# Otimização da Largura de Kernels RBF para Máquinas de Vetores de Suporte: Uma Abordagem Baseada em Estimativa de Densidades

Murilo V. F. Menezes, Luiz C. B. Torres*, Antônio P. Braga

Universidade Federal de Minas Gerais, Departamento de Engenharia Eletrônica
Av.Antônio Carlos, 6627, Pampulha 30161-970, Belo Horizonte, MG, Brasil
murilovfm@gmail.com, luizlitc@gmail.com, apbraga@ufmg.br

**Resumo** Kernels são ferramentas utilizadas para modelar não-linearidades em dados, desempenhando um papel principal em modelos como SVMs. A otimização de seus parâmetros para se ajustar a cada conjunto de dados é um desafio frequentemente enfrentado. Este problema é usualmente dirimido usando validação cruzada, técnica baseada no cálculo de desempenho sobre uma faixa de valores, não levando em consideração informações diretas sobre a disposição dos dados. Este trabalho propõe uma abordagem alternativa, baseada na estimativa de densidades, sob a qual se analisa a estrutura do conjunto de dados, possibilitando assim o projeto de um kernel adequado para cada problema.

**Palavras-Chave:** Classificação, Kernel RBF, SVM, Estimativa de Densidades

## 1 Introdução

O ajuste de parâmetros em métodos de aprendizado indutivo é usualmente descrito como um problema de otimização de mútiplas funções-objetivo, relacionadas ao ajuste do modelo a um conjunto de dados amostrados (erro) e à complexidade da função utilizada para descrever o modelo. Do ponto de vista prático, uma função representativa da complexidade, que descreva a dimensão V-C do modelo [1], deve ser escolhida para viabilizar o cálculo do ajuste de complexidade. A norma do vetor de pesos é usualmente utilizada como medida indireta da complexidade, já que a mesma está relacionada à margem de separação entre as classes definidas em problemas de classificação de padrões.

Há na literatura várias abordagens para o tratamento do problema de bi-objetivo, caracterizado pelas funções de erro e de norma do vetor de parâmetros, tais como a otimização Multi-objetivo de Redes Neurais [2] e a otimização com restrição adotada no treinamento de Máquinas de Vetores de Suporte (SVM) [1]. Qualquer que seja a abordagem adotada, o modelo final sempre dependerá

* Bolsista do CNPq–Brasil (N°150254/2016-4)

de uma terceira função objetivo que descreverá a estratégia de decisão para a seleção final do modelo, sendo que não existe prova formal de qual deva ser o melhor critério de decisão a ser adotado.

Uma estratégia de decisão frequentemente adotada, particularmente no treinamento de SVMs, é a utilização de validação cruzada que, apesar de eficiente em vários problemas, é também dependendente do tamanho da amostra e da sua representatividade. Assim, não há um critério geral que defina de maneira clara um método universal para a seleção de modelos, problema este caracterizado como um "dilema"no ajuste de modelos baseados em dados ("*bias and variance dilemma*"[3]). Não obstante, há na literatura da área uma busca constante por métodos de treinamento autônomos ou que tenham menor dependência do usuário na inicialização de parâmetros ou escolha do critério para a seleção final do modelo [4, 5].

Este trabalho descreve um método para ajuste de parâmetros da função de kernel de modelos SVM, especificamente a largura de base de funções de kernel radiais, o qual é baseado no comportamento das projeções das funções de similaridade calculadas com base nestes kernels, para problemas de classificação binária. O método é baseado no algoritmo de estimativa de densidades KDE (*Kernel Density Estimator*) [6, 7]. Partindo-se da estrutura dos dados, podemos encontrar um parâmetro que separe as amostras de classes distintas, ainda assim mantendo a capacidade de generalização do modelo, permitindo a discriminação efetiva das classes. Este método é apresentado como uma alternativa a métodos que não consideram informações da disposição das amostras, como validação cruzada [8].

O artigo apresenta, primeiramente, o Método Kernel e o Kernel RBF, alvo deste trabalho, na seção 2. O método KDE é coberto na seção 3, que descreve também o espaço de verossimilhanças, componente essencial do trabalho. A seção 4 percorre o algoritmo SVM, utilizado para aplicar a técnica proposta. As seções 5 e 6 expõem a metodologia do trabalho e os resultados obtidos, e, por fim, a seção 7 apresenta as conclusões.

## 2 Método Kernel

Em um problema de classificação, dados são estruturados sob diversos perfis. Devido a esta diversidade, muitas vezes é necessário lidar com dados não-linearmente separáveis.

Para um espaço d-dimensional, uma função $g(\mathbf{x})$ de discriminação linear pode ser definida a partir de um vetor $\mathbf{x} = x_1, x_2, ..., x_d$, de forma geral, de acordo com a Equação 1.

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i, \tag{1}$$

onde $w_0, w_1, ..., w_d$ são componentes de um vetor de pesos $\mathbf{w}$, como descrito em [9]. Um problema é não-linearmente separável se não existe um vetor $\mathbf{w}$ que

consiga classificar corretamente todas as amostras. O método kernel é então uma ferramenta utilizada para mapear os dados do espaço de entrada para um espaço onde este conjunto de coeficientes $\mathbf{w}$ exista, sendo possível separá-los.

Um kernel consiste em uma função de mapeamento, com a qual pode-se obter relações entre amostras no espaço transformado, sem que se tenha a necessidade de explicitá-lo [10]. Dentre vários tipos de kernels, o kernel de base radial (*Radial Basis Function - RBF*) é muito usado, utilizando uma função Gaussiana, de forma $K(\mathbf{x_1}, \mathbf{x_2}) = \exp(-\frac{||\mathbf{x_1}-\mathbf{x_2}||^2}{2\sigma^2})$ [11], para calcular a similaridade entre duas amostras. Para isto, é calculado o valor da distância entre as amostras no espaço de entrada, e, utilizando um parâmetro $\sigma$, chamado largura de base da função Gaussiana, define-se a relação entre a distância Euclidiana e o valor de similaridade. Assim, por meio de uma função não-linear, conseguimos modelar não-linearidades em dados.

SVMs, por utilizarem uma função linear como superfície de separação, fazem uso intenso de kernels para contornar não-linearidades. Neste modelo, o processo de treinamento precisa encontrar o parâmetro $\sigma$ que melhor se ajusta à base de dados que se deseja classificar, diminuindo o erro de teste.

A largura do kernel utilizado, caso muito pequena, pode causar *overfitting*, em que o modelo se especializa no conjunto de treinamento [9]. Nesta situação, o desempenho para o treinamento é bom, mas o modelo terá uma capacidade de generalização muito limitada, acarretando em um alto erro para o conjunto de teste. Por outro lado, ao se utilizar um valor de largura muito alto, o modelo se tornará pouco descritivo, perdendo informações específicas sobre grupos de dados.

Deve-se então encontrar o valor ótimo de largura para cada base de dados visando evitar estas duas situações, definindo o kernel que descreva bem as características do conjunto e mantenha a capacidade de generalizá-lo, descrevendo também amostras desconhecidas ao processo de treinamento.

## 3 Estimador de Densidades Kernel

Uma função de kernel também pode ser aplicada para estimar densidades de probabilidade de uma distribuição de dados. Na técnica de Estimativa de Densidades Kernel (KDE), atribuindo-se uma função Gaussiana d-dimensional com centro em cada amostra, podemos combiná-las de forma a sintetizar uma função densidade de probabilidade para o grupo de amostras [6].

Assumindo independência e o mesmo raio $\sigma$ para todas as dimensões, a estimativa de densidade pelo KDE Gaussiano em um determinado ponto arbitrário $\mathbf{x}_i$ pode ser obtida por meio da soma dos produtos acumulados em todas as dimensões para todos os padrões do conjunto de amostras, conforme a Equação 2 a seguir:

$$\hat{f}_h(\mathbf{x}_i) = \frac{1}{N}\sum_{k=1}^{N}\left\{\prod_{j=1}^{d}\frac{1}{\sigma}e^{-\left(\frac{x_{ij}-x_{kj}}{\sigma}\right)^2}\right\} \tag{2}$$

onde $\hat{f}_h(x)$ é o valor da função de densidade em $\mathbf{x}$, $N$ é o tamanho da amostra e $d$ o número de dimensões de entrada.

O produtório da Equação 2 pode ser escrito como

$$\frac{1}{\sigma^d} e^{-\frac{(\mathbf{x}_i - \mathbf{x}_k)^2}{\sigma^2}}$$

Assim, a Equação 2 pode ser reescrita como na Equação 3.

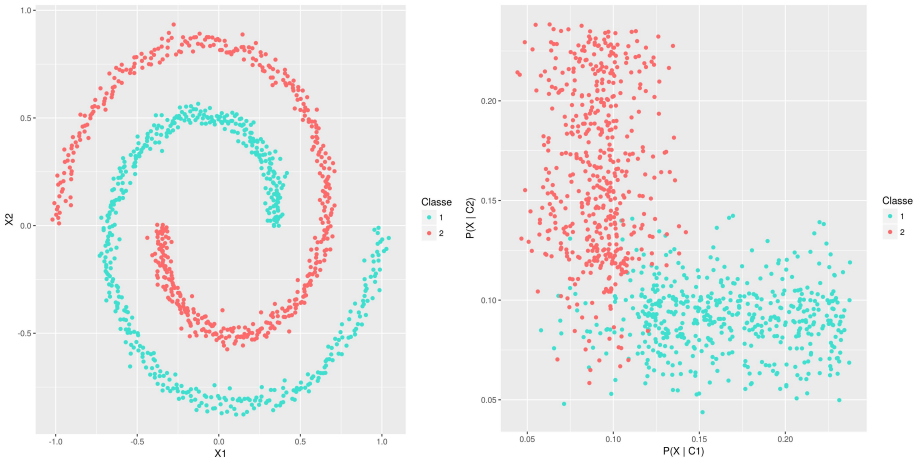
$$\hat{f}_h(\mathbf{x}_i) = \frac{1}{N\sigma^d} \sum_{k=1}^{N} e^{-\frac{(\mathbf{x}_i - \mathbf{x}_k)^2}{\sigma^2}} \tag{3}$$

O somatório da Equação 3 corresponde à soma de todos os elementos de uma linha (ou coluna) da matriz de kernel Gaussiano $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_k) = [k_{ij}]$ com raio $\sigma$, podendo a mesma ser reescrita na forma da Equação 4 [6].

$$\hat{f}_h(\mathbf{x}_i) = \frac{1}{N\sigma^d} \sum_{k=1}^{N} \mathbf{K}(\mathbf{x}_i, \mathbf{x}_k) \tag{4}$$

Assim, para se obter o valor da função de densidade estimada pelo KDE com função de kernel Gaussiana em um ponto arbitrário $\mathbf{x}_i$, basta somar todas as colunas da matriz de kernel Gaussiana com parâmetro $\sigma$ na linha $i$ e dividir pelo produto $N\sigma^d$. Portanto, assumindo-se independência entre as variáveis de entrada e um único raio $\sigma$ para todas as dimensões, a densidade estimada pelo KDE para um padrão arbitrário $\mathbf{x}_i$ pode ser obtida diretamente a partir da matriz de kernel $\mathbf{K}$. O problema de estimativa da densidade $\hat{f}_h(\mathbf{x}_i)$ segundo a Equação 4 se resume a encontrar então o valor de $\sigma$ que satisfaça a alguma restrição ou função-objetivo.

Em problemas de classificação, uma vez estimadas as densidades de um determinado conjunto, teremos a probabilidade das amostras pertencerem a cada classe. Com esta informação, podemos mapeá-las a um novo espaço. Neste espaço, cada eixo representa a probabilidade da amostra pertencer a cada uma das classes. Este espaço é chamado espaço de verossimilhanças, uma vez que mapeia cada amostra de acordo com a verossimilhança desta pertencer a cada classe. A Figura 1b ilustra este espaço para o problema das espirais [12] mostrado na Figura 1a, discriminando as amostras pertencentes a cada uma das classes e utilizando KDE com valor de $\sigma$ igual a $0,27$.




(a) Problema não linearmente separável no espaço de entrada

(b) Projeção dos dados no espaço de verossimilhanças, utilizando KDE com $\sigma = 0,27$

Figura 1: Exemplo de um problema não linearmente separável projetado no espaço de verossimilhanças

## 4 Máquinas de Vetores de Suporte

Dado o conjunto de dados linearmente separável $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i) \mid i = 1 \dots N\}$, onde $y_i \in \{+1, -1\}$ representa o rótulo da classe de cada padrão $\mathbf{x}_i \in \mathbb{R}^n$, um hiperplano ótimo pode ser definido através de uma função linear $f(\mathbf{x}) = \text{sgn}\left(\mathbf{w}^T\mathbf{x} + b\right)$, onde $\mathbf{w} \in \mathbb{R}^n$ é o vetor normal, $b \in \mathbb{R}$ representa o bias e $\text{sgn}(\cdot)$ é uma função sinal. A função $f(\mathbf{x})$ produz a maior margem de separação entre as amostras das classes [13, 14]. O problema de aprendizado pode ser dado como um problema de otimização, onde os parâmetros que deverão ser encontrados, $\mathbf{w}$ e $b$ respectivamente, são aqueles que maximizam a margem e asseguram que todas as amostras do conjunto de treinamento sejam corretamente classificadas. A equação do hiperplano de separação é dada como $\mathbf{w}^T\mathbf{x} + b = 0$ conforme [15], sendo que a distância entre o hiperplano de separação determinado por $(\mathbf{w}, b)$ e o padrão de entrada $\mathbf{x}$ mais próximo definem a margem de separação. A classificação dos padrões é dada por sua posição em relação ao hiperplano [15], como é mostrado a seguir

$$\begin{cases} \mathbf{w}^T\mathbf{x}_i + b \geq 0, & \mathbf{y}_i = +1 \\ \mathbf{w}^T\mathbf{x}_i + b < 0, & \mathbf{y}_i = -1 \end{cases} \tag{5}$$

Os padrões onde primeira e a segunda expressão da Equação 5 são satisfeitos com uma igualdade são chamados de vetores de suporte.

No caso em que o conjunto $\mathcal{S}$ não é linearmente separável, é introduzido um conjunto de variáveis escalares não negativas $(\xi)_{i=1}^N$, onde $N$ é o tamanho do

conjunto de dados de entrada. De acordo com [15], a definição para o hiperplano de margem flexível pode ser definida como

$$\mathbf{y}_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, ..., N, \tag{6}$$

onde $\xi$ é uma variável de folga. O objetivo do treinamento é encontrar um hiperplano que tenha o menor erro de classificação dos dados de entrada. Isso pode ser feito através da minimização da função

$$\begin{aligned} \min_{(\mathbf{w},b,\xi)} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \quad + \quad \mathbf{C}\sum_{i=1}^{N}\xi_i \\ \text{sujeito a,} \quad & \mathbf{y}_i[\mathbf{w}^T\mathbf{\Phi}(\mathbf{x}_i) + b] \quad \geq \quad 1 - \xi_i, \\ & \xi_i \quad \geq \quad 0 \end{aligned} \tag{7}$$

onde $\mathbf{\Phi}(\cdot)$ é a função de mapeamento (kernel), e $\mathbf{C}$ controla o *tradeoff* entre a complexidade da máquina e o número de padrões que podem violar a restrição imposta pela Equação 7.

## 5 Método Proposto

O método de estimativa KDE define um valor de probabilidade condicional a uma dada classe em um dado ponto calculando a média das probabilidades relacionadas a cada amostra de treinamento pertencente a esta classe. Este trabalho define um valor de semelhança de uma amostra a uma classe de forma análoga: para um ponto arbitrário $\mathbf{x}_i$, o valor de semelhança $B(\mathbf{x}_i, C_k)$ para uma dada classe $C_k$ é a média dos valores de similaridade deste ponto a todas as $N$ amostras de treinamento pertencentes a esta classe, calculados usando kernels RBF.

$$B(\mathbf{x}_i, C_k) = \frac{1}{N}\sum_{j=1}^{N} K(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{N}\sum_{j=1}^{N} \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}) \tag{8}$$

Utilizando a Equação 8, calcula-se a semelhança das amostras de treinamento às classes, mapeando-as para um espaço de semelhanças análogo ao espaço de verossimilhanças, onde cada eixo representa a semelhança de uma amostra a uma dada classe. Ao alterar a largura de base $\sigma$ do kernel aplicado, pode-se observar o comportamento das amostras mapeadas neste espaço para que se possa otimizar $\sigma$.

Com os dados mapeados, calcula-se as médias das semelhanças associadas a cada classe. Um ponto médio de uma dada classe neste espaço representa a semelhança média de uma amostra, sabidamente pertencente a esta classe, a cada uma das classes existentes, incluindo a própria. Calcula-se assim um valor de similaridade das classes entre si. A similaridade de uma classe $C_i$, com $N$ amostras, a uma classe $C_j$ é definida na Equação 9.

$$Sim(C_i, C_j) = \frac{1}{N} \sum_{k=1}^{N} B(\mathbf{x}_k, C_j) \tag{9}$$

Torna-se possível expressar os valores de similaridade de uma classe a si própria e a todas as demais como um vetor no espaço de semelhanças. Define-se $\mathbf{V_1}$ e $\mathbf{V_2}$ os vetores que descrevem as similaridades para um problema de classificação binária como

$$\mathbf{V_1} = \begin{bmatrix} Sim(C_1, C_1) \\ Sim(C_1, C_2) \end{bmatrix}$$

$$\mathbf{V_2} = \begin{bmatrix} Sim(C_2, C_1) \\ Sim(C_2, C_2) \end{bmatrix}$$

No espaço de semelhanças, define-se primeiramente a distância Euclidiana $||\mathbf{V_1} - \mathbf{V_2}||$. Esta distância possui um valor expressivo em situações onde as classes são claramente definidas, ou seja, a semelhança $B(\cdot)$ da maioria das amostras à própria classe é bem superior à semelhança destas a outras classes. Com o aumento da largura $\sigma$, a exemplo das funções densidade de probabilidade com o KDE [6], as semelhanças se tornam mais distribuídas, aproximando as classes. Assim, $\mathbf{V_1}$ e $\mathbf{V_1}$ se aproximam, levando à diminuição da distância Euclidiana. Por outro lado, em valores muito baixos de $\sigma$, cada amostra de treinamento terá apenas o valor de semelhança à própria classe enquanto a semelhança a outras classes é nula. Apesar de parecer ideal, esta situação é indesejada, pois se perde capacidade de generalização para amostras não presentes no conjunto de treinamento, caracterizando o *overfitting*.

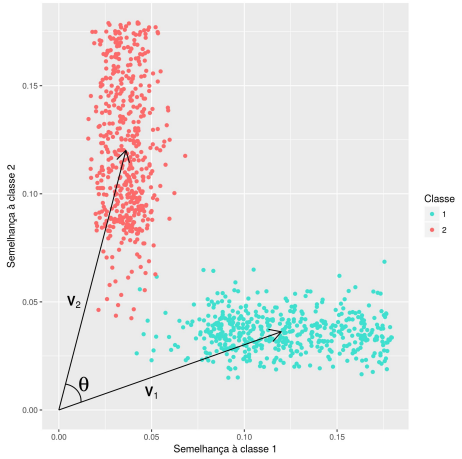



Figura 2: Vetores $\mathbf{V_1}$ e $\mathbf{V_2}$ ilustrados no espaço de semelhanças

Devido a este problema, introduz-se uma segunda medida, que consiste no cosseno do ângulo $\theta$ entre os vetores $\mathbf{V_1}$ e $\mathbf{V_2}$. $\mathbf{V_1}$, $\mathbf{V_2}$ e $\theta$ são ilustrados na Figura 2.

Quando ocorre o *overfitting*, percebe-se que o cosseno é nulo ou ao menos muito próximo de zero, já que o ângulo $\theta$ entre os vetores será muito próximo de 90°. Com o aumento da largura de base do kernel, haverá valores não-nulos de semelhança entre amostras e classes que não a própria, determinando um valor também não-nulo de $\cos\theta$. Nesta situação, o modelo ganha capacidade de generalização, admitindo a presença de novas amostras além do conjunto de treinamento.

Define-se então a função de distância $\mathbb{D}(\mathbf{V_1}, \mathbf{V_2})$ entre duas classes, calculada a partir dos pontos médios, de acordo com a Equação 10.

$$\mathbb{D}(\mathbf{V_1}, \mathbf{V_2}) = ||\mathbf{V_1} - \mathbf{V_2}|| \cdot \cos\theta \tag{10}$$

O valor da distância entre classes é, então, o produto entre os dois termos, representando a distância Euclidiana entre as médias e o cosseno entre $\mathbf{V_1}$ e $\mathbf{V_2}$.

O processo de otimização de $\sigma$ se resume a encontrar o compromisso entre os termos de separação das classes, $||\mathbf{V_1} - \mathbf{V_2}||$, e de capacidade de generalização do modelo, $\cos\theta$, maximizando a função de distância $\mathbb{D}(\mathbf{V_1}, \mathbf{V_2})$. O problema de encontrar o valor máximo do produto destes dois objetivos conflitantes pode ser interpretado como uma faceta do dilema viés-variância [3], onde se deseja maximizar a separação entre classes, mas ao mesmo tempo obter um modelo descritivo o suficiente para novas amostras.

Uma vez obtidas as similaridades, a função apresentada é computacionalmente calculada de forma eficiente: para dois vetores d-dimensionais no espaço $\mathbb{R}^d$, o cosseno entre eles é igual ao produto escalar destes vetores dividido pelo produto das normas [16].Podemos então reescrevê-la como na Equação 11.

$$\mathbb{D}(\mathbf{V_1}, \mathbf{V_2}) = \frac{\mathbf{V_1} \cdot \mathbf{V_2}}{||\mathbf{V_1}|| \cdot ||\mathbf{V_2}||} \cdot ||\mathbf{V_1} - \mathbf{V_2}|| \tag{11}$$

Os vetores $\mathbf{V_1}$ e $\mathbf{V_2}$ são dependentes apenas das semelhanças dos dados de entrada às classes. O cálculo das semelhanças para um dado conjunto de treinamento, por sua vez, recebe apenas um parâmetro: a largura da base do kernel utilizado. Assim sendo, podemos interpretar a função de distâncias $\mathbb{D}(\mathbf{V_1}, \mathbf{V_2})$ como uma função de dissimilaridade $\mathbb{S}(X, \sigma)$, sendo $\sigma$ a largura do kernel e X os dados de treinamento, juntamente aos rótulos.

Para encontrarmos, então, o valor ótimo de largura $\sigma^*$ dado um conjunto de treinamento $X$, temos um problema de otimização unidimensional, formalizado na Equação 12, onde devemos maximizar a dissimilaridade entre as classes.

$$\sigma^* = \arg\max_{\sigma} \ \mathbb{S}(X, \sigma) \tag{12}$$

Evidências baseadas nos dados utilizados neste trabalho indicam que a função $\mathbb{S}(\cdot)$ possui comportamento unimodal.

## 6 Resultados

Para validar o método proposto, serão utilizados dois classificadores SVM, utilizando kernels RBF encontrados de maneiras distintas. Neste trabalho, estes classificadores serão denominados **Classificador 1** e **Classificador 2**. No Classificador 1, ambos $\sigma$ e $\mathbf{C}$ foram otimizados via validação cruzada. Já no Classificador 2, $\sigma$ foi encontrado utilizando o método proposto neste trabalho, enquanto apenas $\mathbf{C}$ foi encontrado via validação cruzada após definido $\sigma$.

Foram utilizadas 18 bases reais. A base *"Appendicitis data set"* foi encontrada no repositório KEEL-datasets [17] e a base *"Breast Cancer Hess Probes"*, encontrada em [18]. As demais bases foram obtidas do repositório UCI [19]. Para avaliação, mediu-se acurácia de cada classificador nestes dados. Para a validação cruzada, escolheu-se os parâmetros utilizando a acurácia como métrica de referência. Em seguida, foi executado o teste estatístico de Wilcoxon para se averiguar se os classificadores são equivalentes.

As bases *"segmentation"* e *"glass"* não são, originalmente, problemas de classificação binária, de forma que foram transformadas em problemas *"1 contra todos"*, como descrito em [20].

Tabela 1: Acurácia dos classificadores

| Base | Classificador 1 | Classificador 2 |
|---|---|---|
| appendicitis | 86,818 ± 9,989 | 87,818 ± 9,727 |
| australian | 85,797 ± 4,773 | 86,377 ± 4,223 |
| banknote | 100,000 ± 0,000 | 100,000 ± 0,000 |
| breastcancer | 96,345 ± 2,414 | 96,775 ± 1,808 |
| breastHess | 82,637 ± 10,119 | 81,868 ± 10,866 |
| bupa | 68,412 ± 8,203 | 68,966 ± 11,087 |
| climate | 91,667 ± 4,880 | 95,741 ± 3,387 |
| diabetes | 76,822 ± 4,548 | 76,558 ± 4,456 |
| fertility | 88,000 ± 4,216 | 87,000 ± 4,830 |
| german | 75,600 ± 3,307 | 75,400 ± 4,142 |
| glass | 96,255 ± 2,991 | 96,234 ± 3,745 |
| haberman | 72,849 ± 7,568 | 71,903 ± 6,846 |
| heart | 83,704 ± 9,272 | 83,704 ± 8,765 |
| ILPD | 71,334 ± 6,338 | 70,629 ± 6,354 |
| ionosphere | 94,865 ± 3,518 | 94,873 ± 3,241 |
| parkinsons | 89,237 ± 7,723 | 91,237 ± 6,030 |
| segmentation | 98,095 ± 2,459 | 99,048 ± 2,008 |
| sonar | 80,810 ± 9,712 | 88,024 ± 5,558 |

### 6.1 Acurácia

Na tabela 1 estão os resultados de acurácia dos classificadores.Vemos que a difereça entre o desempenho de uma SVM com seus parâmetros calculados por ambos os métodos foi muito baixa. O método proposto no Classificador 2 teve,

inclusive, desempenho melhor que a validação cruzada em algumas bases, como *"climate"* e *"sonar"*.

### 6.2 Teste Estatístico de Wilcoxon

O teste estatístico de Wilcoxon se baseia nas diferenças entre o desempenho de dois classificadores em uma série de bases de dados. A partir dos rankings dos valores absolutos das diferenças, bem como qual classificador teve melhor performance em cada base, calcula-se o valor $Z$, como descrito em [21]. Para um nível de significância de 0,05, a hipótese nula pode ser descartada caso $Z < -1,96$ [21].

Com os valores de acurácia dos classificadores, o valor $Z$ calculado foi de $-1.0083$. Assim, verificamos que não se pode descartar a hipótese nula, o que mostra que a diferença entre os classificadores **não** é significante.

## 7 Conclusões

A abordagem alternativa proposta neste trabalho se mostrou promissora. Partindo de valores de similaridade entre amostras calculados pelo kernel RBF, definiu-se a função de semelhança de amostras a classes, chegando à medida de similaridade entre classes. Desta medida, introduziu-se a função de dissimilaridade $\mathbb{S}(\cdot)$, que pôde ser utilizada para a otimização do kernel.

Como o cálculo de similaridades é feito do grupo de treinamento para ele próprio, sem a presença de um grupo diferente para validação, há alto risco de haver *overfitting*. A medida do cosseno se mostrou uma boa solução ao problema. Como o modelo deve ser capaz de descrever dados desconhecidos, admite-se às amostras similaridades não-nulas a amostras de classes sabidamente incorretas. Assim, com a representação das classes em vetores, definiu-se a função de distância entre classes, com a qual, a partir do produto de seus dois termos, foi possível encontrar uma solução para o projeto do kernel.

Como visto nos testes executados em 18 bases de dados reais, a acurácia de um classificador SVM utilizando o método proposto foi próxima à acurácia do mesmo classificador usando validação cruzada. Este desempenho se manteve, inclusive, em bases desbalanceadas, como a base *"diabetes"*, em que o número de amostras de uma classe excede a outra em mais de 80%, e também em bases com muitas dimensões, como é o caso da base *"sonar"*, que possui 60 características. Como as similaridades entre classes são calculadas com base na média das semelhanças das amostras, há um efeito de normalização, o que explica o procedimento não se degradar em bases desbalanceadas.

Além disso, o teste estatístico de Wilcoxon mostrou que a diferença entre os classificadores para as bases utilizadas é insignificante, o que mostra coerência na utilização do método para se encontrar $\sigma$ de acordo com informação presente na disposição dos dados, não sendo necessária a estimativa baseada na performance de modelos em um conjunto de validação.

## 8 Agradecimentos

Os autores agradecem à FAPEMIG, à CAPES e à CNPq pelo suporte a este trabalho.

## Referências


[1] Vapnik, V.: The nature of statistical learning theory. Springer science & business media (2013)

[2] de Albuquerque Teixeira, R., Braga, A.P., Takahashi, R.H., Saldanha, R.R.: Improving generalization of mlps with multi-objective optimization. Neurocomputing **35**(1) (2000) 189–194

[3] Geman, S., Bienenstock, E., Doursat, R.: Neural networks and the bias/variance dilemma. Neural computation **4**(1) (1992) 1–58

[4] Hansen, P.C.: The L-curve and its use in the numerical treatment of inverse problems. IMM, Department of Mathematical Modelling, Technical Universityof Denmark (1999)

[5] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553) (2015) 436–444

[6] Silverman, B.W.: Density estimation for statistics and data analysis. Volume 26. CRC press (1986)

[7] Wang, S., Deng, Z., Chung, F.l., Hu, W.: From gaussian kernel density estimation to kernel methods. International Journal of Machine Learning and Cybernetics **4**(2) (2013) 119–137

[8] Duan, K., Keerthi, S.S., Poo, A.N.: Evaluation of simple performance measures for tuning svm hyperparameters. Neurocomputing **51** (2003) 41–59

[9] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification. John Wiley & Sons (2012)

[10] Hofmann, T., Schölkopf, B., Smola, A.J.: Kernel methods in machine learning. The annals of statistics (2008) 1171–1220

[11] Vert, J.P., Tsuda, K., Schölkopf, B.: A primer on kernel methods. Kernel Methods in Computational Biology (2004) 35–70

[12] Leisch, F., Dimitriadou, E.: mlbench: Machine Learning Benchmark Problems. (2010) R package version 2.1-1.

[13] Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: COLT '92: Proceedings of the fifth annual workshop on Computational learning theory, New York, NY, USA, ACM (1992) 144–152

[14] Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3) (1995) 273–297

[15] Haykin, S.: Neural networks and learning machines. Volume 3. Prentice Hall (2009)

[16] Noble, B., Daniel, J.W., et al.: Applied linear algebra. Volume 3. Prentice-Hall New Jersey (1988)

[17] Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic & Soft Computing **17** (2011)

[18] 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., et al.: 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

[19] Lichman, M.: UCI machine learning repository (2013)

[20] Castro, C.L., Braga, A.P.: Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. IEEE transactions on neural networks and learning systems **24**(6) (2013) 888–899

[21] Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine learning research **7**(Jan) (2006) 1–30