

Avaliação de Classificadores utilizando Técnicas de Estimativa de Densidades

Victor São Paulo Ruela
Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
Email: victorspruela@ufmg.br

Resumo—A modelagem de dados não-lineares com redes neurais artificiais depende da qualidade projeção aplicada sobre as entradas, geralmente feita através de funções de kernel. A otimização de seus parâmetros é uma etapa importante e pode ser feita via técnicas de estimativa de densidade. Além de fornecer uma forma automática de seleção dos parâmetros ótimos, estas técnicas possuem a tendência de gerar projeções ortogonais das entradas no espaço de projetado. A partir desta observação, este trabalho tem como objetivo avaliar o desempenho de classificadores lineares sobre esta projeção, considerando problemas de benchmark presentes na literatura.

I. INTRODUÇÃO

O desempenho de redes neurais artificiais sobre dados não-lineares é altamente dependente da projeção aplicada sobre as entradas, geralmente feita através de kernels. Estas funções possuem diversos parâmetros a serem ajustados, os quais irão afetar diretamente a qualidade do modelo obtido. O ajuste de seus parâmetros é geralmente realizado via técnicas de busca exaustiva, como validação cruzada [1]. Embora amplamente utilizadas, tais técnicas não utilizam informações presentes nos dados. Isso motivou o desenvolvimento de, por exemplo, técnicas baseadas em estimativa de densidade para analisar a estrutura dos dados e reduzir a necessidade de interação do usuário [2].

Baseado no KDE (*Kernel Density Estimation*) [3], esta abordagem explora o comportamento da projeção das funções de similaridade calculadas sobre kernel escolhido. É determinada uma função sobre os parâmetros do kernel, a qual pode ser utilizada para determinar os parâmetros que melhor se ajustam aos dados. É possível, por exemplo, usar este conceito para a determinação da largura ótima de kernels radiais (RBF) para o algoritmo SVM [2]. Um resultado importante desta técnica está no fato dela poder gerar projeções ortogonais no espaço de verossimilhanças. Isso sugere a possibilidade de utilizar modelos lineares sobre esta projeção, como o Perceptron e aprendizado Hebbiano.

Este trabalho tem como objetivo avaliar o desempenho dos modelos Perceptron simples [4] e Hebbiano [5] sobre as projeções no espaço de verossimilhanças. Serão considerados tanto os kernels gaussiano e perceptron de múltiplas camadas (MLP), disponibilizados pelo professor. Além destes modelos, serão considerados também o ELM com regularização [6], SVM [7], [2] e RBF com otimização de largura, aplicados sobre o espaço das entradas. Este último modelo não possui

publicação associada e é proposto como forma de extensão ao enunciado original do trabalho. Um experimento será desenhado para compará-los estatisticamente sobre diferentes bases de dados de *benchmark* disponíveis na literatura.

II. REVISÃO DA LITERATURA

A. Métodos baseados em Kernel

III. METODOLOGIA

A. Desenho do experimento

A partir das recomendações para desenho de experimento para comparação de algoritmos proposta em [9], a seguinte metodologia será adotada:

- Para cada base de dados:
 - 1) Particionar os dados D em k partições para validação cruzada, mantendo a mesma proporção entre os rótulos
 - a) Criar o conjunto de treino $T = D - k$
 - b) Para cada modelo:
 - i) Executar busca exaustiva com validação cruzada de z partições sobre T para os coeficientes de regularização λ
 - ii) Escolher λ que obtém o melhor ajuste médio
 - iii) Avaliar a métrica do modelo sobre k
 - 2) Estimar o intervalo de confiança de 95% do valor médio da métrica sobre k usando *bootstrapping*

O número de partições consideradas será de $k = 10$. Para o item 1(b)i, serão considerados um número fixo de valores igualmente espaçados dentro de um intervalo pré-definido. Além disso, serão considerados $z = 5$ partições, como forma de controlar um pouco o tamanho do experimento. Serão consideradas as acurácia como a métrica para comparação dos algoritmos. A implementação deste experimento, bem como dos modelos utilizados, será feita em Python e utilizando principalmente os pacotes *numpy* [10] e *scikit-learn* [11]. O experimento será realizado em um Notebook Intel Core i7 Quad Core com 8Gb de memória RAM, sendo que o uso de paralelização será utilizado sempre que possível visto a enorme quantidade de vezes que os modelos serão treinados.

B. Aprendizagem Hebbiana

O uso de aprendizado Hebbiano após a camada escondida é uma abordagem simples e eficiente para se controlar a generalização do modelo. Conforme sugerido em [12], podemos substituir o cálculo da pseudo-inversa da matriz de projeção, o que caracteriza a regra de aprendizado do Perceptron simples, por um Perceptron Hebbiano com pesos normalizados. Essa abordagem é melhor descrita em [5], da qual podemos retirar a seguinte regra para problemas de classificação binários:

$$w = \frac{\sum_{i=1}^N y_i \mathbf{h}_i}{\left\| \sum_{i=1}^N y_i \mathbf{h}_i \right\|} \quad (1)$$

onde \mathbf{h}_i é a i -ésima linha da matriz de projeção. É importante ressaltar que esta regra assume que os dados foram normalizados para possuir média zero e desvio padrão unitário. Conforme os resultados obtidos no anterior da disciplina, o uso do aprendizado Hebbiano é capaz de atingir a regularização desejada, porém seu desempenho é limitado a projeções que tornam os dados linearmente separáveis. Logo, podemos concluir que esta abordagem poderá ter desempenho satisfatório caso a projeção no espaço de verossimilhanças seja adequada.

C. RBF com Otimização de Largura

As redes RBF foram inicialmente introduzidas por [13] e são caracterizadas por um aprendizado que envolve duas etapas: (i) aplicar uma transformação aos padrões através de um kernel Gaussiano (ii) encontrar os pesos usando o estimador mínimos quadrados usado no Perceptron simples. Portanto, a qualidade do modelo obtido será diretamente afetado pela escolha dos centróides e larguras das funções radiais de cada neurônio.

Esta definição é geralmente feita utilizando o algoritmo *k-means* [14], porém podemos facilmente ver que a mesma abordagem de otimização da largura de SVMs com kernel RBF proposta por [2] pode ser aplicada. Se considerarmos que cada neurônio usará a largura ótima obtida pela técnica anterior, precisamos definir somente os centróides na sequência, os quais serão amostrados uniformemente no espaço de entradas, resultando em um menor esforço computacional e complexidade. Na etapa final de treinamento, também será feito o uso de regularização para um maior controle de sua generalização. Espera-se um desempenho similar ao SVM, porém sem a expectativa de maximização de margem durante o treinamento, a qual será controlada pelo coeficiente de regularização.

IV. RESULTADOS

Os classificadores avaliados serão identificados por: **GK-H**, **GK-P**, **MLPK-H**, **MLPK-P**, **GK-SVM** e **GK-RBF**. Os prefixos **GK** e **MLPK** representam a projeção utilizada no espaço de verossimilhanças: kernel Gaussiano e MLP, respectivamente. Os sufixos **H**, **P**, **SVM** e **RBF** representam o modelo utilizado: Perceptron Hebbiano, Perceptron Simples, SVM e RBF respectivamente.

A. Análise sobre problemas bi-dimensionais

Considerando três conjuntos de dados gerados artificialmente para diferentes estruturas, a superfície de decisão para cada classificador pode ser vista nas Figuras 1, 2 e 3. Observe que à medida em que a complexidade do modelo é aumentada, maior é a dificuldade encontrada pelos modelos lineares.

Para o problema da Figura 1, é possível ver que todos os modelos estimaram com precisão a superfície de separação. Entretanto, note que para o problema da Figura 2 os modelos lineares não tiveram um bom desempenho, provavelmente devido à qualidade da projeção obtida. Isso é evidenciado pelo problema da Figura 3, para o qual os modelos lineares obtiveram desempenho superior quando utilizada a projeção MLP. Destaca-se também que o RBF se ajustam muito bem aos dados e é de forma bem similar ao SVM, indicando que a estratégia adotada é bastante promissora.

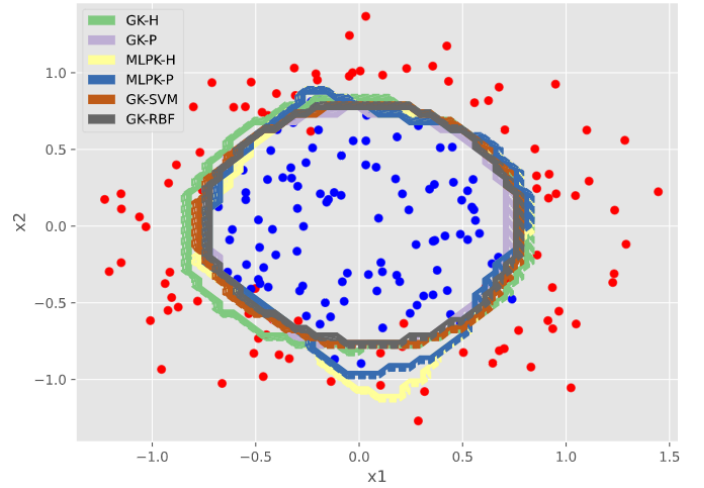


Fig. 1. Superfície de decisão para o problema de círculos concêntricos

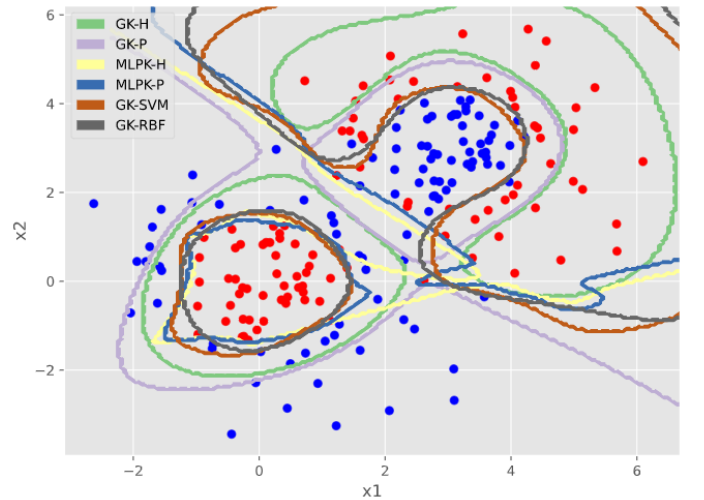


Fig. 2. Superfície de decisão para o problema de dois círculos concêntricos alternados

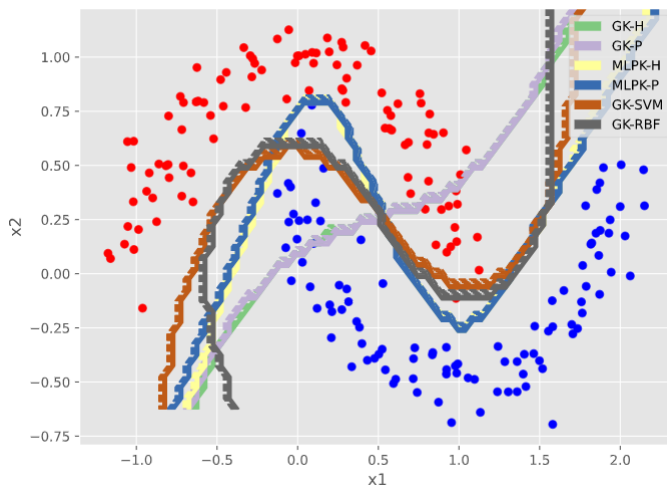


Fig. 3. Superfície de decisão para o problema das espirais

B. Análise do experimento

Para os problemas de classificação foram considerados os algoritmos Perceptron, RBF, ELM e ELM com aprendizado Hebbiano. Foi estabelecido um número fixo de 20 neurônios na camada escondida e não foi aplicada regularização ao ELM Hebbiano. 50 valores para o coeficiente de regularização foram escolhidos no intervalo $[0, 1]$.

A partir destes resultados, podemos chegar às seguintes conclusões:

- Para um intervalo de confiança de 95%, podemos afirmar que todos os algoritmos possuem desempenho médio superior ao modelo RBF nas bases de dados avaliadas.
- Para a base de dados *Breast Cancer*, os modelos ELM e Perceptron obtiveram desempenhos bastante similares. Além disso, podemos afirmar que eles obtiveram desempenho médio superior ao ELM Hebbiano para o intervalo de confiança de 95%.
- Para a base de dados *Statlog (Heart)* não podemos rejeitar a hipótese de que os modelos ELM, ELM Hebbiano e Perceptron possuam desempenhos iguais para o intervalo de confiança de 95%.
- Para a base de dados *Liver Disorder*, podemos afirmar que para um intervalo de confiança de 95%, o ELM possui desempenho superior aos demais modelos.
- Uma hipótese para o desempenho muito abaixo do esperado do RBF pode estar no número de neurônios escolhidos para a camada escondida. Este hiperparâmetro não foi ajustado para que a comparação com o ELM seja justa e também para limitar o tamanho do experimento a ser executado.
- A diferença de desempenho entre o ELM e sua versão com aprendizado Hebbiano está no fato deste última poder estar resultando em uma regularização excessiva. Isso sugere que uma implementação diferente do modelo Hebbiano é indicado para controlar melhor a sua generalização.

- É interessante notar que para problemas lineares não houve uma perda muito grande na AUC média dos modelos ELM e ELM Hebbiano. Entretanto, nota-se uma maior variabilidade para este último modelo, o que sugere uma maior sensibilidade à base de dados em estudo.

V. CONCLUSÕES

Neste trabalho foi feita uma avaliação do desempenho dos modelos ELM, RBF, Adaline, Perceptron e ELM Hebbiano sobre bases de dados de benchmark presentes na literatura. A partir de um experimento desenhado para tal finalidade, os resultados foram comparados estatisticamente utilizando a técnica de *bootstrapping* para a média das métricas AUC e erro quadrático médio, considerando um intervalo de confiança de 95%. Destaca-se o excelente desempenho do ELM em todos os conjuntos de dados, tanto de regressão e classificação. É importante destacar também o desempenho muito abaixo do esperado para o RBF nos problemas de classificação, sugerindo a necessidade de um melhor ajuste do número de neurônios na camada escondida, por exemplo. Notou-se também que os modelos lineares avaliados obtiveram bons resultados dependendo da base de dados em questão, conseguindo atingir um desempenho estatisticamente equivalente aos demais.

Uma surpresa foi o ELM Hebbiano, onde embora tenha sido usada uma abordagem bem simples de aprendizado Hebbiano, não apresentou um resultado muito inferior em relação ao ELM original para classificação. Destaca-se o fato de que não foi possível utilizar a estratégia de aprendizado Hebbiano com sucesso em problemas de regressão. Isso sugere que uma abordagem diferente deve ser utilizada nesta situação. Durante a execução do experimento, pôde ser observado também que o modelo RBF possui um tempo de treinamento muito maior que os demais. Isso fez com que ele fosse o gargalo de todo o experimento neste quesito, o que limitou um pouco o potencial de executar um número maior de partições para a validação cruzada, bem como avaliar um intervalo maior para o fator de regularização.

Como trabalhos futuros, é sugerido um ajuste fino de demais hiper-parâmetros do RBF como forma de tentar melhorar o seu desempenho. Além disso, também é interessante avaliar outras formas de aprendizado Hebbiano que poderiam ser utilizados com o ELM, bem como para melhorar seu poder de generalização e também ser aplicável a problemas de regressão com sucesso. Outra tarefa importante seria a realização de uma etapa de pré-processamento mais completa sobre bases de dados considerados, o qual não foi feita por restrições de tempo mas teria potencial para melhorar os resultados obtidos de todos os modelos.

REFERÊNCIAS

- [1] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [2] Murilo VF Menezes, Luiz CB Torres, and Antonio P Braga. Width optimization of rbf kernels for binary classification of support vector machines: A density estimation-based approach. *Pattern Recognition Letters*, 128:1–7, 2019.

TABELA I
ACURÁCIA DOS CLASSIFICADORES

| Base | ELM | GK-H | GK-P | GK-RBF | GK-SVM | MLPK-H | MLPK-P |
|--------------|---------------------|---------------------|---------------------|----------------------|------------------------|------------------------|------------------------|
| ILPD | 70.97 (69.82,72.20) | 52.95 (49.00,56.68) | 66.29 (63.18,69.65) | 71.45 (71.01,71.92) | 71.43 (70.94,71.96) | 60.77 (55.69,66.12) | 71.43 (70.90,72.02) |
| appendicitis | 85.82 (81.09,90.00) | 76.36 (66.45,86.37) | 87.64 (82.00,93.46) | 84.73 (78.00,90.91) | 86.73 (80.36,92.55) | 61.55 (49.99,74.28) | 75.36 (70.91,80.64) |
| australian | 85.94 (83.04,89.13) | 86.09 (83.33,88.99) | 87.60 (85.35,89.70) | 86.09 (83.62,89.13) | 85.80 (82.90,88.84) | 85.19 (82.93,87.60) | 83.91 (81.88,85.95) |
| banknote | 97.96 (97.45,98.54) | 91.76 (90.44,93.08) | 91.91 (90.38,93.30) | 99.78 (99.56,100.00) | 100.00 (100.00,100.00) | 100.00 (100.00,100.00) | 100.00 (100.00,100.00) |
| breastcancer | 96.48 (95.42,97.70) | 92.09 (89.99,94.02) | 94.53 (93.74,95.31) | 91.21 (90.33,92.09) | 98.82 (98.04,99.62) | 96.88 (96.30,97.26) | 96.31 (94.91,97.54) |
| bupa | 70.45 (67.33,73.45) | 50.14 (45.03,55.31) | 60.65 (58.71,62.75) | 61.10 (57.97,63.86) | 70.81 (65.53,76.24) | 55.96 (51.24,60.89) | 57.10 (53.28,60.34) |
| climate | 91.11 (90.37,91.67) | 57.59 (54.81,60.00) | 91.48 (90.93,92.04) | 91.11 (90.37,91.85) | 91.30 (90.74,91.85) | 78.15 (70.37,87.04) | 89.81 (88.52,91.11) |
| fertility | 86.00 (83.00,89.00) | 52.22 (42.22,61.14) | 86.00 (83.00,89.00) | 87.78 (85.56,90.00) | 87.00 (84.00,90.00) | 60.00 (50.00,70.00) | 87.78 (85.56,91.11) |
| glass | 96.26 (94.37,98.48) | 86.84 (79.99,94.32) | 97.08 (95.35,98.32) | 97.68 (95.84,99.94) | 96.26 (93.90,99.05) | 95.30 (92.49,98.61) | 96.26 (93.94,98.55) |
| haberman | 76.24 (74.43,77.98) | 53.59 (50.04,57.03) | 74.69 (73.27,76.10) | 72.23 (70.04,74.55) | 72.56 (71.01,74.11) | 52.57 (44.63,60.08) | 72.54 (71.30,74.05) |
| heart | 84.77 (83.54,86.01) | 78.19 (74.48,82.30) | 82.22 (78.52,85.56) | 85.19 (82.59,87.78) | 79.63 (76.67,82.59) | 82.22 (78.52,85.93) | 87.50 (84.25,90.28) |
| ionosphere | 88.31 (85.98,91.17) | 70.08 (66.00,74.65) | 87.00 (84.11,90.20) | 89.47 (85.14,93.22) | 93.73 (91.17,96.32) | 92.89 (90.63,95.14) | 93.17 (90.88,95.62) |
| pima | 76.42 (74.18,78.51) | 67.44 (65.56,69.13) | 72.52 (69.45,75.45) | 75.91 (73.45,78.52) | 77.74 (75.83,79.59) | 75.52 (72.63,78.08) | 76.77 (75.32,78.02) |
| segmentation | 99.65 (99.48,99.83) | 57.06 (54.93,59.18) | 88.79 (87.97,89.65) | 98.61 (98.14,99.09) | 99.70 (99.48,99.91) | 78.16 (75.71,80.42) | 85.71 (85.71,85.71) |
| sonar | 65.03 (61.01,69.58) | 46.88 (37.57,56.86) | 65.93 (56.19,75.98) | 60.38 (49.12,70.64) | 53.90 (51.85,56.28) | 69.36 (61.19,77.29) | 66.48 (58.54,73.54) |

- [3] Shitong Wang, Zhaohong Deng, Fu-lai Chung, and Wenjun Hu. From gaussian kernel density estimation to kernel methods. *International Journal of Machine Learning and Cybernetics*, 4(2):119–137, 2013.
- [4] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.
- [5] Manuel Fernandez-Delgado, Jorge Ribeiro, Eva Cernadas, and Senén Barro Ameneiro. Direct parallel perceptrons (dpps): fast analytical calculation of the parallel perceptrons weights with margin control for classification tasks. *IEEE transactions on neural networks*, 22(11):1837–1848, 2011.
- [6] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, volume 2, pages 985–990. IEEE, 2004.
- [7] M Menezes, L Torres, and A Braga. Otimização da largura de kernels rbf para máquinas de vetores de suporte: Uma abordagem baseada em estimativa de densidades. In *XIII Congresso Brasileiro de Inteligência Computacional*, 2017.
- [8] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [9] Steven L Salzberg. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data mining and knowledge discovery*, 1(3):317–328, 1997.
- [10] Charles R. Harris, K. Jarrod Millman, St’efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernandez del Rio, Mark Wiebe, Pearu Peterson, Pierre G’erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
- [11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [12] Euler Guimarães Horta. Aplicação de máquinas de aprendizado extremo ao problema de aprendizado ativo. 2015.
- [13] D. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Syst.*, 2, 1988.
- [14] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice-Hall, Inc., 2007.