

# Classificador SVM e LSSVM

Savio Lopes Rabelo

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)  
Programa de Pós-Graduação em Ciência da Computação (PPGCC)  
Campus Fortaleza – CE – Brasil

savio.rabelo@ppgcc.ifce.edu.br

**Resumo.** *Este relatório descreve a implementação do classificador SVM e LS-SMV. A metodologia utilizada para a implementação é constituída por duas fases: treinamento e teste, com cada conjunto sendo composto por 80% e 20% das bases de dados, respectivamente. Foram usadas quatro bases de dados disponíveis online no repositório UCI Machine Learning. Os resultados são bastante satisfatórios, chegando em taxas de acerto em 100% em algumas bases.*

## 1. Introdução

As Máquinas de Vetor de Suporte de Mínimos Quadrados (do inglês, *Least Squares Support Vector machines*, LSSVM) são versões de mínimos quadrados de Máquinas de Vetores de Suporte (do inglês, *Support Vector Machines*, SVM), que são um conjunto de métodos de aprendizado supervisionados relacionados que analisam dados e reconhecem padrões e que são usados para classificação e regressão. Na versão LSSVM a solução é encontrada resolvendo um conjunto de equações lineares em vez de um problema de programação quadrática convexa para SVMs clássicos. Classificadores SVM de mínimos quadrados foram propostos por [Suykens and Vandewalle 1999]. As LSSVMs são uma classe de métodos de aprendizado baseados em Kernel.

Para a Função de Kernel  $K$ , normalmente, as seguintes opções são utilizadas:

- Kernel Linear:  $K(x, x_i) = x_i^T x$ ,
- Kernel Polinomial de grau  $d$ :  $K(x, x_i) = (x_i^T x + 1)^d$ ,
- Kernel Radial Basis Function (RBF):  $K(x, x_i) = \exp\left(\frac{-\|x - x_i\|^2}{\sigma^2}\right)$ ,
- Kernel MLP:  $K(x, x_i) = \tanh(k x_i^T x + \theta)$ ,

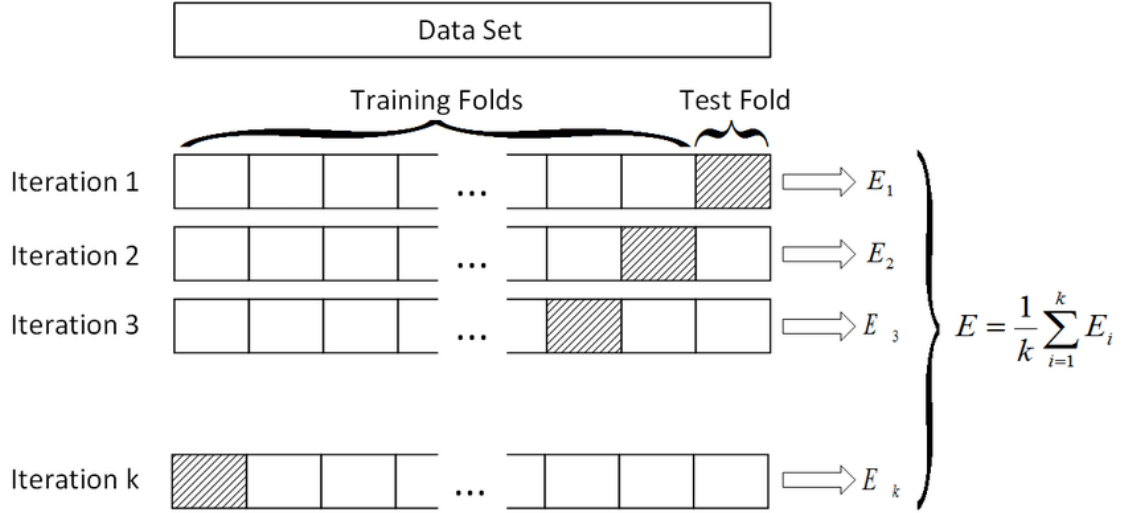
onde  $d, \sigma, k, \theta$  são constantes.

## 2. Metodologia

No primeiro momento foi realizada a separação do conjunto de dados em dois subconjuntos: treinamento e teste. Os valores utilizados para os conjuntos equivalem a 80% do conjunto original para a fase de treinamento e 20% do conjunto original para a fase de teste. Logo depois, os dados foram normalizados para eliminação de redundâncias indesejadas e também foram embaralhados.

Além disso, também foi utilizada a busca em grade com validação cruzada  $k$ -fold. A busca em grade é uma busca com o objetivo de encontrar os melhores parâmetros. Já o método de validação cruzada  $k$ -fold consiste em dividir o conjunto total de dados em  $k$  subconjuntos mutuamente exclusivos do mesmo tamanho e, a partir disto, um subconjunto

é utilizado para teste e os  $k-1$  restantes são utilizados para estimação dos parâmetros e calcula-se a acurácia do modelo. Este processo é realizado  $k$  vezes alternando de forma circular o subconjunto de teste. A Figura 1 mostra o esquema realizado pelo  $k$ -fold. Ao final das  $k$  iterações calcula-se a acurácia sobre os erros encontrados, obtendo assim uma medida mais confiável sobre a capacidade do modelo de representar o processo gerador dos dados.



**Figura 1. Método k-fold**

Para a avaliação dos resultados alcançados na classificação, foram utilizados as seguintes métricas: a precisão ou valor preditivo positivo, taxa de sensibilidade ou taxa positiva verdadeira, especificidade ou taxa real negativa e acurácia. As Equações são apresentadas a seguir:

$$Precisao = \frac{VP}{VP + FP}, \quad (1)$$

$$Sensibilidade = \frac{VP}{VP + VN}, \quad (2)$$

$$Especificidade = \frac{VN}{N} = \frac{VN}{FP + VN}, \quad (3)$$

$$Acuracia = \frac{VP + VN}{P + N}, \quad (4)$$

onde P e N é o número de padrões de cada classe. VP é o verdadeiro positivo. VN é o verdadeiro negativo. FP é o falso positivo e FN é o falso negativo.

Para análise comparativas neste estudo, foram usados quatro conjuntos de dados: *Iris Flower Data Set* em sua versão binária (setosa vs outras), *Vertebral Column Data Set* em sua versão binária (paciente patológico vs paciente normal), *Dermatology Data*

*Set* (paciente patológico vs paciente normal) e *Breast Cancer Wisconsin Data Set* (paciente com câncer benigno vs paciente com câncer maligno); todos disponíveis online no repositório *UCI Machine Learning* [Lichman 2013].

### 3. Simulações Computacionais

Para realizar os experimentos, foi utilizado um computador com a seguinte configuração: processador Intel(R) Core(TM) i7-6500U a 2.5 GHz com 8 GB de RAM e executando Windows 10. Além disso, foi utilizado a linguagem de programação MATLAB. Todos os testes foram feitos com 50 realizações em cada base.

A Tabela 1 mostra os resultados do classificador SVM com Kernel Linear em todas as bases de dados, levando em consideração as métricas já mencionadas.

Métricas (%)	Bases de Dados				
	Íris	Coluna	Dermatologia	Câncer	Artificial
Acurácia	100,00	84,16	100,00	96,43	100,00
Taxa Mínima	100,00	75,81	100,00	93,00	100,00
Taxa Máxima	100,00	93,55	100,00	100,00	100,00
Desvio Padrão	00,00	4,12	00,00	2,07	00,00
Sensibilidade	100,00	76,80	100,00	95,70	100,00
Especificidade	100,00	87,77	100,00	97,00	100,00
Precisão	100,00	75,26	100,00	92,65	100,00
Tempo (s)	61,95	138,47	139,41	120,78	59,29

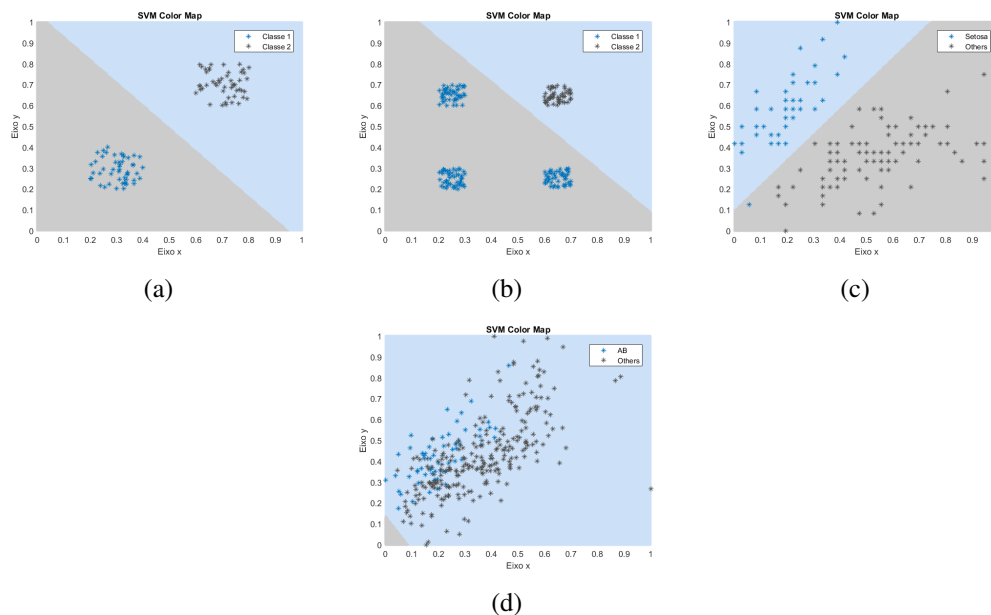
**Tabela 1. Resultados do classificador SVM com Kernel Linear.**

A Tabela 2 mostra os resultados do classificador LSSVM com Kernel Linear em todas as bases de dados, levando em consideração as métricas já mencionadas.

Métricas (%)	Bases de Dados				
	Íris	Coluna	Dermatologia	Câncer	Artificial
Acurácia	100,00	82,16	99,86	94,86	100,00
Taxa Mínima	100,00	72,58	97,30	92,86	100,00
Taxa Máxima	100,00	91,94	100,00	96,43	100,00
Desvio Padrão	00,00	5,02	00,49	1,85	00,00
Sensibilidade	100,00	64,80	100,00	89,47	100,00
Especificidade	100,00	91,28	99,59	97,82	100,00
Precisão	100,00	77,69	99,80	95,63	100,00
Tempo (s)	29,47	45,33	62,38	35,42	29,44

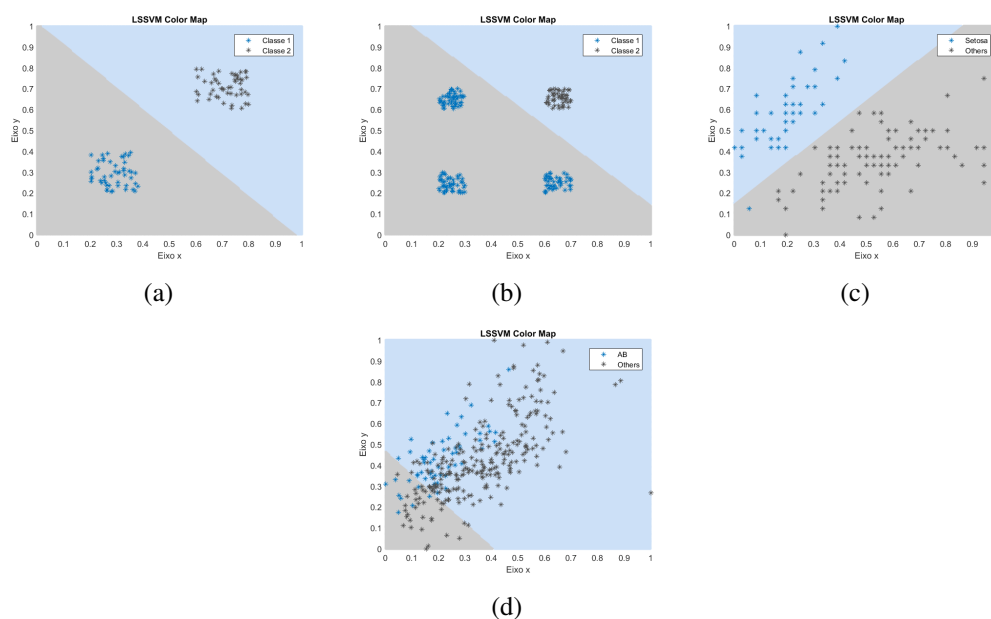
**Tabela 2. Resultados do classificador LSSVM com Kernel Linear.**

Na Figura 2 é apresentada a superfície de decisão com o classificador SVM com Kernel Linear.



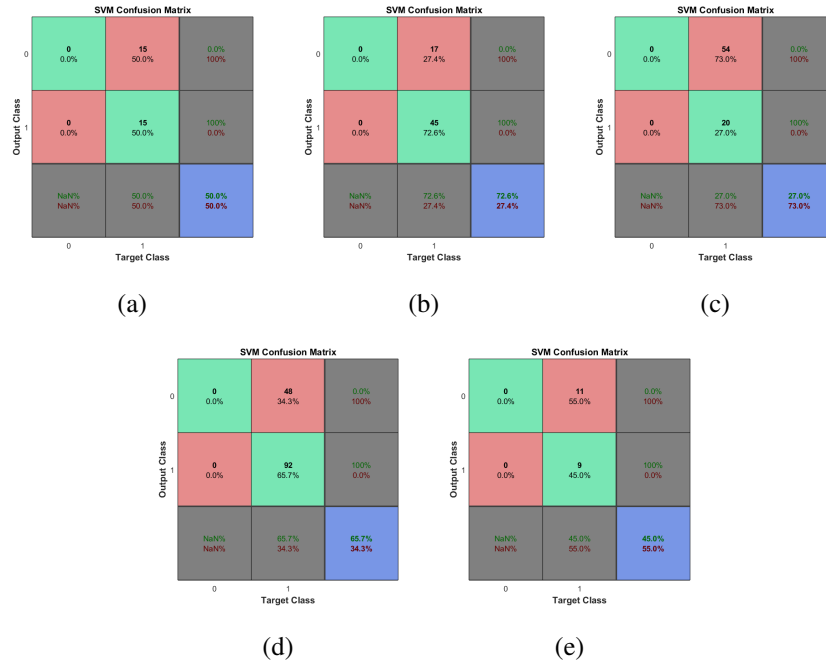
**Figura 2. Superfície de decisão do classificador SVM com Kernel Linear. (a) Base artificial com 2 classes. (b) Base artificial com 3 classes. (c) Base artificial AND. (d) Íris com dois primeiros atributos. (e) Coluna com dois primeiros atributos.**

Na Figura 3 é apresentada a superfície de decisão com o classificador SVM com Kernel Linear.



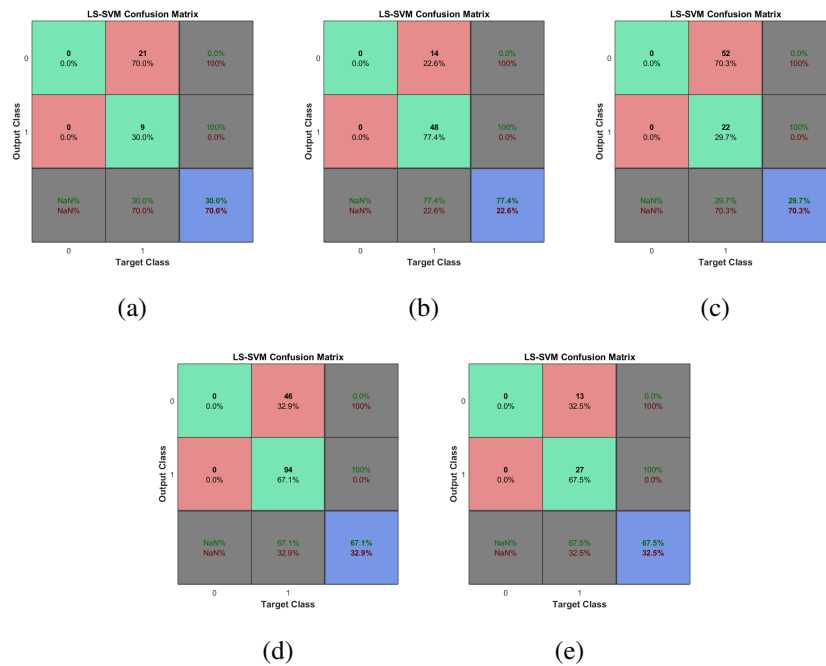
**Figura 3. Superfície de decisão do classificador LSSVM com Kernel Linear. (a) Base artificial com 2 classes. (b) Base artificial com 3 classes. (c) Base artificial AND. (d) Íris com dois primeiros atributos. (e) Coluna com dois primeiros atributos.**

A Figura 4 apresenta uma matriz de confusão de cada base de dados do classificador SVM com Kernel Linear. Essa matriz é a matriz que ficou mais perto da acurácia.



**Figura 4. Matriz de Confusão para o classificador SVM com Kernel Linear. (a) Íris. (b) Coluna. (c) Dermatologia. (d) Câncer. (e) Artificial (2C).**

A Figura 5 apresenta uma matriz de confusão de cada base de dados do classificador LSSVM com Kernel Linear. Essa matriz é a matriz que ficou mais perto da acurácia.



**Figura 5. Matriz de Confusão para o classificador LSSVM com Kernel Linear. (a) Íris. (b) Coluna. (c) Dermatologia. (d) Câncer. (e) Artificial (2C).**

## 4. Resultados

Classificadores LSSVM, são bastante parecidos com classificadores SVM. Isso porque partem do mesmo princípio de encontrar o hiperplano de separação ótimo entre duas classes. Entretanto, o processo de treinamento é diferente do SVM, sendo o classificador estudado nesta atividade mais rápido para treinar o modelo de classificação.

Nesta atividade, pode-se constatar a similaridade dos classificadores LSSVM em termos de acurácia sobre classificadores baseados no teorema de Bayes. Um fato que deve ser mencionado é o uso das funções Kernels no classificador. O que foi observado é que nem sempre um Kernel que aparenta ser melhor em termos de generalização, terá de fato melhor desempenho. O caso do Kernel Linear constata isto, quando houve bases de dados em que o Kernel Linear trouxe melhor desempenho para o LSSVM. Portanto, o uso do Kernel está ligado diretamente ao problema a ser solucionado.

## Referências

- Lichman, M. (2013). UCI machine learning repository. <http://archive.ics.uci.edu/ml>. Acesso em março de 2019.
- Suykens, J. A. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300.