

# IMPLEMENTAÇÃO DO FRE<sub>x</sub>\_SVM: MÁQUINAS DE VETOR SUPORTE PARA CLASSIFICAÇÃO EM MÚLTIPLAS CLASSES

**Aluno: Aarão Irving Manhães Marins**  
**Orientador: Marley Maria Bernardes Rebuzzi Vellasco**

## Introdução

A máquina de vetor suporte é uma técnica de aprendizado estatístico. Esta técnica utiliza funções que mapeiam os dados de entrada em um espaço de características de dimensão maior, onde é construído um hiperplano de separação, para solucionar problemas de classificação.

Houve um estudo sobre o funcionamento de máquinas de vetor suporte [1] e de regras fuzzy. Foram analisados vários modelos para classificação binária usando máquinas de vetor suporte. Escolhendo um desses modelos, pôde-se implementar a classificação em múltiplas classes.

## Objetivos

Implementar um software baseado no modelo FRE<sub>x</sub>\_SVM [2] para problemas de classificação, tanto binários quanto em múltiplas classes, e preparar este software para adaptação de métodos para extração de regras fuzzy.

## Metodologia

Para solucionar o problema de classificação de um conjunto de dados com duas classes, é necessário usar um conjunto de treinamento. Aplicando máquinas de vetor suporte neste problema, é possível produzir um classificador para estes dados.

Este conjunto de dados foi normalizado, para a obtenção de melhores resultados pelas máquinas de vetor suporte, pois assim, evita-se a perda de generalização durante a resolução do problema de minimização gerado.

A formulação da máquina de vetor suporte para classificação [3] gera um problema de programação quadrática. Para resolver esse problema, é necessário calcular os multiplicadores de Lagrange.

O aplicativo foi implementado na linguagem orientada a objeto C#, porém foi preciso utilizar a função 'quadprog' de outro aplicativo (MATLAB), que serve para o cálculo dos multiplicadores de Lagrange. Para importar estes métodos do MATLAB, foi utilizada uma biblioteca pronta, que envia linhas de comando e recebe os resultados de volta na aplicação.

O usuário do aplicativo deve informar uma constante de regularização, escolher uma função Kernel, assim como seus parâmetros. Também pode escolher o método de classificação em múltiplas classes.

De acordo com a constante de regularização, pode-se definir o peso do número de erros. Deste modo, quanto maior o valor dessa constante, menor o número de erros.

Foram usadas funções Kernel para construir um mapeamento dos dados num espaço de dimensão mais alta, o espaço de características. Existem várias funções deste tipo, e foram escolhidas três para serem implementadas neste aplicativo: linear, gaussiana e polinomial.

Depois de calcular os multiplicadores de Lagrange, verifica-se se eles são valores significantes (com módulo maior que zero), para então obter os vetores suporte. Além deles, também foram calculados termos de bias, para o ajuste da função de classificação.

Tendo então o treinamento das máquinas de vetor suporte funcionado para problemas duas classes, foram implementados dois métodos para classificação em múltiplas classes [4], ambos com uma proposta de dividir o problema de múltiplas classes em vários problemas binários. Para isto, foi realizada uma reestruturação dos dados do aplicativo para armazenar os resultados de diversos treinamentos.

O método Decomposição um por classe (“One-Against-All”) utiliza todos os dados em todos os treinamentos, e o número de treinamentos é igual ao número de classes. Para cada treino, uma classe é separada das outras, sendo que todas as outras funcionam como se fossem uma classe apenas.

Já no método Separação das classes duas a duas (“One-Against-One”), a quantidade de treinamentos é maior, pois neste caso, todas as classes se combinam. Além disso, somente os dados das classes envolvidas em cada combinação são enviados para seu respectivo treinamento.

Uma vez obtidos os resultados de todos os treinamentos binários, pôde-se testar os métodos. Para realizar estes testes, o usuário deve informar um conjunto de dados para teste, que seja do mesmo tipo do conjunto usado para o treinamento.

Foram realizados testes com diversos bancos de dados e parâmetros diferentes, e os resultados, em geral, foram compatíveis com os resultados observados em outras implementações.

## Conclusões

A pesquisa realizada sobre diversas formas de implementação de máquinas de vetor suporte para classificação colaborou para o desenvolvimento do aplicativo. Como resultado, a forma de implementação foi escolhida para se obter resultados mais próximos da teoria.

O aplicativo permite a entrada da constante de regularização, a escolha da função Kernel a ser utilizada no treinamento, assim como seus atributos. Também é possível escolher o método de classificação para o problema de classificação em múltiplas classes.

O método de treinamento dos dados tem um grande custo computacional, sendo sempre melhor não utilizar conjuntos muito grandes de dados para esta fase do programa. Já no método de teste, o tamanho do conjunto de teste quase não requer esforço, possibilitando assim testar a eficiência dos treinamentos em grandes conjuntos de dados.

## Referências

- 1 - CRISTIANINI, N., SHAWE-TAYLOR J., **An introduction to support Vector Machines: and other kernel-based learning methods**. New York: Cambridge University Press, 1999. 189p.
- 2 - CHAVES, A. C. F. **Extração de Regras Fuzzy para Máquinas de Vetores Suporte (SVM) para Classificação em Múltiplas Classes**. Rio de Janeiro, 2006. 225p. Tese de Doutorado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.
- 3 - GUNN, S. Support Vector Machines for Classification and Regression. **ISIS Technical Report**. 1998.
- 4 - HSU, C. W., LIN, C. J., A comparison of methods for multiclass Support Vector Machines. **IEEE Trans on Neural Networks**, v.13, n.2, p. 415-425. 2002.