

Máquinas de Vetores Suporte em Otimização Sem Derivadas

Adriano Verdério

PPGMA - Programa de Pós-graduação em Matemática - UFPR
CEM - Centro de Engenharias da Mobilidade - UFSC
Campus Joinville
89218-000, Joinville, SC
E-mail: adriano.verderio@ufsc.br

Elizabeth W. Karas, Lucas G. Pedroso.

Departamento de Matemática - UFPR
Centro Politécnico - Jardim das Américas
81531-980 Caixa Postal 19081, Curitiba, PR
E-mail: ewkaras@ufpr.br, lucaspedroso@ufpr.br.

Resumo: *As Máquinas de Vetores suporte são uma classe de algoritmos de Aprendizagem de Máquinas motivados por resultados da Teoria de Aprendizagem Estatística. No início, foram usadas para a classificação de padrões e posteriormente estendidas para a regressão de funções. De um certo modo é uma generalização das técnicas usuais de regressão. Nosso objetivo é utilizá-las para aproximação de funções as quais temos conhecimento limitado, não conseguindo por exemplo calcular derivadas.*

Palavras-chave: *Máquinas Vetores Suporte, Otimização sem Derivadas*

1 Introdução

Considere o problema de programação não linear

$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeita a} & x \in \Omega, \end{array} \quad (1)$$

com $\Omega \subset \mathbb{R}^n$ não vazio, fechado, convexo e $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função continuamente diferenciável.

Conejo et al. em [1] propõem um algoritmo globalmente convergente de região de confiança para resolver o problema (1) e que gera uma sequência de minimizadores aproximados para os subproblemas restritos. O algoritmo permite grande liberdade nas construções e resoluções dos subproblemas, e o caso de interesse é quando as derivadas da função objetivo não estão disponíveis embora é suposto que existam. Problemas desse tipo são conhecidos como problemas de otimização sem derivadas, que podem surgir quando queremos otimizar uma função que venha de uma simulação, por exemplo.

Um algoritmo de região de confiança define a cada iteração um modelo da função objetivo e uma região em torno do ponto corrente na qual acreditamos que o modelo é confiável. Calculamos então um minimizador aproximado do modelo na região de confiança. Caso este ponto forneça uma redução razoável no valor da função objetivo aceitamos o iterando repetimos o processo. Caso contrário, pode ser que o modelo não represente adequadamente a função. Neste caso, o ponto é recusado e reduzimos o tamanho da região para encontrar um novo minimizador

No algoritmo proposto em [1] qualquer modelo pode ser utilizado desde que forneça uma aproximação suficientemente precisa da função objetivo. Nosso objetivo é mostrar que os modelos construídos por Máquinas de Vetores Suporte aproximam bem funções na região de confiança, uma hipótese necessária para a convergência.

2 Máquinas de Vetores Suporte

Sejam \mathcal{X} e \mathcal{Y} subconjuntos de espaços vetoriais normados, normalmente \mathbb{R}^n e \mathbb{R} , respectivamente. Suponha que é dado algum conjunto de entradas $X = \{x^1, x^2, \dots, x^m\} \subset \mathcal{X}$ e um conjunto $Y = \{y^1, y^2, \dots, y^m\} \subset \mathcal{Y}$ de forma que sejam independentes e identicamente distribuídos de acordo com alguma medida de probabilidade $d\mathbb{P}(x, y)$. Vamos supor que os pontos de Y são as imagens dos pontos de X para a função f que gostaríamos de minimizar, ou seja, $f(x^i) = y^i$. Problemas desse tipo podem ser encontrados em simulações ou para um programa ao qual não temos acesso ao código fonte.

Chamaremos X o conjunto de amostra e Y o conjunto de rótulos. As Máquinas de Vetores Suporte para regressão são construídas para encontrar uma função $h : \mathcal{X} \rightarrow \mathcal{Y}$ que aproxima bem os dados de amostra, ou seja, $h(x^i) \approx y^i$ para $(x^i, y^i) \in X \times Y$, chamaremos essa função h de preditor. Também esperamos que nosso preditor h seja uma boa aproximação para a função f .

Primeiramente, precisamos encontrar um preditor que apresenta o menor erro e para tal usaremos uma medida para avaliar sua performance. Uma maneira natural de mensurar o erro cometido é através de uma função perda, que mede o erro que um preditor comete no conjunto de amostra.

Definição 1. *Seja a tripla $(x, y, h(x)) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$ que consiste de um padrão x , um rótulo y e a predição $h(x)$. Então a aplicação $\ell : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \rightarrow [0, \infty)$ com a propriedade $\ell(x, y, y) = 0$ para todo $x \in \mathcal{X}$ e $y \in \mathcal{Y}$ é chamada de função perda.*

Uma vez definida a função perda a ser utilizada, conseguimos determinar como os erros são penalizados em cada ponto da amostra. Precisamos agora encontrar uma maneira de combinar essas penalidades locais e conseguirmos avaliar a qualidade de um preditor.

Como os dados utilizados são independentes e identicamente distribuídos de acordo com alguma medida de probabilidade $\mathbb{P}(x, y)$, o valor esperado da função perda é

$$R[h] = \mathbb{E}[\ell(x, y, h(x))] = \int_{\mathcal{X} \times \mathcal{Y}} \ell(x, y, h(x)) d\mathbb{P}(x, y),$$

que é chamado de risco esperado, ou simplesmente risco. Minimizando o risco de um preditor, encontramos o melhor candidato a aproximar nossos dados.

No entanto, minimizar o risco esperado de um preditor é impossível, pois não conhecemos a medida de probabilidade $\mathbb{P}(x, y)$. Para resolver este problema, Vapnik em sua Teoria de Aprendizagem Estatística [10] desenvolveu o Princípio Indutivo da Minimização do Risco Empírico, no qual o Risco é determinado pelo conjunto de amostra.

O Princípio Indutivo da Minimização do Risco Empírico pode ser descrito como

1. Substituir o Risco Esperado pelo Risco Empírico

$$R_{emp}[h] = \frac{1}{m} \sum_{i=1}^m \ell(x_i, y_i, h(x_i)),$$

que é a perda média encontrada no conjunto de amostra.

2. Utilizar como preditor a aplicação h que minimiza o Risco Empírico.

Apesar de em um primeiro momento parecer que o Princípio Indutivo da Minimização do Risco Empírico resolve o problema, essa técnica sozinha não é suficiente. Para um método de reconhecimento de padrões ser eficaz, deve apresentar pelo menos duas características: boa capacidade de *generalização*, permitindo que dados semelhantes sejam igualmente classificados; boa capacidade de *discriminação*, que assegura a correta separação entre as classes. Minimizar o risco empírico pode gerar instabilidades numéricas ou ainda não alcançar uma boa generalização.

Segundo Schölkopf e Smola [6], uma maneira de evitar esses problemas é restringindo a classe de soluções possíveis a um conjunto compacto. Essa técnica foi introduzida por Tykhonov e Arsenin [8] para resolver problemas inversos e tem sido aplicada em problemas de aprendizagem com bastante sucesso, trabalhando com a função risco regularizada.

Em geral, adicionamos um termo de estabilização $\Omega[f]$ à função objetivo original, que em nosso caso é o risco empírico $R_{emp}[f]$. Ou seja, consideramos a seguinte classe de risco regularizado

$$R_{reg}[f] := R_{emp}[f] + \lambda\Omega[f],$$

em que $\lambda > 0$ é o parametro regularizador que especifica o balanço entre a minimização do risco empírico e a simplicidade do nosso classificador que é alcançado com um $\Omega[f]$ pequeno.

Vapnik [10] concebeu a chamada função perda ε -insensível

$$\ell(x, y, h(x)) = |y - h(x)|_\varepsilon = \max\{0, |y - h(x)| - \varepsilon\},$$

a qual não penaliza erros menores que uma tolerância $\varepsilon > 0$ escolhida previamente. Seu algoritmo, ε -SVR (ε -Support Vector Regression), procura estimar funções por um preditor

$$h(x) = w^\top x + b \quad w, x \in \mathbb{R}^n, b \in \mathbb{R}$$

baseado nos dados X e Y .

Assim o problema é minimizar o risco empírico regularizado, ou seja, queremos minimizar

$$\frac{1}{2}\|w\|^2 + C \sum_{i=1}^m |y_i - h(x_i)|_\varepsilon \tag{2}$$

onde $\frac{1}{2}\|w\|^2$ é o termo regularizador e C é o parâmetro de regularização.

A interpretação geométrica de utilizar o regularizador $\frac{1}{2}\|w\|^2$ é encontrar a função f mais achatada possível com suficiente qualidade na aproximação, além de capturar a ideia principal da Teoria de Aprendizagem estatística de tentar obter um risco pequeno controlando tanto o erro como a complexidade do modelo [6].

Desta maneira, para encontrarmos um preditor linear iremos resolver o problema

$$\min \frac{1}{2}\|w\| + C \frac{1}{m} \sum_{i=1}^m |x^i - h(x^i)|_\varepsilon,$$

que é equivalente a resolver o problema

$$\begin{aligned} \min \quad & \frac{1}{2}\|w\|^2 + C \frac{1}{m} \sum_{i=1}^m (\xi_i + \xi'_i) \\ \text{s.a.} \quad & w^\top x_i + b - y_i \leq \varepsilon + \xi_i \\ & y_i - w^\top x_i + b \leq \varepsilon + \xi'_i \\ & \xi_i, \xi'_i \geq 0, \end{aligned}$$

que por sua vez é um problema quadrático convexo.

Seja

$$P^\top = \begin{bmatrix} (x^1)^\top \\ (x^2)^\top \\ \vdots \\ (x^m)^\top \end{bmatrix}, \quad Q = \begin{bmatrix} PP^\top & -PP^\top \\ -PP^\top & PP^\top \end{bmatrix}, \quad z = \begin{bmatrix} \alpha \\ \gamma \end{bmatrix},$$

$$v = \begin{bmatrix} -f(P) + \varepsilon e \\ f(P) - \varepsilon e \end{bmatrix} \quad \text{e} \quad A = [-e^\top, e^\top],$$

o problema dual pode ser escrito como

$$\begin{aligned} \min \quad & z^\top Qz + v^\top z \\ \text{s.a.} \quad & Az = 0 \\ & 0 \leq z \leq C. \end{aligned}$$

Para construir um modelo não linear, podemos levar nossos dados em um espaço de dimensão maior por meio de uma aplicação $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^q$, com $q > n$ e construir um modelo linear nesse espaço. Desse modo, queremos encontrar o modelo definido por $q(x) = w^\top \varphi(x) + b$ tal que

$$|y^i - (w^\top \varphi(x^i) + b)| \leq \varepsilon \quad \forall i = 1, \dots, m.$$

Ou seja, precisamos resolver exatamente o mesmo problema que no caso linear, exceto que agora os elementos de Q são definidos por $\varphi(x^i)^\top \varphi(x^j)$ em vez de $(x^i)^\top x^j$.

Para o caso em que queremos modelos quadráticos, definimos

$$\phi(x) = (x_1^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \dots, x_2^2, \dots, \sqrt{2}x_{n-1}x_n, x_n^2, x_1, \dots, x_n),$$

de modo que

$$\phi(x^i)^\top \phi(x^j) = (x^i)^\top x^j + ((x^i)^\top x^j)^2,$$

o que facilita a construção da matriz Q do nosso problema de programação quadrática.

Assim, para construirmos modelos de uma função avaliada em alguns pontos, precisamos resolver um problema do tipo

$$\begin{aligned} \min \quad & z^\top Qz + v^\top z \\ \text{s.a.} \quad & Az = 0 \\ & 0 \leq z \leq C \end{aligned}$$

com Q simétrica e semidefinida positiva.

3 Construção dos Modelos

Para construção de modelos de uma função avaliada em alguns pontos precisamos de um controle na geometria dos pontos de amostra, de modo que não formem um espaço afim. Este controle é feito geralmente controlando o posicionamento dos pontos da amostra. Para as Máquinas de Vetores Suporte a geometria é controlada com a definição abaixo.

Definição 2 (Λ -posicionamento Modelos Lineares). *Seja $P = \{x^0, \dots, x^n\} \subset \mathbb{R}^n$. Dizemos que P é Λ -posicionado em $B(x^0, \Delta)$, para uma constante $\Lambda > 0$, se para todo $x \in B(x^0, \Delta)$,*

$$x - x^0 = \sum_{i=1}^m \lambda_i (x^i - x^0) \quad \text{com} \quad |\lambda_i| \leq \Lambda, i = 1, 2, \dots, n.$$

Com essa definição conseguimos provar que podemos construir um modelo que aproxima bem a função original.

Teorema 1. *Sejam $P = \{x^0, x^1, \dots, x^n\}$ um conjunto Λ -posicionado em $B(x^0, \Delta)$ e m o modelo linear construído por máquina de vetores suporte com margem $\varepsilon \leq \Delta_k$ e com folgas ξ e ξ^l nulas. Então existe constante $\kappa_1 > 0$ tal que*

$$|f(x) - m(x)| \leq \kappa_1 \Delta^2$$

para todo $x \in B(x^0, \Delta)$.

Para a construção de modelos quadráticos precisamos de algumas modificações na definição de posicionamento.

Definição 3 (Λ -posicionamento Modelos Não Lineares). *Seja $P = \{x^0, \dots, x^q\} \subset \mathbb{R}^n$. Dizemos que P é Λ -posicionado com respeito a aplicação φ , para uma constante $\Lambda > 0$, se para todo $x \in B(x^0, \Delta)$,*

$$\varphi(x) - \varphi(x^0) = \sum_{i=1}^m \lambda_i (\varphi(x^i) - \varphi(x^0)) \quad \text{com} \quad |\lambda_i| \leq \Lambda, i = 1, 2, \dots, q.$$

Teorema 2. *Sejam $P = \{x^0, x^1, \dots, x^q\}$ um conjunto Λ -posicionado em $B(x^0, \Delta)$ com respeito a aplicação φ e m o modelo quadrático construído por máquina de vetores suporte com margem $\varepsilon \leq \Delta^2$ e com folgas ξ e ξ' nulas. Então existe constante $\kappa_2 > 0$ tal que*

$$|f(x) - m(x)| \leq \kappa_2 \Delta^2$$

para todo $x \in B(x^0, \Delta)$.

A seguir apresentamos modelos construídos por Máquinas de Vetores Suporte para a função de Rosenbrock

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2.$$

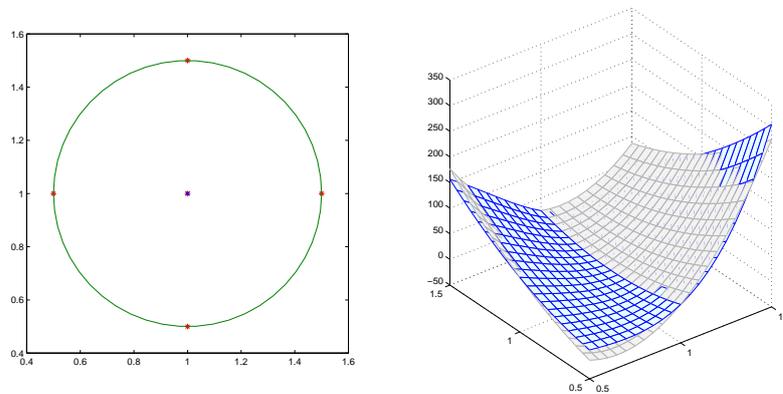


Figura 1: Modelo da Função de Rosenbrock próximo do ponto (1, 1), com raio = 0.5

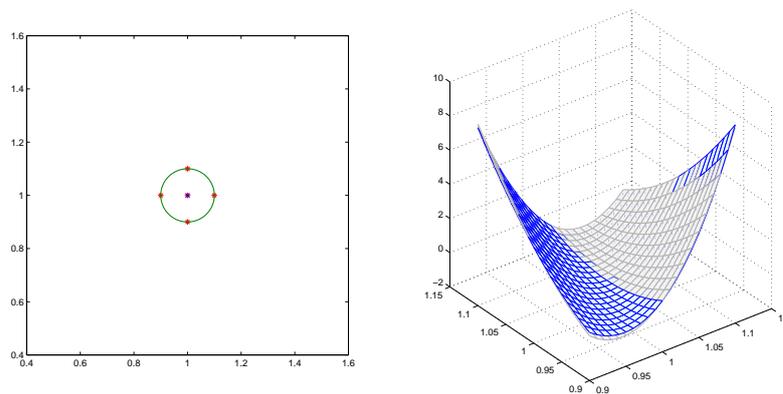


Figura 2: Modelo da Função de Rosenbrock próximo do ponto (1, 1), com raio = 0.1

No lado esquerdo de cada figura apresentamos a região de confiança e os pontos da amostra. Do lado direito apresentamos a função original em azul e o modelo em cinza.

4 Conclusão

As máquinas de vetores suporte apresentam características importantes para a aproximação de funções. Para a construção de um modelo com a técnica precisamos resolver um problema de programação quadrática. Algumas vantagens em se utilizar as máquinas de vetores suporte em otimização sem derivadas reside na quantidade de pontos na amostra, pois a técnica permite a construção de modelos conhecendo apenas um ponto da função original. Essa é uma vantagem relevante quando temos um alto custo computacional para avaliar funções a serem minimizadas.

A implementação do método de região de confiança sem derivadas com modelos construídos por máquinas de vetores suporte é a próxima etapa natural do trabalho. Além de consolidar a teoria que garante a convergência do método.

Referências

- [1] P. D. Conejo, E. W. Karas, L. G. Pedroso, A. A. Ribeiro, e M. Sachine. Global convergence of trust-region algorithms for convex minimization without derivatives. *Applied Mathematics and Computation*, 220:324–330, 2013.
- [2] A. R. Conn, N. I. Gould, e Ph. L. Toint. *Trust-Region Methods*. Society for Industrial and Applied Mathematics, Philadelphia, 2000.
- [3] A. R. Conn, K. Scheinberg, e L. N. Vicente. *Intoduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, 2009.
- [4] J. Nocedal e S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.
- [5] A. A. Ribeiro e E. W. Karas. *Otimização Contínua: Aspectos Teóricos e Computacionais*. Cengage Learning, São Paulo, 2013.
- [6] B. Schölkopf e A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. The MIT Press, Cambridge, 2002.
- [7] A. J. Smola e B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222, 2004.
- [8] A. N. Tikhonov e V. Y. Arsenin. *Solution of Ill-Posed Problems*. V. H. Winston & Sons, Washington, 1977.
- [9] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.
- [10] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 2000.