Aprendizado de Métrica Supervisionado para Classificador por Arestas de Suporte

Igor Pereira Gomes[1⋆], Luiz Carlos Bambirra Torres[2⋆⋆], Antônio de Pádua Braga[3]

Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901
Belo Horizonte, MG, Brasil
[1] igor-gomes@ufmg.br,[2] luizlitc@gmail.com,[3] apbraga@ufmg.br

Resumo Como outros modelos baseados em informações de margem, os Classificadores por Arestas de Suporte (CLAS) utilizam propriedades intrínsecas dos Grafos de Gabriel para filtrar amostras na região de separação visando a suavizar a resposta do classificador. A abordagem utilizada está sujeita à distribuição das amostras na região de superposição e pode depender do conjunto de dados. Como alternativa, propõe-se, neste trabalho, uma etapa de aprendizado de métrica supervisionado baseado no método *Large Margin Nearest Neighbors* no treinamento de modelos CLAS com o efeito de reduzir a sobreposição entre as classes e controlar também a suavização do modelo.

1 Introdução

Os modelos da família CLAS (Classificadores por Arestas de Suporte) [1] são baseados na estrutura do conjunto de dados e necessitam de pouca intervenção do usuário para inicialização de parâmetros de treinamento [2]. Estes modelos evitam os custosos processos de ajuste de parâmetros através de busca em *grid* e validação cruzada, utilizados frequentemente para o ajuste de modelos tais como as Máquinas de Vetor de Suporte (SVM). Resultados apresentados para um dos modelos da família CLAS utilizando um *benchmark* de 17 bases de dados reais mostraram desempenho estatisticamente equivalente ao das SVMs com *kernels* RBF e Polinomial [3]. O classificador CHIP-CLAS apresentou, inclusive, melhor *rank* médio para o teste estatístico de Bonferroni-Dunn [4].

CLAS e SVM baseiam-se na margem de separação, ou seja, na distância entre a superfície de separação e as amostras do conjunto de dados. Como a margem é uma região de maior incerteza, o estabelecimento de uma condição inflexível de margem larga ou margem máxima pode levar a um modelo rígido e com menor desempenho na separação das funções geradoras [5] (Figura 1). No treinamento de SVMs, isto é feito com a relaxação da condição de margem máxima com uma

⋆ O presente trabalho foi realizado com o apoio financeiro da CAPES - Brasil.
⋆⋆ Bolsista do CNPq–Brasil (N°150254/2016-4).

variável de folga, exigindo então o ajuste do parâmetro de Custo (C) [6]. Nos Classificadores CLAS, desconsidera-se no processo de treinamento os dados nas regiões de sobreposição [3], considerando-se propriedades intrínsecas das arestas do Grafo de Gabriel na região de separação. Apesar de esta estratégia para o CLAS ter resultado em modelos com desempenho equivalente às SVMs, os efeitos desta estratégia ainda são objeto de estudo.

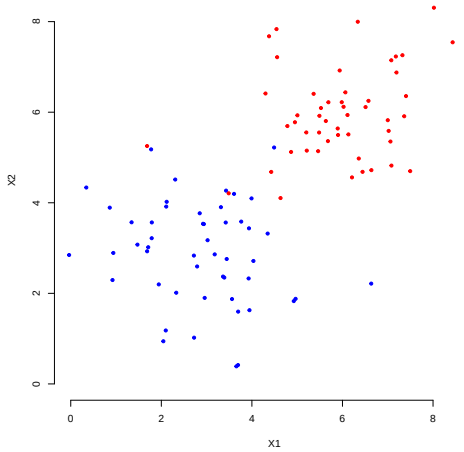

Figura 1: Conjunto de dados com sobreposição.

Propõe-se neste trabalho uma forma de reduzir a quantidade de dados desconsiderada no treinamento do classificador CHIP-Clas, através de aprendizado de métrica supervisionado. No aprendizado de métrica, uma métrica de distância é parametrizada com base nos dados de treinamento de forma a aumentar a margem de separação entre as classes. Utiliza-se um método para aprendizado de métrica baseado no algoritmo Large Margin Nearest Neighbours (LMNN) [7], obtendo-se os parâmetros para uma Métrica de Mahalanobis que cumpra este objetivo.

Os resultados obtidos nos experimentos realizados indicam desempenho estatisticamente equivalente ao da abordagem original do CHIP-Clas, com maior utilização do conjunto de dados.

O artigo é organizado como descrito a seguir. A Seção 2 contém a fundamentação teórica, apresentando os métodos CHIP-Clas e LMNN. A Seção 3 apresenta a abordagem de aprendizado de métrica proposta por este artigo. A Seção 4 descreve os experimentos realizados e mostra os resultados obtidos, os avaliando através de métodos estatísticos. Por último, os resultados são discutidos na conclusão, na Seção 5.

2 Fundamentação Teórica

2.1 Classificador por Arestas de Suporte

Os Classificadores por Arestas de Suporte [3] constituem uma família de algoritmos de classificação de margem larga com métodos de aprendizado baseados em Grafos de Gabriel [8]. Os Grafos de Gabriel são grafos não-orientados onde dois pontos são interconectados se e somente se não existe um terceiro ponto no interior da hiperesfera cujo diâmetro é definido por estes dois pontos. Nos classificadores CLAS, é construído o Grafo de Gabriel correspondente ao conjunto de dados e são então definidas as Arestas de Suporte, que são arestas que separam pontos de classes distintas. Através delas e de seus pontos médios, são extraídos parâmetros para configuração e construção de classificadores de margem larga [9] [10] [11], além de um decisor [1] utilizado para o método de treinamento multiobjetivo de redes neurais [12].

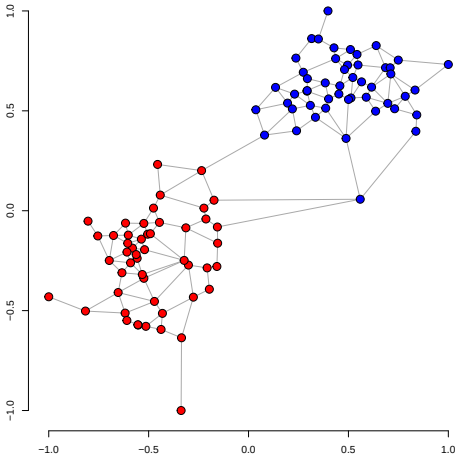

Figura 2: Grafo de Gabriel de um conjunto de dados amostrados de duas gaussianas.

Como base para este trabalho, utiliza-se o classificador CHIP-CLAS [11]. Este cria para cada aresta de suporte um hiperplano de separação que passa pelo ponto médio da mesma e maximiza a margem de separação. A classificação é feita através de votação deste conjunto de hiperplanos. O voto de cada hiperplano é ponderado pela distância dos pontos médios das arestas de suporte ao ponto a ser classificado.

Sobreposição Como ocorre nos classificadores baseados em margem, a família CLAS perde desempenho de generalização para bases de dados com sobreposição. O classificador deve possuir, portanto, um meio de flexibilizar a condição de margem larga para controlar o erro de generalização. Nas SVMs, este problema é contornado com a adição de uma variável de folga para a restrição de margem máxima na formulação do problema de otimização [6]. Para o CLAS, este problema é solucionado excluindo-se os pontos do conjunto de dados que caracterizam a sobreposição. Isso é feito atribuindo-se a cada ponto um fator de qualidade dado pela razão entre o número de pontos de mesma classe conectados ao nó correspondente e o número total de pontos conectados a ele. Exclui-se os pontos cujo fator de qualidade é menor que a média do fator de todos os pontos.

O método CHIP-CLAS, em particular, procura identificar se existe sobreposição nos dados antes que o processo de eliminação seja iniciado. Não há, porém, controle sobre a quantidade de dados excluída no processo, podendo a exclusão demasiada de dados levar à perda de desempenho do classificador.

Distância de Mahalanobis

A distância de Mahalanobis [13] foi inicialmente criada como uma medida de distância entre um ponto e uma distribuição de probabilidade multivariada. Ela é definida pela Equação 1, onde $\mathbf{X}$ é o vetor que indica a localização do ponto, $\mathbf{Y}$ é a média e M é o inverso da matriz covariância da distribuição de probabilidade.

$$D_M(\mathbf{X}, \mathbf{Y}) = \sqrt{(\mathbf{X} - \mathbf{Y})^T M (\mathbf{X} - \mathbf{Y})} \tag{1}$$

Pode-se generalizar este conceito e considerar a distância de Mahalanobis entre dois pontos $\mathbf{X}$ e $\mathbf{Y}$ quaisquer, sendo a matriz M um parâmetro para a métrica. Chamamos esta de matriz de Mahalanobis, com a restrição que M deve ser uma matriz semidefinida positiva d por d, onde d é a dimensão do espaço em que os pontos estão definidos.

A distância de Mahalanobis pode ser considerada uma generalização da distância Euclidiana, sendo esta última equivalente à distância de Mahalanobis com a matriz M igual à identidade. O conjunto dos pontos equidistantes a um centro utilizando distância de Mahalanobis gera uma superfície elipsoidal, enquanto para a distância Euclidiana, esta superfície é circular.

Large Margin Nearest Neighbors (LMNN)

A maioria dos métodos baseados em distâncias, como o SVM, o KNN e o próprio CLAS, foram descritos utilizando-se a distância Euclidiana. Para alguns problemas a distância Euclidiana entre alguns pontos de mesma classe pode ser maior que a distância entre pontos de classes distintas. Para solucionar este problema, pode-se usar métricas parametrizadas de distância, sendo os melhores parâmetros para cada problema obtidos através de um processo de otimização. O LMNN [7] é um processo criado para aprendizado de métrica para classificadores KNN.

A melhor matriz de Mahalanobis é encontrada através da minimização de uma função convexa baseada no erro Leave-One-Out (LOO) deste classificador.

O método recebe o número de vizinhos mais próximos do KNN como parâmetro (K). Pode-se definir a função objetivo para o LMNN como composta de dois termos. O primeiro penaliza a soma das distâncias de cada ponto a seus vizinhos mais próximos, tendo efeito de aproximá-los, sendo dado pela Equação 2, onde $j \rightsquigarrow i$ significa que j está entre os K vizinhos mais próximos de i.

$$\varepsilon_{pull}(M) = \sum_{j \rightsquigarrow i} D_M^2(\mathbf{x}_i, \mathbf{x}_j) \tag{2}$$

O segundo termo penaliza curtas distâncias entre cada ponto e pontos de classes distintas entre seus vizinhos mais próximos (impostores). É definido pela Equação 3.

$$\varepsilon_{push}(M) = \sum_{i,j \rightsquigarrow i} \sum_{l} (1 - y_{il})[1 + D_M^2(\mathbf{x}_i, \mathbf{x}_j) - D_M^2(\mathbf{x}_i, \mathbf{x}_l)] \tag{3}$$

Para implementação direta no CLAS, que trabalha com distância Euclidiana, a matriz M pode ser decomposta como o quadrado de uma matriz simétrica. A distância de Mahalanobis pode então ser obtida através da distância Euclidiana das transformações lineares dos pontos por esta matriz simétrica, como visto na Equação 5.

$$M = LL \tag{4}$$

$$D_M = dist(L\mathbf{X}, L\mathbf{Y}) \tag{5}$$

Assim, a classificação utilizando a distância Euclidiana, utilizando-se esta transformação linear L nos dados de entrada, é equivalente à utilização da distância de Mahalanobis. Somando-se as penalidades, adicionando a restrição Semidefinida-Positiva para a matriz M e modificando o segundo termo da função objetivo de forma a adicionar variáveis de folga e colocá-la numa forma mais adequada para a solução, temos a formulação final do problema de otimização:

$$\begin{aligned} L* = \arg\min_{L} \quad & \sum_{j \rightsquigarrow i} d(L\mathbf{x}_i, L\mathbf{x}_j) + \sum_{i,j \rightsquigarrow i} (1 - y_{il})\xi_{ijl} \\ \text{sujeito a} \quad & d(L\mathbf{x}_i, L\mathbf{x}_l) - d(L\mathbf{x}_i, L\mathbf{x}_j) \geq 1 - \xi_{ijl}, \\ & \xi_{ijl} \geq 0, \\ & LL \succeq 0. \end{aligned} \tag{6}$$

Após o aprendizado de métricas, o algoritmo LMNN toma a decisão utilizando o classificador KNN com a métrica de distância aprendida. Assim, cada amostra do conjunto de testes é classificada de acordo com seus K vizinhos mais próximos segundo a métrica de Mahalanobis obtida.

3 Metodologia

O aprendizado de métrica do método LMNN foi adaptado neste trabalho para utilização em classificadores CLAS. Espera-se que o processo não possua hiperparâmetros, de forma que continue sendo desnecessário o ajuste de parâmetros através de validação cruzada e busca em *grid*. Para isso, modifica-se a função objetivo do LMNN. Ao invés de considerar os K vizinhos mais próximos para cada ponto, a função é calculada considerando-se os vizinhos conectados a ele em um Grafo de Gabriel construído utilizando distância Euclidiana. Elimina-se assim a necessidade de um parâmetro K e leva-se em conta no aprendizado de métrica a estrutura geométrica do problema, também utilizada na classificação.

A formulação do problema de otimização mantém-se na forma vista na Equação 6, com $j \rightsquigarrow i$ significando que j é conectado a i no Grafo de Gabriel. Isto mantém as características de convexidade e de restrições esparsamente violadas do problema de otimização do LMNN.

Obtida a matriz L, são feitas as transformações lineares nos conjuntos de treino e teste e o problema de classificação é solucionado pelo algoritmo CHIP-CLAS.

4 Experimentos e Resultados

A nova abordagem para o CHIP-CLAS com Aprendizado de Métrica (chamada nos resultados de **AM-CHIP-CLAS**) foi avaliada através de *10-fold Cross Validation* [14]. Mediu-se a porcentagem dos dados desconsiderados no treinamento e o desempenho da classificação através de AUC (área sob a curva ROC). Os experimentos foram realizados com 13 bases de dados reais obtidas através do repositório UCI [15] e 2 problemas de expressão gênica: *Golub* [16] e *BcrHess* [17].

A porcentagem desconsiderada dos dados foi comparada com a obtida para o algoritmo CHIP-CLAS em sua abordagem original, sem aprendizado de métrica. O desempenho foi comparado com o algoritmo CHIP-CLAS sem aprendizado de métrica e com o classificador SVM com *Kernels* RBF e Polinomial. Os melhores parâmetros para o SVM foram encontrados através de *10-fold Cross Validation* e busca em *grid*.

Ao fim, buscou-se visualizar os efeitos do aprendizado de métrica na superfície de separação.

4.1 Utilização dos Dados

A porcentagem dos dados desconsiderados no treinamento para cada execução da validação cruzada foi medida. Calculou-se a razão entre a quantidade de amostras descartadas no processo de filtragem e o total de dados da base, com e sem aprendizado de métrica. Os resultados médios obtidos para as execuções se encontram na Tabela 1.

Tabela 1: Porcentagem média desconsiderada dos dados.

dataset	CHIP-CLAS	AM-CHIP-CLAS
1 sonar	0.00	0.00
2 breastcancer	15.32	10.85
3 australian	37.97	38.89
4 diabetes	44.68	44.73
5 breastHess	38.94	26.16
6 bupa	48.28	48.76
7 haberman	45.97	45.24
8 banknote	0.00	0.00
9 fertility	43.11	24.78
10 parkinsons	10.36	3.24
11 climate	39.55	22.55
12 ILPD	47.55	47.27
13 german	46.86	47.30
14 heart	43.25	43.66
15 golub	37.05	0.00

Para 6 das 15 bases testadas, a porcentagem desconsiderada dos dados diminuiu consideravelmente, sofrendo variação de menos de 1% para cima ou para baixo nas bases restantes. Isto sugere uma maior utilização dos dados para o AM-CHIP-CLAS. A significância estatística desta superioridade pode ser estabelecida através de um teste estatístico de Wilcoxon pareado [4]. O teste unilateral foi utilizado, com nível de confiança de 95% ($\alpha = 0.05$). O Valor-p obtido no teste foi $p = 0.040$, de forma que $p < \alpha$, confirmando estatisticamente a maior utilização dos dados para a nova abordagem com 95% de confiança.

4.2 Desempenho

A AUC para cada execução da validação cruzada foi medida e foi extraída a média para cada base de dados. Os resultados obtidos se encontram na Tabela 2, juntamente com a média da posição de cada classificador num *ranking* de desempenho para cada base de dados.

Para avaliação estatística dos resultados de múltiplos classificadores, é indicado o teste de Friedman [4]. Para um nível de confiança de 95% ($\alpha = 0.05$), foi obtido um Valor-p de $p = 0.445$. O resultado obtido não é suficiente para rejeitar a hipótese nula de que nenhum dos classificadores possui desempenho estatisticamente diferente dos demais. Para melhor visualizar o desempenho dos classificadores, foi feito o teste *post-hoc* de Bonferoni-Dunn [4], obtendo-se o gráfico da Figura 3, com o eixo horizontal indicando o *rank* (quanto menor, melhor o desempenho).

Verifica-se que o desempenho da abordagem AM-CHIP-CLAS não difere significativamente do classificador CHIP-CLAS sem aprendizado de métrica, com *rank* médio pouco superior a este. Ambos CHIP-CLAS e AM-CHIP-CLAS se

Tabela 2: AUC Média das execuções e Rank Médio dos Classificadores.

dataset	AM-CHIP-CLAS	CHIP-CLAS	RBF-SVM	Poly-SVM
sonar	0.84	0.88	0.84	0.87
breastcancer	0.97	0.96	0.97	0.96
australian	0.86	0.85	0.86	0.87
diabetes	0.71	0.72	0.71	0.71
breastHess	0.83	0.81	0.76	0.77
bupa	0.58	0.61	0.67	0.72
haberman	0.54	0.56	0.52	0.50
banknote	1.00	0.99	1.00	1.00
fertility	0.50	0.59	0.50	0.50
parkinsons	0.89	0.90	0.77	0.81
climate	0.85	0.84	0.53	0.72
ILPD	0.57	0.57	0.49	0.50
german	0.70	0.67	0.66	0.68
heart	0.81	0.80	0.83	0.83
golub	0.55	0.77	0.80	0.78
Rank Mean	2.20	2.40	2.93	2.47

mostram também superiores no *rank* médio aos classificadores SVM testados, para o *benchmark* utilizado.

4.3 Visualização da Superfície de Separação

Para visualização do efeito do aprendizado de métrica, o método AM-CHIP-CLAS e o CHIP-CLAS original foram utilizados para separação de um conjunto de dados sintético de duas dimensões. Foi criado para tal um conjunto de dados consistindo em fileiras intercaladas de classes distintas alinhadas com o eixo X, adicionadas de ruído gaussiano em ambas dimensões. Desta forma, algumas amostras significativas para o treinamento ficam próximas de mais pontos da classe oposta que as demais. Assim, induz-se ao erro o método para eliminação de sobreposição do CHIP-CLAS original, destacando assim a diferença entre ambas as metodologias.

O método CHIP-CLAS original desconsiderou 41.67% dos dados no processo de classificação, gerando a superfície de separação da Figura 4. O método AM-CHIP-CLAS não desconsiderou nenhuma amostra no processo de classificação, gerando a superfície da Figura 5.

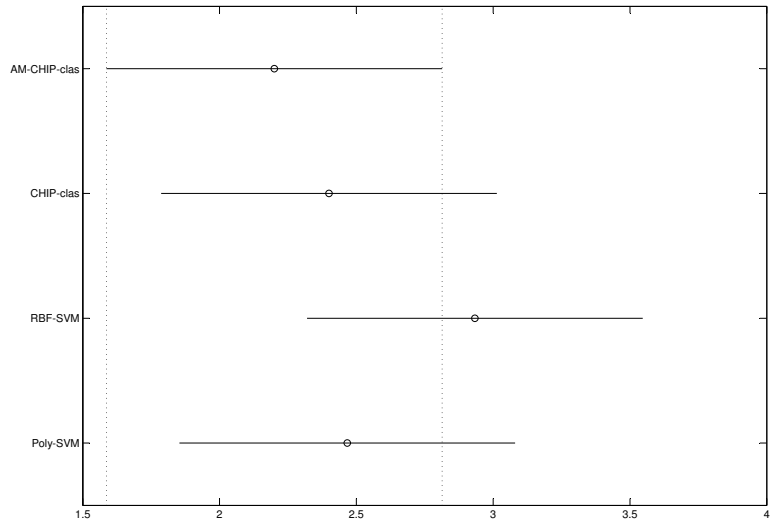

Figura 3: Visualização para o teste *post-hoc* de Bonferroni-Dunn

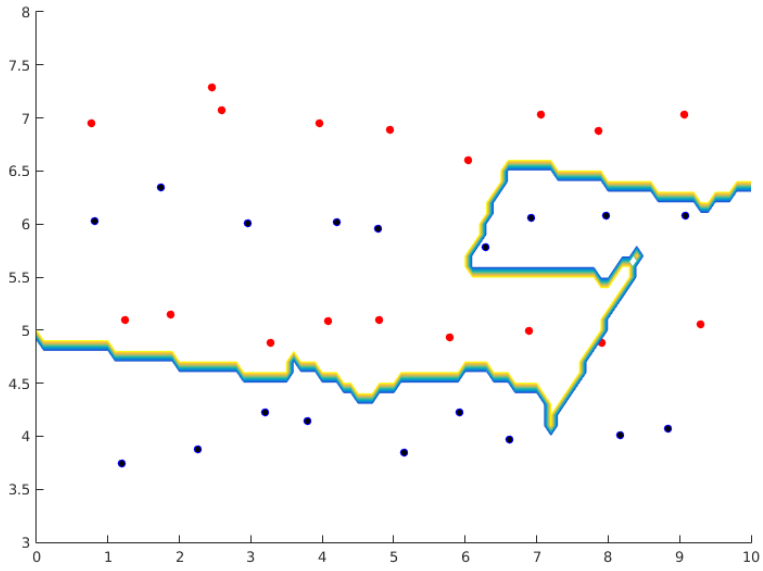

Figura 4: Visualização da superfície de separação para o CHIP-CLAS original

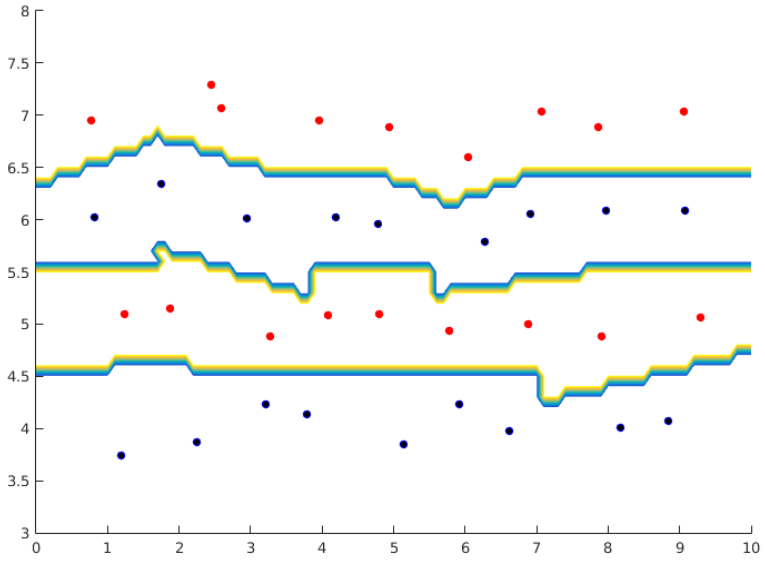

Figura 5: Visualização da superfície de separação para o AM-CHIP-CLAS

5 Conclusões e Discussões

Este trabalho apresentou uma etapa de aprendizado de métrica supervisionado para o classificador CHIP-CLAS baseada no método Large Margin Nearest Neighbors e na estrutura de Grafos de Gabriel. Pôde-se concluir a partir dos testes realizados que o aprendizado de métrica teve sucesso em reduzir a quantidade de dados desconsiderados no processo de eliminação de sobreposição para a classificação.

O desempenho obtido para o novo método com o *benchmark* testado foi estatisticamente equivalente ao do classificador CHIP-CLAS original e ao SVM com *kernel* RBF (RBF-SVM) e polinomial (Poly-SVM). O *rank* médio encontrado foi superior a todos os outros classificadores testados, incluindo o CHIP-CLAS em sua abordagem original. Dada a maior utilização do conjunto de dados e o desempenho apresentado, justifica-se a utilização do novo método na classificação de conjuntos de dados com reduzido número de observações, situação em que a sub-utilização dos dados resulta em menor desempenho. Novos testes devem ser realizados utilizando como *benchmark* conjuntos de dados com estas características. Mais testes também devem ser realizados para análise do custo computacional de ambas as abordagens, CHIP-CLAS e AM-CHIP-CLAS, visto que não existem na literatura estudos detalhados e quantitativos sobre tais custos.

Por último, pode-se destacar a robustez da utilização da estrutura de Grafos de Gabriel para se realizar o aprendizado de métrica supervisionado sem necessidade de ajustar hiperparâmetros, sugerindo a utilização do mesmo para o método LMNN.

Referências

1. Torres, L., Castro, C., Braga, A.: A Computational Geometry Approach for Pareto-Optimal Selection of Neural Networks. International Conference on Artificial Neural Networks (22) (2012)
2. Torres, L., Castro, C., Braga, A.: A Parameterless Mixture Model for Large Margin Classification. International Joint Conference on Neural Networks (2015)
3. Torres, L.C.B.: Classificador por Arestas de Suporte (CLAS): Métodos de Aprendizado Baseados em Grafos de Gabriel. PhD thesis, Universidade Federal de Minas Gerais (2016)
4. Demšar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research **7** (2006) 1–30
5. Vapnik, V.N.: The Nature of Statistical Learning Theory. Volume 8. (2000)
6. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning **20**(3) (1995) 273–297
7. Weinberger, K.Q., Saul, L.K.: Distance Metric Learning for Large Margin Nearest Neighbor Classification. The Journal of Machine Learning Research **10** (2009) 207–244
8. Gabriel, K.R., Sokal, R.R.: A New Statistical Approach to Geographic Variation Analysis. Systematic Zoology **18**(3) (1969) 259–278

9. Torres, L., Lemos, A., Castro, C., Braga, A.: A geometrical approach for parameter selection of radial basis functions networks. In: Lecture Notes in Computer Science. Volume 8681 LNCS. (2014)
10. Torres, L., Castro, C., Braga, A.: Gabriel Graph for Dataset Structure and Large Margin Classification: A Bayesian Approach. Proceedings of the European Symposium on Neural Networks 2015 (2015) 237–242
11. Torres, L., Castro, C., Coelho, F., Sill Torres, F., Braga, A.: Distance-based large margin classifier suitable for integrated circuit implementation. Electronics Letters **51**(24) (2015) 1967–1969
12. de Albuquerque Teixeira, R., Braga, A.P., Takahashi, R.H.C., Saldanha, R.R.: Improving generalization of MLPs with multi-objective optimization. Neurocomputing **35** (2000) 189–194
13. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. (2000)
14. Kohavi, R.: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Proceedings of the 14th International Joint Conference on Articial Intelligence (IJCAI) (1995) 1137–1145
15. Bache, K., Lichman, M.: UCI Machine Learning Repository (2013)
16. Golub, T.R.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science **286**(5439) (1999) 531–537
17. Hess, K.R., Anderson, K., Symmans, W.F., Valero, V., Ibrahim, N., Mejia, J.A., Booser, D., Theriault, R.L., Buzdar, A.U., Dempsey, P.J., Rouzier, R., Sneige, N., Ross, J.S., Vidaurre, T., Gomez, H.L., Hortobagyi, G.N., Pusztai, L.: Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. Journal of Clinical Oncology **24**(26) (2006) 4236–4244