SVM: o Problema de Otimização, as Garantias de Aprendizado e o Truque do Kernel

Alek Fröhlich & Douglas Soares Gonçalves

Universidade Federal de Santa Catarina (UFSC)

alek.frohlich@posgrad.ufsc.br, douglas@mtm.ufsc.br

Resumo

Support Vector Machine (SVM) é um dos modelos mais fascinantes dentro do contexto de Aprendizagem de Máquina, usufruindo de fortes garantias teóricas e de boa performance prática. Neste trabalho, coletamos resultados teóricos instigantes à respeito do treino e do uso de SVMs.

Introdução

O SVM foi introduzido na década de 90 por Vapnik e Cortes [2] para lidar com o problema de classificação. De forma simplificada, o problema consiste em: dada uma amostra $\boldsymbol{S} = (\mathrm{x}_1, \boldsymbol{y}_1), (\mathrm{x}_2, \boldsymbol{y}_2), ..., (\mathrm{x}_m, \boldsymbol{y}_m)$ gerada de maneira i.i.d. a partir de uma distribuição $\boldsymbol{\mathcal{D}}$ sobre $\mathbb{R}^n \times \{\pm 1\}$, determinar uma função de classificação $\boldsymbol{h} : \mathbb{R}^n \to \{\pm 1\}$ que minimize o erro esperado

$$\boldsymbol{R}(\boldsymbol{h}) = \boldsymbol{P}[\boldsymbol{h}(\mathrm{x}) \neq \boldsymbol{y}]. \tag{1}$$

Como não conheçemos $\boldsymbol{\mathcal{D}}$, na prática utilizamos alguma forma de erro empírico, por exemplo, $\hat{\boldsymbol{R}}_{\boldsymbol{S}}(\boldsymbol{h}) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{h(\mathrm{x}_i)\neq y_i}$ e buscamos relacionar $\boldsymbol{R}(\boldsymbol{h})$ e $\hat{\boldsymbol{R}}_{\boldsymbol{S}}(\boldsymbol{h})$. Na prática de aprendizagem de máquina é comum buscar por $\boldsymbol{h}$ dentro de uma classe $\boldsymbol{\mathcal{H}}$. No caso do SVM, a classe corresponde aos classificadores lineares:

$$\boldsymbol{\mathcal{H}} = \left\{\mathrm{x} \mapsto \mathbf{sign}(\mathrm{w}^T\mathrm{x} + \boldsymbol{b}) : \mathrm{w} \in \mathbb{R}^n, \boldsymbol{b} \in \mathbb{R}\right\}. \tag{2}$$

Dada uma amostra $\boldsymbol{S}$, o algoritmo de treinamento do SVM busca encontrar um hiperplano $(\mathrm{w}, \boldsymbol{b})$ que minimize o erro de treino e maximize o tamanho da margem. O segundo objetivo é um diferencial do método e contribui para suas garantias de aprendizado.

Treinando SVMs

O problema de treinamento pode ser expresso como o problema de otimização a seguir:

$$\begin{aligned} \min_{\mathrm{w},b,\xi} \quad & \frac{1}{2}\mathrm{w}^T\mathrm{w} + C\sum_{i=1}^{m}\boldsymbol{\xi}_i \\ \text{s.a.} \quad & \boldsymbol{y}_i(\mathrm{w}^T\mathrm{x}_i + \boldsymbol{b}) \geq 1 - \boldsymbol{\xi}_i, \\ & \boldsymbol{\xi}_i \geq 0, \end{aligned} \tag{3}$$

em que $\boldsymbol{\xi}$ representa as violações da margem e $\boldsymbol{C}$ controla a penalização. O problema é um programa quadrático convexo (CQP) e sempre admite solução, a qual nem sempre é única [1]. As condições de otimalidade para o problema garantem que $\mathrm{w} = \sum_{i=1}^{m}\boldsymbol{\alpha}_i\boldsymbol{y}_i\mathrm{x}_i$ e $\boldsymbol{\alpha}^T\mathrm{y} = 0$, isso somado à dualidade forte nos permite expressar o problema na sua forma dual:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\boldsymbol{\alpha}_i\boldsymbol{\alpha}_j\boldsymbol{y}_i\boldsymbol{y}_j(\mathrm{x}_i^T\mathrm{x}_j) - \sum_{i=1}^{m}\boldsymbol{\alpha}_i \\ \text{s.a.} \quad & 0 \leq \boldsymbol{\alpha} \leq \boldsymbol{C}, \\ & \boldsymbol{\alpha}^T\mathrm{y} = 0. \end{aligned} \tag{4}$$

O dual também é um CQP com solução (domínio compacto). Essa é a versão típicamente resolvida, uma vez que permite a troca de $\mathrm{x}_i^T\mathrm{x}_j$ por $\boldsymbol{K}(\mathrm{x}_i, \mathrm{x}_j)$, em que $\boldsymbol{K}$ é um kernel. Os métodos comumente utilizados para resolver (4) são baseados no algoritmo SMO de Platt [6]. O algoritmo resolve uma sequência de subproblemas, nos quais apenas duas variáveis são otimizadas por vez $(\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j)$. Essa abordagem escala melhor do que métodos de programação quadrática usuais, uma vez que o número de variáveis cresce linearmente com o tamanho do conjunto de treino. Por fim, a corretude do algoritmo é garantida pelo Teorema de Osuna [5].

Sobre a generalização das SVMs

A noção da margem $\boldsymbol{\rho}$ viabiliza cotas para $\boldsymbol{R}(\boldsymbol{h})$ que independem da dimensão do espaço de entrada $\boldsymbol{\mathcal{X}}$, sem as quais o Truque do Kernel não seria possível.

Definição 1. *Fixe $\boldsymbol{\rho} > 0$, definimos a função de perda $\boldsymbol{L}_{\boldsymbol{\rho}}$ como sendo $\boldsymbol{L}_{\boldsymbol{\rho}}(\boldsymbol{y}, \boldsymbol{y'}) = \boldsymbol{\Phi}_{\boldsymbol{\rho}}(\boldsymbol{y}\boldsymbol{y'})$, em que*

$$\boldsymbol{\Phi}_{\boldsymbol{\rho}}(\boldsymbol{x}) = \begin{cases} 1, & \text{se } \boldsymbol{x} \leq 0 \\ 1 - \frac{\boldsymbol{x}}{\boldsymbol{\rho}}, & \text{se } 0 \leq \boldsymbol{x} \leq \boldsymbol{\rho} \\ 0, & \text{se } \boldsymbol{\rho} \leq \boldsymbol{x}, \end{cases}$$

e denotamos por $\hat{\boldsymbol{R}}_{\boldsymbol{S},\boldsymbol{\rho}}(\boldsymbol{h})$ o erro empírico de $\boldsymbol{h}$ sobre uma amostra $\boldsymbol{S}$ com respeito à $\boldsymbol{L}_{\boldsymbol{\rho}}$.

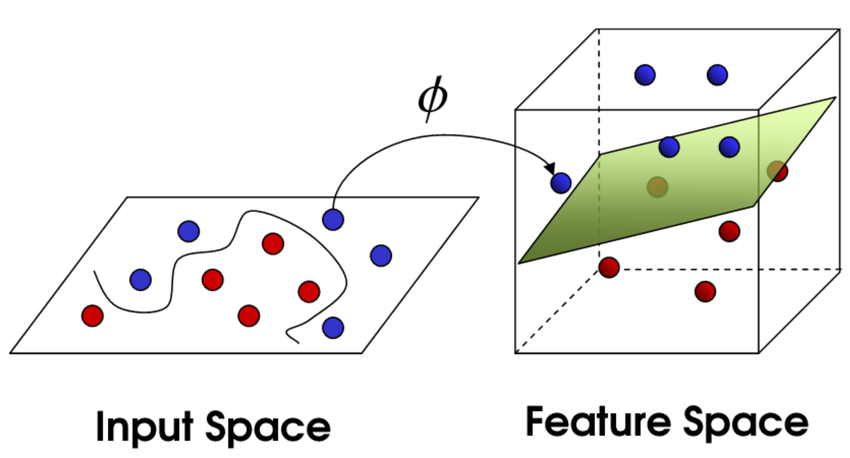
Utilizando cotas de generalização baseadas em Complexidade de Rademacher e uma estimativa para a complexidade da classe $\boldsymbol{\mathcal{H}} = \{\mathrm{x} \mapsto \mathrm{w}^T\mathrm{x} : ||\mathrm{w}|| \leq \boldsymbol{\Lambda}\}$, conseguimos estabelecer o seguinte resultado:

Teorema 1. *Seja $\boldsymbol{\mathcal{H}} = \{\mathrm{x} \mapsto \mathrm{w}^T\mathrm{x} : ||\mathrm{w}|| \leq \boldsymbol{\Lambda}\}$ e assuma que $\boldsymbol{\mathcal{X}} \subset \{\mathrm{x} : ||\mathrm{x}|| \leq \boldsymbol{R}\}$. Fixe $\boldsymbol{\rho} > 0$, então, para $\boldsymbol{\delta} > 0$, com probabilidade ao menos $1 - \boldsymbol{\delta}$, temos que o seguinte vale para toda $\boldsymbol{h} \in \boldsymbol{\mathcal{H}}$:*

$$\boldsymbol{R}(\boldsymbol{h}) \leq \hat{\boldsymbol{R}}_{\boldsymbol{S},\boldsymbol{\rho}}(\boldsymbol{h}) + 2\sqrt{\frac{\boldsymbol{\Lambda}^2\boldsymbol{R}^2}{\boldsymbol{m}\boldsymbol{\rho}^2}} + \sqrt{\frac{\log\frac{1}{\boldsymbol{\delta}}}{2\boldsymbol{m}}}.$$

O Truque do Kernel

Embora o SVM clássico seja um modelo linear, é possível adaptá-lo para contextos não lineares através da introdução de núcleos positivo-definidos (kernels). Essa conexão se dá através do Truque do Kernel, que corresponde a implicitamente mapear o conjunto de treino a um espaço de Hilbert de dimensão maior (potencialmente infinita, no caso do kernel RBF) e então aplicar o algoritmo linear sob os dados representados nesse outro espaço.



A principal motivação para o truque vem do fato que é mais fácil separar pontos em $\mathbb{R}^N$ do que em $\mathbb{R}^n$ para $\boldsymbol{N} >> \boldsymbol{n}$ [3]. O seguinte teorema dá uma caracterização simples para os kernels:

Teorema 2. *Seja $\boldsymbol{K} : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}} \to \mathbb{R}$ uma função simétrica e positiva definida, então existe um espaço de Hilbert $\boldsymbol{\mathcal{H}}$ e uma função $\boldsymbol{\phi} : \boldsymbol{\mathcal{X}} \to \boldsymbol{\mathcal{H}}$ tal que $\boldsymbol{K}(\mathrm{x}, \mathrm{y}) = \langle\boldsymbol{\phi}(\mathrm{x}), \boldsymbol{\phi}(\mathrm{y})\rangle_{\boldsymbol{\mathcal{H}}}$.*

Referências

[1] C. Burges and D. Crisp. Uniqueness of the svm solution. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

[2] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[3] T. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, June 1965.

[4] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2nd edition, 2018.

[5] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. *Neural Networks for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop*, pages 276–285, 1997.

[6] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft, April 1998.

[7] R. Rifkin, M. Pontil, and A. Verri. A note on support vector machine degeneracy. Technical report, MIT Artificial Intelligence Laboratory, 1999.