Projeto 3 FSI

Matheus Eiji Endo
Estudante de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
matheus.endo@hotmail.com

Johannes Peter Schulte
Estudante de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
johpetsc@gmail.com

***Resumo*—Relatório do projeto 3 da disciplina de Fundamentos de Sistemas Inteligentes.**
***Index Terms*—SVM, Classificação**

I. Introdução

De acordo com o Instituto Nacional do Câncer, o câncer de mama é o tipo de câncer mais comum em mulheres no Brasil, para o ano de 2018 estimam-se 59.700 casos novos de câncer de mama. Dessa forma, utilizar novas tecnologias para tentar combater essa doença é de extrema importância nos dias atuais.

Por isso este relatório apresenta uma breve explicação sobre o método de classificação supervisiona chamado de Máquina de Vetores de Suporte, ou Support Vector Machine(SVM) em inglês, e o seu uso para classificar dados sobre câncer de mama e uma avaliação sobre seus resultados.

II. Support Vector Machine

O SVM é um algoritmo supervisionado, logo exige que os dados sejam rotulados, ou seja, dado os dados de entrada também se tem os dados de saída, a partir desses dados o algoritmo cria um modelo e assim pode classificar novos dados, que não sejam rotulados.

O objetivo desse algoritmo é considerar os dados como pontos e determinar um hiperplano, uma reta, que separe as classes, maximizando a distância entre cada classe e o determinado hiperplano, esse hiperplano é determinado por um subconjunto de pontos da classes, que formam vetores, chamados de vetores de suporte, daí o nome do algoritmo.

Um grande números de hiperplanos pode ser considerado, mas o hiperplano de interesse do algoritmo é o chamado hiperplano ótimo, que é aquele que maximiza a margem, ou seja, maximiza a distância entre o hiperplano e cada classe. Um exemplo desse hiperplano ótimo pode ser visto na fig. 1, onde o hiperplano é formado a partir dos dois vetores de linha pontilhada.

Essa técnica se aplica somente aos casos onde os espaços são lineares, no caso dos dados não serem lineares não é possível determinar uma reta que os separes de forma eficaz no espaço. Para resolver esses casos, a ideia é realizar operações com os dados de entrada, para mapeá-los para um espaço de

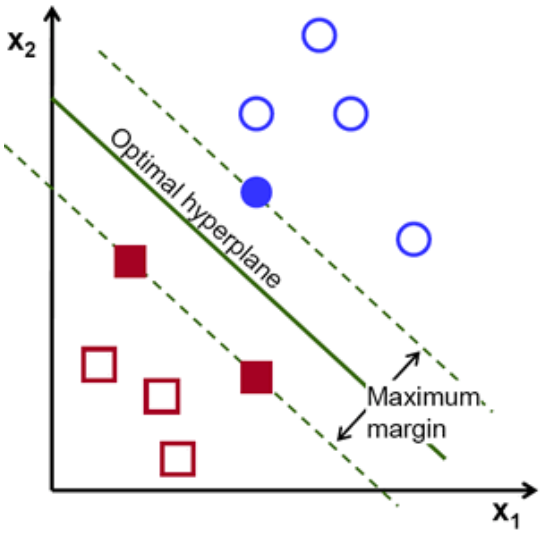

Figura 1. [1]Hiperplano ótimo.

maior dimensão que seja linear, chamado de espaço de característica ou *feature space* em inglês. As funções denominadas de Kernel para SVMs são funções que fazem justamente isso, transformam espaços não lineares em lineares, pode-se ver na fig. 2 que um espaço que antes era não linear se torna linear após aplicar uma função kernel, permitindo a determinação de um hiperplano ótimo no novo espaço. Entre as diversas funções as mais comuns são a função kernel linear,polinomial, gaussiana, de base radial(FBR) e a sigmoide. A fórmula de algumas dessas funções podem ser vistas na tabela 1.

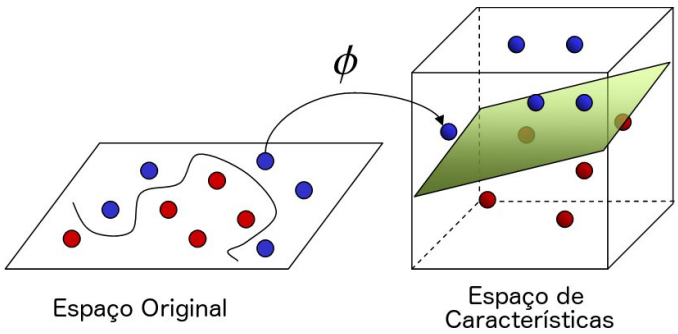

Figura 2. Exemplo kernel.

[1]https://aitrends.com/ai-insider/support-vector-machines-svm-ai-self-driving-cars/

Kernel	Função	Parâmetros height
Polinomial	$(\delta(x_i \cdot x_j) + \kappa)^d$	$\delta, \kappa ed$
Gaussiano	$exp(-\frac{\|x-y\|^2}{2\sigma^2}))$	σ
FBR	exp(-$\gamma \|x_i - x_j\|^2$)	γ
Sigmoide	tanh($\delta(x_i \cdot x_j) + \kappa$)	$\delta e \kappa$

Tabela I
TABELA DE ALGUMAS FUNCÕES KERNEL.

III. PROBLEMA

O problema do projeto 3 é utilizar o algoritmo de classificação SVM para classificar um conjunto de dados sobre o câncer de mama, fazendo uso de dados rotulados e dividindo-os em dois conjuntos, teste e treinamento e analisar o desempenho para diferentes funções Kernel.

Os dados podem ser encontrados em aqui. Os dados forem adquiridos analisando fotos de células de seios, e através dessas fotos forem decididos atributos. Há 569 instâncias das classes, que são divididas em Maligno(célula cancerígena) e Benigno(célula saudável), há também 32 atributos de cada instância. Os dois primeiros atributos são o id e a classe a qual o instância o dado pertence, já os outros 30 atributos são os que devem ser utilizados para o treinamento do algoritmo.

- *radius*
- *texture*
- *perimeter*
- *area*
- *smoothness*
- *compactness*
- *concavity*
- *concave points*
- *symmetry*
- *fractal dimension*

Para cada um desses atributos, há a média, o desvio padrão e a pior medida de cada desses atributos, totalizando 30.

IV. IMPLEMENTAÇÃO

Para a implementação do terceiro projeto da disciplina foi utilizada a linguagem de programação *Python(3)*, devido a sua vasta coleção de bibliotecas referentes à aprendizagem de máquinas. A biblioteca *sklearn* foi novamente escolhida para a parte aplicada do projeto, devido a eficiência que teve no primeiro e segundo projeto e por apresentar funções específicas para o tema do projeto, Support Vector Machine.

O script deve ser executado num terminal Linux com as seguintes bibliotecas instaladas: *pandas*, para leitura do arquivo csv; *numpy* para o melhor manuseamento de matrizes no código; *sklearn*, para os cálculos referentes a Floresta Randômica; *matplotlyb*, para gerar a matriz de confusão em gráfico; e *sys*, para manter a execução do script de forma linear porém com mais controle. Funções de sklearn utilizadas:

- *confusion_matriz*, que gera uma matriz de confusão baseada nos dados e resultados dos testes.
- *accuracy_score*, que testa a precisão dos testes realizados.
- *classification_report*, que reporta as principais métricas de classificação.
- *train_test_split*, que divide os dados em 70/30 para teste como requisitado.
- *SVC*, é a função responsável por realizar os testes com diferentes kernels.

Para a execução do script, o primeiro passo foi ler os dados do arquivo wdbc.data.csv com a biblioteca pandas, dividindo os atributos com seus labels logo depois. A função train_test_split é chamada para realizar a divisão dos dados de treinamento e teste em 70/30. Para o teste do kernel linear foi utilizado um loop para serem testados todos os valores de 2^5 até 2^15. Dentro do loop são gerados os valores de precisão e de erro para os testes. O teste para kernel gaussiano foi feito com uma função especifica por não existir parâmetro para ela na função SVC. Para o teste FBR foi feito da mesmo forma que o kernel linear, apenas mudando o parâmetro da função.

V. RESULTADOS

Após rodar o programa algumas vezes foi possível perceber um padrão nos resultados de kernel gaussiano e kernel FBR, por outro lado os testes com kernel linear tiveram valores bem variados e resultados não muito conclusivos.

Para o kernel gaussiano foi utilizado o valor de Sigma = 10, e apresentou uma precisão, em média, de 85% onde a precisão para benignos foi de 89% e para malignos foi 80%. Observando a figura 3 podemos perceber esses resultados, onde 0-0 representa a relação benigno-benigno, e 1-1 maligno-maligno. A taxa de erro pode ser observada em 0-1 e 1-0.

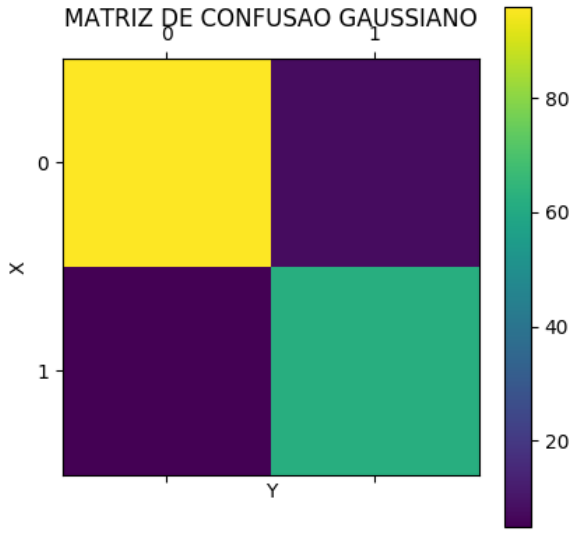

Figura 3. Matriz de Confusão do kernel gaussiano .

No kernel FBR observamos o pior resultado de todos os testes, com uma precisão média de 66% para os casos benignos, pois o modelo não classificou nenhum dos casos como maligno, mesmo utilizando diversos valores de γ, esse

modelo não se apresentou eficiente na detecção de tumores para câncer de mama. A matriz de confusão abaixo pode ser analisada da mesma forma que a matriz do kernel gaussiano.

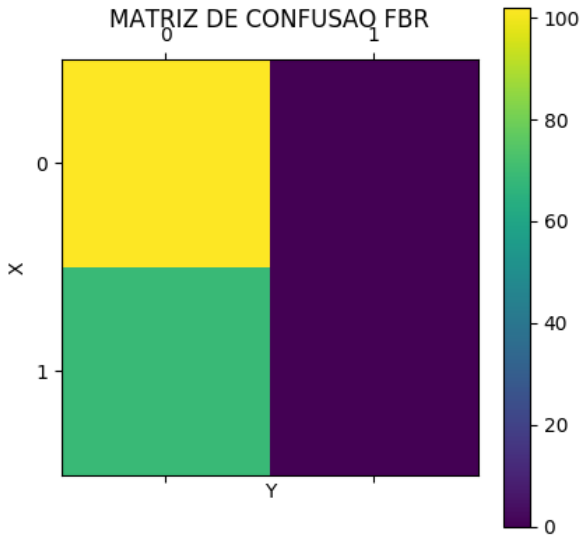

Figura 4. Matriz de Confusão do kernel FBR.

Os valores para o kernel linear variam bastante de teste para teste, sendo observado que os valores para treino e teste são aleatórios, os resultados foram muito dependentes dessa seleção. Mesmo assim, foi o kernel com a maior precisão média, variando dos piores valores de 90% até uma média de 97% em alguns casos, com a precisão de casos benignos sendo um pouco melhor. Porém, não conseguimos achar um padrão para os resultados de diferentes valores de C. Abaixo são apresentados 3 execuções do programa, com os valores de erro sendo apresentados nas tabelas. Algumas observações foram possíveis, como, por exemplo, que os valores medianos (entre 1 e 9) tiveram a melhore média de erros, mas os resultados não foram constantes o suficiente para concluir que são melhores. Também foi possível observar que os valores mais altos e mais baixo possuem variações bem maiores entre eles, diferente dos valores medianos.

Por fim, mesmo com resultados pouco conclusivos, podemos observar que, mesmo com uma tempo de execução maior que o kernel gaussiano e RBF, usando o linear temos resultados mais precisos, e usando valores de C entre 2ˆ 2 e 2ˆ 9, temos resultados mais confiáveis. Assim podemos pensar que os dados sejam bem linearmente separáveis, pois o svm com melhor resultado foi o com o kernel linear.

Referências

[1] Documentação da função SVC, da biblioteca sklearn, http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[2] Dados retirados de http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

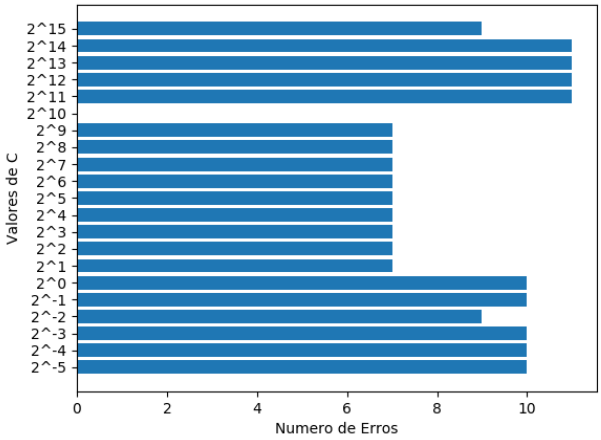

Figura 5. Número de erros para cada C exemplo 1.

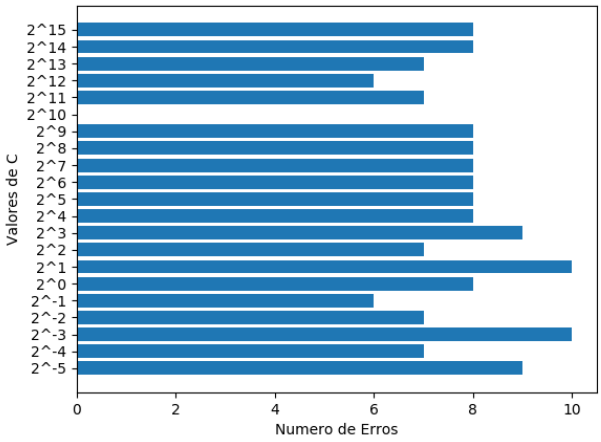

Figura 6. Número de erros para cada C exemplo 2.

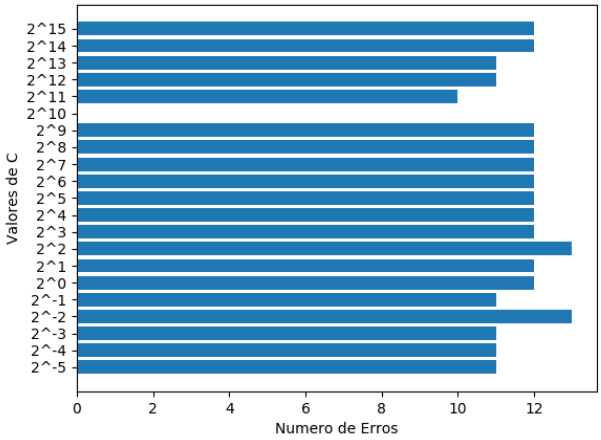

Figura 7. Número de erros para cada C exemplo 3.