Princípios de Redes Neurais Artificiais e de Reconhecimento de Padrões

Prof. Antônio de Pádua Braga
Departamento de Engenharia Eletrônica
Escola de Engenharia da UFMG

17 de maio de 2017

Sumário

Capítulo 1

Máquinas de Vetores de Suporte - SVMs

1.1 Problema Primal

Para um problema de classificação binária, considere que os rótulos y_i do conjunto de dados $D_L = \{\mathbf{x}_i, y_i\}_{i=1}^N$, estejam restritos ao conjunto $\{-1, +1\}$. Considere também que o vetor arbitrário de entrada $\mathbf{x}_i$ é mapeado em um espaço intermediário por meio de p funções $\psi_j(\mathbf{x}_i, \mathbf{z}_j)$. A resposta $\hat{y}_i$ do modelo é obtida a partir da combinação linear das funções de mapeamento $\psi_j(\mathbf{x}_i, \mathbf{z}_j)$ por parâmetros w_j. Assim, o vetor $\mathbf{x}_i$ será classificado corretamente se $u_i = \sum w_j \psi_j(\mathbf{x}_i, \mathbf{z}_j) + b$ e y_i tiverem os mesmos sinais, já que a resposta $\hat{y}_i$ do modelo para a entrada $\mathbf{x}_i$ é obtida através do sinal de u_i ($\hat{y}_i = sinal(u_i)$). Assim, para que todos os vetores $\mathbf{x}_i$ sejam classificados corretamente, a desigualdade apresentada em 1.1 deve ser satisfeita para todos os N pares $(\mathbf{x}_i, y_i)$.

$$y_i \left(\sum_{j=1}^{p} w_j \psi_j(\mathbf{x}_i, \mathbf{z}_j) + b \right) \geq +1 \tag{1.1}$$

onde w_j é um elemento do vetor $\mathbf{w} = [w_1, w_2, \cdots, w_j, \cdots, w_p]^T$.

O atendimento à desigualdade da expressão 1.1 garante, portanto, que o separador linear caracterizado pelo vetor de parâmetros $\mathbf{w}$ separe corretamente os vetores das classes -1 e $+1$. No entanto, deseja-se também que este separador tenha margem de separação máxima e que, portanto, a norma $||\mathbf{w}||$ seja também minimizada. O problema de aprendizado resultante pode então ser descrito como se segue.

> *Dada a matriz* $\mathbf{Z}$*, encontre o vetor de parâmetros* $\mathbf{w}$ *que satisfaça às restrições*
>
> $$y_i(\mathbf{w}^T \psi(\mathbf{x}_i, \mathbf{Z}) + b) \geq +1 \tag{1.2}$$
>
> *para todos os* N *pares* $(\mathbf{x}_i, y_i)$ *e que tenha norma mínima descrita pelo funcional:*

$$\frac{1}{2}\mathbf{w}^T\mathbf{w} \tag{1.3}$$

Considerando que a matriz de pesos $\mathbf{Z}$ seja conhecida previamente, as restrições caracterizadas pela desigualdade 1.2 são lineares e a função de custo quadrática caracterizada pela expressão 1.3 é convexa, tendo o problema, portanto, um mínimo global garantido. A solução do problema de otimização correspondente pode então ser obtida ao se adicionar as restrições ponderadas por Multiplicadores de Lagrange [NW06] à função de custo, conforme apresentado na Equação 1.4, cujas derivadas parciais em relação aos parâmetros livres $\mathbf{w}$ e b, obtidas nas Equações 1.5 e 1.6, determinam as soluções do problema de otimização.

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i \left[y_i(\mathbf{w}^T\psi(\mathbf{x}_i, \mathbf{Z}) + b) - 1 \right] \tag{1.4}$$

$$\frac{\partial J}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \psi(\mathbf{x}_i, \mathbf{Z}) = \mathbf{0} \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \psi(\mathbf{x}_i, \mathbf{Z}) \tag{1.5}$$

$$\frac{\partial J}{\partial b} = \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{1.6}$$

Uma vez obtidas as soluções do problema, representadas pelas Equações 1.5 e 1.6, estas são então substituídas na expressão do problema primal com Multiplicadores de Lagrange (Equação 1.4), resultando na função de custo dual representada na Equação 1.7.

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \psi(\mathbf{x}_i, \mathbf{Z})^T \psi(\mathbf{x}_j, \mathbf{Z}) \tag{1.7}$$

O problema de otimização, re-escrito na sua forma dual por Multiplicadores de Lagrange pode então ser enunciado como se segue.

> *Dada a matriz* $\mathbf{Z}$*, encontre os parâmetros* α *que maximizem a função de custo*
>
> $$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \psi(\mathbf{x}_i, \mathbf{Z})^T \psi(\mathbf{x}_j, \mathbf{Z}) \tag{1.8}$$
>
> *sujeito às restrições:*
>
> - $\sum_{i=1}^{N} \alpha_i y_i = 0$
> - $\alpha_i \geq 0$

1.1.1 Considerações sobre o Mapeamento

Em SVMs, a função de custo que caracteriza o problema de otimização é descrita pela Equação 1.8, cujos parâmetros livres são apenas os Multiplicadores de Lagrange α_i que determinaram a solução dual. A solução do problema dual assume, portanto, que as funções de mapeamento $\psi(\mathbf{x}, \mathbf{Z})$ sejam conhecidas previamente. A solução do problema é, portanto, obtida para uma determinada matriz de parâmetros $\mathbf{Z}$ e funções $\psi(\cdot)$ previamente selecionadas.

1.1.2 SVMs com Variáveis de Folga

O atendimento às restrições lineares caracterizadas pela Expressão 1.2 para todo o conjunto de treinamento implica na separação linear de todas as amostras mapeadas no novo espaço definido pelas funções $\psi(\mathbf{x}, \mathbf{Z})$. No entanto, caso o mapeamento não resulte em uma separação linear, estas restrições não serão atendidas, já que algumas amostras serão mapeadas com sinal oposto ao da sua classe. Com o objetivo de permitir uma margem de erro, ou seja, que algumas amostras sejam classificadas incorretamente, a restrição linear, caracterizada pela Expressão 1.2 é modificada de forma a incorporar a variável de folga ξ_i, conforme apresentado na Expressão 1.9.

$$y_i(\mathbf{w}^T\psi(\mathbf{x}_i, \mathbf{Z}) + b) + \xi_i \geq +1 \tag{1.9}$$

onde $\xi_i \geq 0$ é a variável de folga associada ao par $(\mathbf{x}_i, y_i)$.

Uma nova função de custo é obtida através da combinação linear da função de custo original do problema primal (Equação 1.3) com um termo associado à variável de folga ξ_i, o qual é ponderado por uma termo de penalização, ou regularização, $C \geq 0$, conforme descrito na Equação 1.10.

$$Q(\mathbf{w}, \xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \tag{1.10}$$

O problema primal original é então re-definido a seguir de forma a incorporar as variáveis de folga ξ e o termo de regularização C.

> *Dados o parâmetro C e a matriz $\mathbf{Z}$, encontre o vetor de parâmetros $\mathbf{w}$ que satisfaça às restrições*
>
> $$y_i(\mathbf{w}^T\psi(\mathbf{x}_i, \mathbf{Z}) + b) + \xi_i \geq +1 \tag{1.11}$$
>
> *para todos os N pares $(\mathbf{x}_i, y_i)$ e cujo funcional seguinte seja mínimo:*
>
> $$\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \tag{1.12}$$

De maneira análoga aos procedimentos anteriores, obtém-se a expressão dual da Equação 1.12 por Multiplicadores de Lagrange a qual é apresentada na Equação 1.13.

$$Q(\mathbf{w}, b, \alpha, \xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left[y_i(\mathbf{w}^T\psi(\mathbf{x}_i, \mathbf{Z}) + b) + \xi_i - 1\right] \tag{1.13}$$

Obtendo-se novamente as derivadas parciais de $Q(\mathbf{w}, b, \alpha, \xi)$ em relação a $\mathbf{w}$, b e ξ_i chega-se às Equações 1.14, 1.15 e 1.16 que são apresentadas a seguir.

$$\frac{\partial Q}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \psi(\mathbf{x}_i, \mathbf{Z}) = \mathbf{0} \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \psi(\mathbf{x}_i, \mathbf{Z}) \tag{1.14}$$

$$\frac{\partial Q}{\partial b} = \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{1.15}$$

$$\frac{\partial J}{\partial \xi_i} = C - \alpha_i = 0 \quad \Rightarrow \quad \alpha_i = C \tag{1.16}$$

Visando a obter uma expressão para a função de custo dual dependente somente dos Multiplicadores de Lagrange α_i, as soluções encontradas nas Equações 1.14, 1.15 e 1.16 são então substituídas na expressão dual original (Equação 1.14) o que leva à expressão dual que é apresentada na Equação 1.17.

$$W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \psi(\mathbf{x}_i, \mathbf{Z})^T \psi(\mathbf{x}_j, \mathbf{Z}) \tag{1.17}$$

O problema de aprendizado de SVMs, formulado agora com variáveis de folga, pode se re-escrito conforme apresentado a seguir.

> *Dados o parâmetro C e a matriz $\mathbf{Z}$, encontre Multiplicadores de Lagrange α_i que maximizem a função*
>
> $$W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \psi(\mathbf{x}_i, \mathbf{Z})^T \psi(\mathbf{x}_j, \mathbf{Z}) \tag{1.18}$$
>
> *e que satisfaçam às seguintes restrições:*
>
> - $\sum_{i=1}^{N} \alpha_i y_i = 0$
> - $0 \leq \alpha_i \leq C$

1.1.3 Mapeamento Implícito com Funções de Kernel

O mapeamento explícito das amostras de entrada no espaço formado pelos produtos $\psi(\mathbf{x}_i, \mathbf{Z})^T \psi(\mathbf{x}_j, \mathbf{Z})$ depende de a função $\psi(\mathbf{x}_i, \mathbf{Z})$ ser conhecida previamente. Como na prática isto não ocorre, a construção de SVMs é realizada por meio do mapeamento implícito com funções de kernel, em que os produtos $\psi(\mathbf{x}_i, \mathbf{Z})^T \psi(\mathbf{x}_j, \mathbf{Z})$ são substituídos por funções de kernel $K(\mathbf{x}_i, \mathbf{x}_j, \mathbf{Z})$, que representam os produtos internos de kernel[1] entre os vetores $\mathbf{x}_i$ e $\mathbf{x}_j$, conforme apresentado na Equação 1.19.

$$K(\mathbf{x}_i, \mathbf{x}_j, \mathbf{Z}) = \psi(\mathbf{x}_i, \mathbf{Z})^T \psi(\mathbf{x}_j, \mathbf{Z}) \tag{1.19}$$

[1] Do Inglês *kernel inner product.*

A substituição de $\psi(\mathbf{x}_i, \mathbf{Z})^T \psi(\mathbf{x}_j, \mathbf{Z})$ por $K(\mathbf{x}_i, \mathbf{x}_j, \mathbf{Z})$ é muitas vezes chamado na literatura de "truque do kernel" [CST00]. Para que esta substituição possa ser realizada, a função de kernel deve atender às condições de Mercer [CST00], que garante que $K(\mathbf{x}_i, \mathbf{x}_j, \mathbf{Z})$ seja equivalente a $\psi(\mathbf{x}_i, \mathbf{Z})^T \psi(\mathbf{x}_j, \mathbf{Z})$. Estas condições são basicamente que a matriz $\mathbf{K}$ deve ser simétrica e não-negativa, ou seja, que seus autovalores sejam não-negativos. Satisfeitas estas condições, o mapeamento implícito através de funções de kernel pode ser realizado. Assim, a formulação final do problema de aprendizado de SVMs pode então ser finalmente apresentada, considerando-se agora a substituição do produto cruzado das funções de mapeamento pelas funções de kernel equivalentes.

Dados o parâmetro C e a matriz $\mathbf{Z}$, encontre Multiplicadores de Lagrange α_i que maximizem a função

$$W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j, \mathbf{Z}) \qquad (1.20)$$

e que satisfaçam às seguintes restrições:

- $\sum_{i=1}^{N} \alpha_i y_i = 0$
- $0 \leq \alpha_i \leq C$

1.1.4 Cálculo da saída de SVMs

A saída de uma SVM para um determinado vetor arbitrário $\mathbf{x}$ é obtida de acordo com a Equação 1.21.

$$\hat{y} = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}, \mathbf{Z}) \qquad (1.21)$$

1.1.5 Comentários sobre o Aprendizado de SVMs

A formulação final de uma SVM com margens flexíveis conforme descrita na seção anterior possui função de custo convexa e restrições lineares, o que garante a existência de um máximo global. No entanto, esta formulação se refere apenas à solução do problema de separação linear já que ela assume o conhecimento do mapeamento $K(\mathbf{x}_i, \mathbf{x}_j, \mathbf{Z})$, o que muitas vezes não é enfatizado quando se analisa somente a solução por Multiplicadores de Lagrange para o problema linear. Por esta razão, neste texto adotou-se a representação $K(\mathbf{x}_i, \mathbf{x}_j, \mathbf{Z})$ para o kernel, deixando assim explícita a dependência da matriz de parâmetros $\mathbf{Z}$ que, na verdade, determina o mapeamento não-linear para o espaço de características onde o problema linear é então resolvido. Talvez esta seja uma das diferenças mais fundamentais entre as SVMs e as RNAs, já que a matriz $\mathbf{Z}$ faz parte do problema de otimização em RNAs, sendo obtida, portanto, como parte do processo de aprendizado. É claro que a dimensão da matriz $\mathbf{Z}$ em RNAs é muito maior do que em SVMs, podendo, em SVMs se reduzir a apenas um escalar para o caso de kernels Gaussianos, por exemplo. Não obstante, mesmo este valor escalar deve ser fornecido a priori para que a função de custo seja

computada, o que demanda um processo interativo de busca por parâmetros satisfatórios durante o processo de treinamento de SVMs.

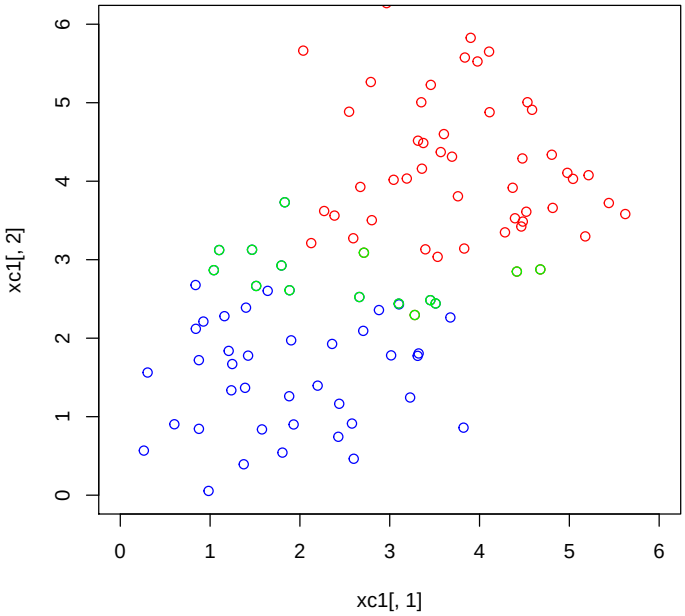
Além dos parâmetros **Z** da matriz de kernel, o usuário deve também arbitrar o valor do termo de regularização C que representa a ponderação entre os dois termos da função de custo do problema primal e que aparece como restrição do limite superior dos Multiplicadores de Lagrange no problema dual. Analogamente aos parâmetros da função de kernel, para que o problema de otimização de SVMs seja, enfim, resolvido, o parâmetro C deve ser também fornecido pelo usuário. Considerando o caso mais simples em que o kernel é determinado por apenas um escalar r, como no caso de um kernel Gaussiano, o problema de treinamento de SVMs se resume então a encontrar os parâmetros (r, C) e os Multiplicadores de Lagrange $\alpha_i, i = 1 \cdots N$ que satisfaçam uma determinada função de avaliação. Para cada par de valores (r, C) fornecidos pelo usuário, os Multiplicadores de Lagrange α_i ótimos são então encontrados e a resposta da SVM correspondente é então avaliada através de uma função de avaliação pré-estabelecida. Esta função de avaliação pode ser simplesmente definida como o erro obtido sobre um conjunto de validação, por exemplo. Desta forma, para que haja uma maior confiabilidade na escolha dos parâmetros (r, C), o usuário deve executar uma busca exaustiva no espaço de parâmetros (r, C) e, para cada par de valores destes parâmetros, avaliar a SVM correspondente, selecionando aquele modelo que tenha melhor resposta à função de avaliação. Como a busca pelos parâmetros (r, C) na forma de *grid search* deve ser discreta, não há garantias de que o mínimo global da função de avaliação seja atingido.

1.2 Exemplo em R

```
> rm(list=ls())
> library("kernlab")
> xc1=replicate(2, rnorm(50)+4)
> xc2=replicate(2, rnorm(50)+2)
> xin=rbind(xc1,xc2)
> yin=rbind(matrix(-1,50,1),matrix(1,50,1))
> plot(xc1[,1],xc1[,2],col='red',type="p",xlim=c(0,6),ylim=c(0,6))
> points(xc2[,1],xc2[,2],col='blue')
> svmtrein <- ksvm(xin,yin,type='C-bsvc',kernel='vanilladot',C=10)


 Setting default kernel parameters


> yhat<-predict(svmtrein,xin,type="response")
> a=alpha(svmtrein)
> ai=SVindex(svmtrein)
> nsvec=nSV(svmtrein)
> points(xin[ai,1],xin[ai,2],col="green")
```


xc1[, 2]
xc1[, 1]
0
1
2
3
4
5
6

Referências Bibliográficas

[CST00] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[NW06] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.