Arcene e Wine), não foi possível rejeitar a hipótese nula de que as médias sejam iguais. Todavia, para as demais 9 bases testadas, com 95% de confiança, foi possível rejeitar a hipótese de igualdade de médias entre o *Seeded k-means* e o MAP *k-means*. Ou seja, os resultados apresentam indícios de que o modelo de aprendizagem contribuiu para que as informações fossem classificadas com maior acurácia.

Mesmo considerando a heterogeneidade das bases de dados testadas, as métricas aprendidas pelo MAP foram equivalentes ou superiores à aplicação do *Seeded k-means*, o que reforça a hipótese de que o MAP, apesar de sua simplicidade, é um método promissor para aprendizado de métricas.

Tabela 4.2: Acurácia dos Algoritmos de Agrupamento por Base

BASE	K-MEANS	SEEDED K-MEANS	MAP K-MEANS	VALOR P (Seeded x MAP)
Artificial	0,797 ± 0,046	0,914 ± 0,024	0,958 ± 0,015	<0,05
Arcene	0,647 ± 0,049	0,675 ± 0,048	0,696 ± 0,060	0,13
Breast Cancer Wisconsin (Diagnostic)	0,911 ± 0,017	0,921 ± 0,012	0,937 ± 0,011	<0,05
Glass	0,414 ± 0,062	0,472 ± 0,041	0,539 ± 0,056	<0,05
Ionosphere	0,704 ± 0,024	0,716 ± 0,018	0,878 ± 0,026	<0,05
Iris	0,868 ± 0,069	0,911 ± 0,024	0,964 ± 0,016	<0,05
LSVT	0,590 ± 0,100	0,688 ± 0,062	0,746 ± 0,067	<0,05
Parkinsons	0,620 ± 0,083	0,736 ± 0,027	0,772 ± 0,031	<0,05
Pima Indians Diabetes	0,664 ± 0,021	0,707 ± 0,023	0,766 ± 0,016	<0,05
Sonar	0,548 ± 0,033	0,672 ± 0,049	0,788 ± 0,045	<0,05
Wine	0,942 ± 0,023	0,959 ± 0,022	0,966 ± 0,020	0,25

5 CONCLUSÕES E TRABALHOS FUTUROS

O Método de Aprendizado de Parâmetros (MAP) elaborado neste trabalho demonstrou nos experimentos resultados relevantes, tendo em vista sua menor complexidade em relação a todos os outros métodos conhecidos na literatura para aprendizado de métricas. Como vantagens, pode-se citar que é um método de aprendizado de métricas globais, originariamente multiclasse, escalável em relação ao número de instâncias e dimensões e que permite identificar possibilidades de redução de dimensionalidade. Nesse sentido, pode ser abordado em trabalhos futuros como um método para seleção de características, uma vez que o aprendizado dos parâmetros do vetor W_{opt} é, na verdade, uma atribuição de pesos para cada característica da base de dados.

Como pontos de atenção, destaca-se que as métricas aprendidas foram testadas em problemas de classificação usando o algoritmo k-NN (COVER; HART, 1967) e SVM (CORTES; VAPNIK, 1995), mas seu uso não proporcionou melhoras na classificação de dados, sendo necessárias investigações mais profundas sobre este insucesso.

Outro fato relevante que também deve ser destacado é a sensibilidade aos parâmetros de entrada, que foi amenizada pela inclusão da taxa de refinamento do aprendizado ϕ. Contudo, a taxa de refinamento deve ser usada com cautela, pois pode induzir à estagnação em ótimos locais, bem como ao aumento excessivo do passo de aprendizado η.

Como outros trabalhos futuros nessa linha de pesquisa, sugere-se a investigação mais profunda do efeito de variações nos parâmetros de entrada do MAP; a avaliação de utilização do MAP como um método de seleção de características; a avaliação da possibilidade de aplicação do método para problemas de classificação semissupervisionada e o uso de *kernel* para soluções de problemas não linearmente separáveis.

REFERÊNCIAS

BASU, S.; BANERJEE, A.; MOONEY, R. Semi-supervised clustering by seeding. In: CITESEER. **In Proceedings of 19th International Conference on Machine Learning (ICML-2002)**, 2002.

BELLET, A.; HABRARD, A.; SEBBAN, M. A survey on metric learning for feature vectors and structured data. **arXiv preprint arXiv:1306.6709**, 2013.

BEN, X.; MENG, W.; YAN, R.; WANG, K. An improved biometrics technique based on metric learning approach. **Neurocomputing**, Elsevier, v. 97, p. 44–51, 2012.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine learning**, Springer, v. 20, n. 3, p. 273–297, 1995.

COVER, T.; HART, P. Nearest neighbor pattern classification. **IEEE transactions on information theory**, IEEE, v. 13, n. 1, p. 21–27, 1967.

DAVIS, J. V.; KULIS, B.; JAIN, P.; SRA, S.; DHILLON, I. S. Information-theoretic metric learning. In: ACM. **Proceedings of the 24th international conference on Machine learning**, 2007. p. 209–216.

EDWARDS, A. W.; CAVALLI-SFORZA, L. L. A method for cluster analysis. **Biometrics**, JSTOR, p. 362–375, 1965.

FAGUNDES, F. L.; BORGES, C. C. H.; FONSECA NETO, R. Aprendizado de métrica utilizando uma função de distância parametrizada e o algoritmo k-means. In: **XIII Encontro Nacional de Inteligência Artificial e Computacional (XIII ENIAC)**, 2016.

HAMERLY, G.; ELKAN, C. Alternatives to the k-means algorithm that find better clusterings. In: ACM. **Proceedings of the eleventh international conference on Information and knowledge management**, 2002. p. 600–607.

HASTIE, T.; TIBSHIRANI, R. Discriminant adaptive nearest neighbor classification. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 18, n. 6, p. 607–616, 1996.

JAIN, P.; KULIS, B.; DHILLON, I. S.; GRAUMAN, K. Online metric learning and fast similarity search. In: **Advances in neural information processing systems**, 2009. p. 761–768.

KULIS, B. et al. Metric learning: A survey. **Foundations and Trends® in Machine Learning**, Now Publishers, Inc., v. 5, n. 4, p. 287–364, 2013.

LAJUGIE, R.; BACH, F.; ARLOT, S. Large-margin metric learning for constrained partitioning problems. In: **International Conference on Machine Learning**, 2014. p. 297–305.

LAW, M. T.; GUTIERREZ, C. S.; THOME, N.; GANÇARSKI, S. Structural and visual similarity learning for web page archiving. In: IEEE. **Content-Based Multimedia Indexing (CBMI), 2012 10th International Workshop on**, 2012. p. 1–6.

MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In: **Proceedings of the fifth Berkeley symposium on mathematical statistics and probability**, 1967. v. 1, p. 281–297.

MAHALANOBIS, P. C. On the generalised distance in statistics. **Proceedings of the National Institute of Sciences of India, 1936**, p. 49–55, 1936.

MCFEE, B.; BARRINGTON, L.; LANCKRIET, G. Learning content similarity for music recommendation. **IEEE transactions on audio, speech, and language processing**, IEEE, v. 20, n. 8, p. 2207–2218, 2012.

SCHULTZ, M.; JOACHIMS, T. Learning a distance metric from relative comparisons. In: **Advances in neural information processing systems**, 2004. p. 41–48.

SHAW, B.; HUANG, B.; JEBARA, T. Learning a distance metric from a network. In: **Advances in Neural Information Processing Systems**, 2011. p. 1899–1907.

SHORT, R.; FUKUNAGA, K. The optimal distance measure for nearest neighbor classification. **IEEE transactions on Information Theory**, IEEE, v. 27, n. 5, p. 622–627, 1981.

TAYLOR, M. E.; KULIS, B.; SHA, F. Metric learning for reinforcement learning agents. In: INTERNATIONAL FOUNDATION FOR AUTONOMOUS AGENTS AND MULTI-

AGENT SYSTEMS. **The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2**, 2011. p. 777–784.

WAGSTAFF, K.; CARDIE, C.; ROGERS, S.; SCHRÖDL, S. et al. Constrained k-means clustering with background knowledge. In: **ICML**, 2001. v. 1, p. 577–584.

WEINBERGER, K. Q.; BLITZER, J.; SAUL, L. K. Distance metric learning for large margin nearest neighbor classification. In: **Advances in neural information processing systems**, 2006. p. 1473–1480.

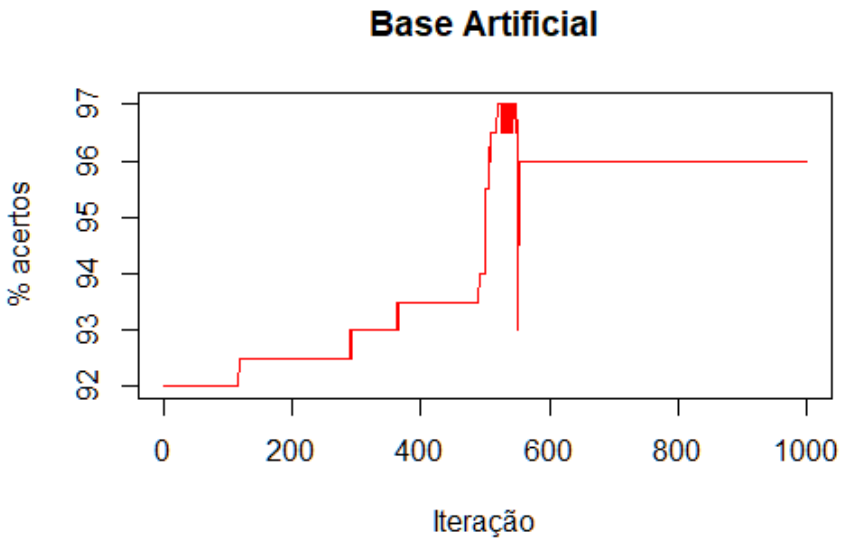
WEINBERGER, K. Q.; SAUL, L. K. Distance metric learning for large margin nearest neighbor classification. **Journal of Machine Learning Research**, v. 10, n. Feb, p. 207–244, 2009.

XING, E. P.; NG, A. Y.; JORDAN, M. I.; RUSSELL, S. Distance metric learning with application to clustering with side-information. In: **NIPS**, 2002. v. 15, n. 505-512, p. 12.

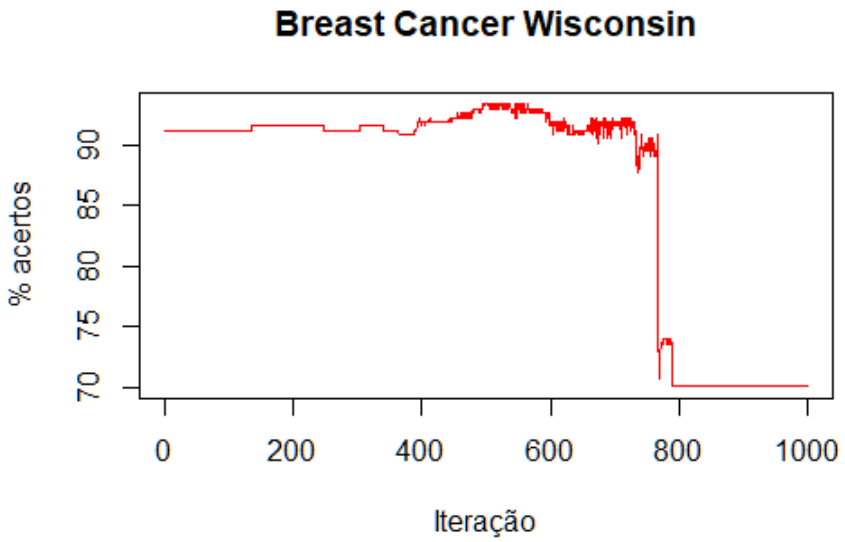
YANG, L.; JIN, R. Distance metric learning: A comprehensive survey. **Michigan State Universiy**, v. 2, n. 2, 2006.

YANG, P.; HUANG, K.; LIU, C.-L. Geometry preserving multi-task metric learning. **Machine learning**, Springer, v. 92, n. 1, p. 133–175, 2013.

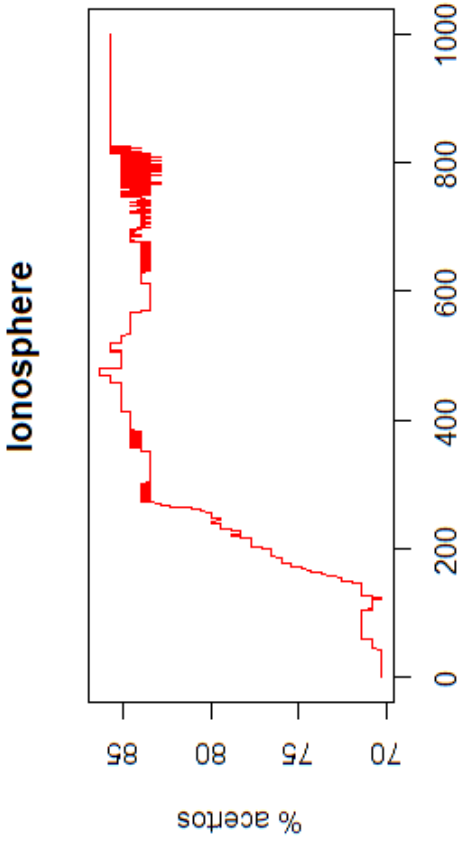
Apêndice A - ITERAÇÕES X ACURÁCIA



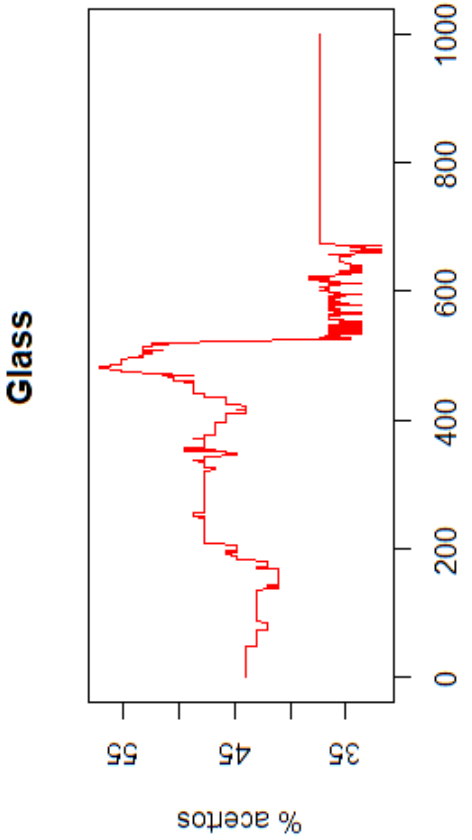
(a)



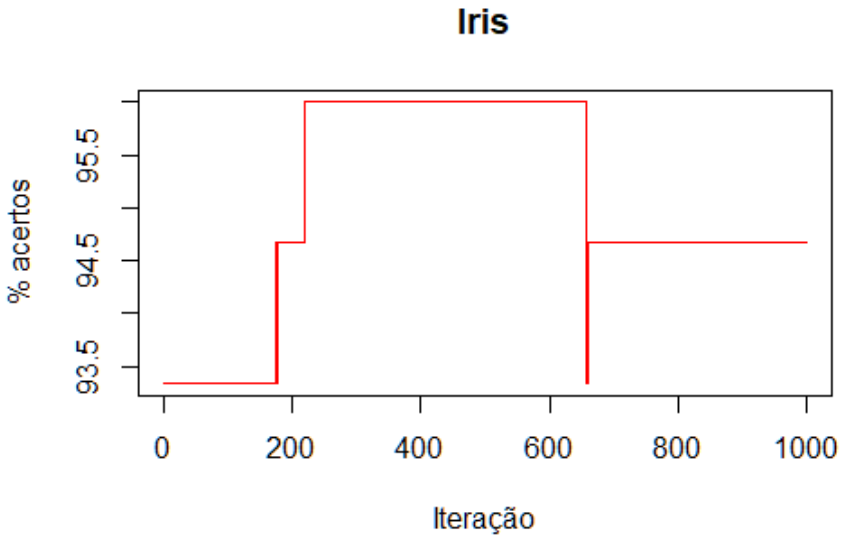
(b)


Ionosphere
% acertos
70
75
80
85
0
200
400
600
800
1000
Iteração

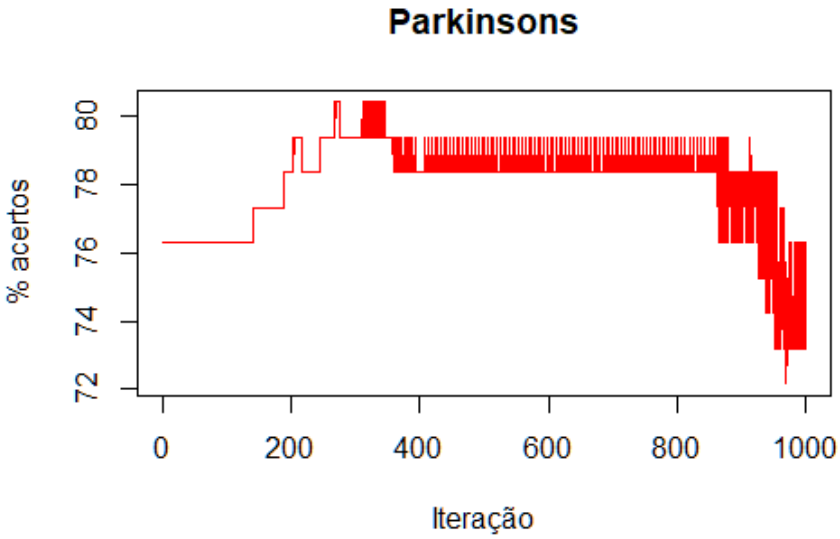
(d)


Glass
% acertos
35
45
55
0
200
400
600
800
1000
Iteração

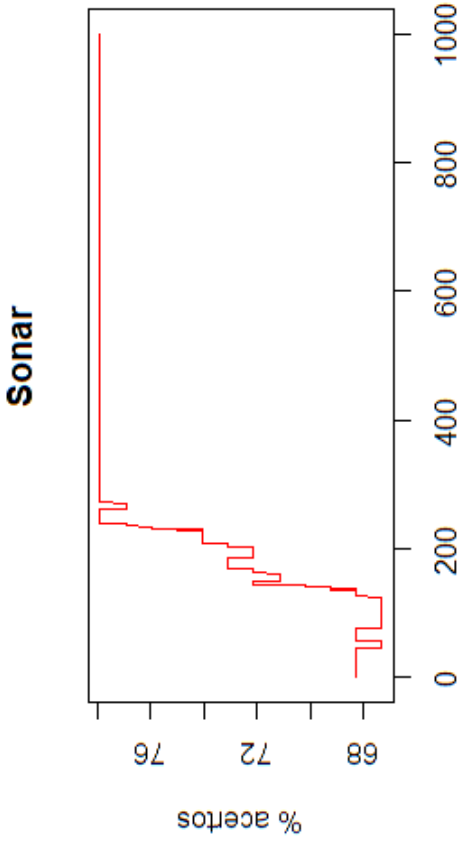
(c)


Iris
% acertos
93.5
94.5
95.5
0
200
400
600
800
1000
Iteração

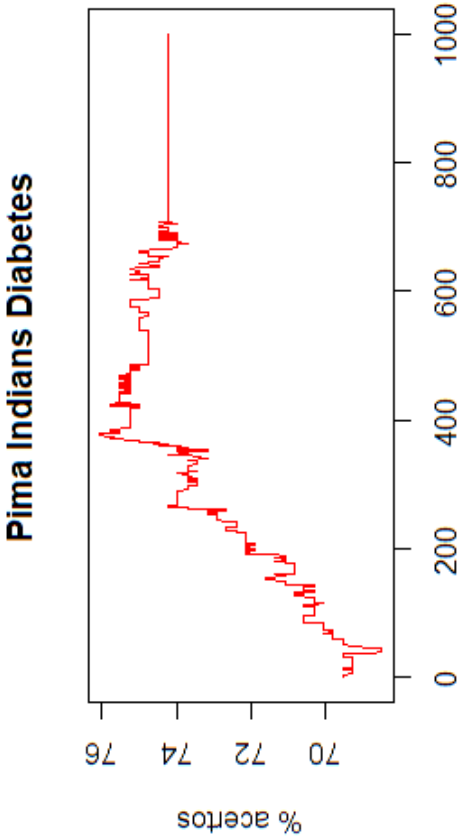
(e)


Parkinsons
% acertos
72
74
76
78
80
0
200
400
600
800
1000
Iteração

(f)


Sonar
% acertos
68
72
76
0
200
400
600
800
1000
Iteração

(h)


Pima Indians Diabetes
% acertos
70
72
74
76
0
200
400
600
800
1000
Iteração

(g)

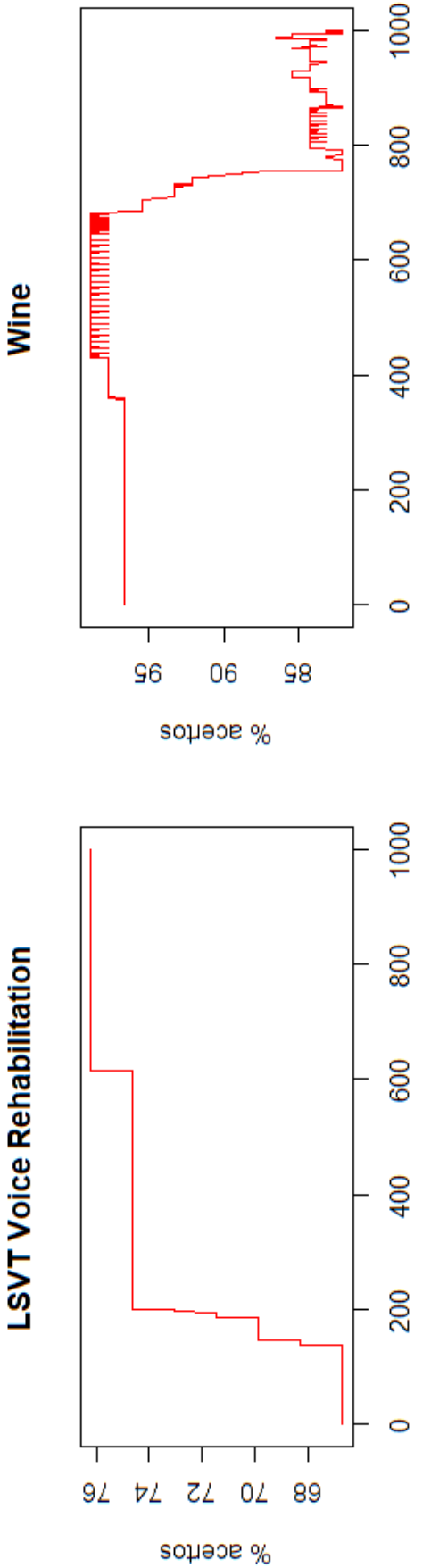


Figura A.1: Acurácia x Número de Iterações