Máquinas de Vetores-Suporte (SVMs)
Parte III

1. Classificação Não-Linear

- Partimos de um problema linear, mas o que foi deduzido pode ser estendido de maneira significativa graças a alguns resultados assaz belos. De fato, é possível realizar classificação não-linear sob a égide dos conceitos expostos.
- Para que compreendamos de que forma isso se dá, consideremos que se disponha de um conjunto de dados $\{\mathbf{x}_i, d_i\}, i = 1, \ldots, N$. Consideremos ainda que cada padrão de entrada $\mathbf{x}_i$ tenha n atributos e que haja duas classes ($d_i = -1$ ou

$d_i = +1$). Basicamente, temos de achar uma superfície que separe os padrões de diferentes classes, e essa superfície pode ter uma forma qualquer.

1.1. Mapeamento e Espaço de Características

- Imaginemos agora que seja aplicado um mapeamento $\boldsymbol{\phi}: \mathbb{R}^n \rightarrow \mathbb{R}^m$ aos padrões $\mathbf{x}_i$, com $m > n$. Esse mapeamento *pode ser não-linear, e mapeia os dados num espaço de maior dimensão*. Essas duas características são interessantes: 1) a não-linearidade permite distorcer a posição dos dados, e 2) num espaço de maior dimensão, tende a ser mais simples separá-los.
- Como temos toda uma formulação adequada ao caso linear (partes I e II do material), surge uma ideia: e se usarmos um mapeamento $\boldsymbol{\phi}$ para "transformar o problema não-linear em um problema linear"? Afinal, esse mapeamento pode retorcer os dados, e isso num espaço de maior dimensão.

- Se o mapeamento tornar o problema linear, basta aplicar o que já estudamos e resolver o problema para os dados "modificados" $\{\boldsymbol{\phi}(\mathbf{x}_i), d_i\}, i = 1, \ldots, N$. De maneira mais rigorosa, diz-se que o problema será tratado como um problema linear no **espaço de características** (FS, ou, em inglês, *feature space*) gerado pelo mapeamento $\boldsymbol{\phi}(\cdot)$.
- A Figura 1 ilustra essa ideia: perceba como um problema não-linear em duas dimensões se torna linearmente separável em três dimensões graças ao mapeamento realizado.

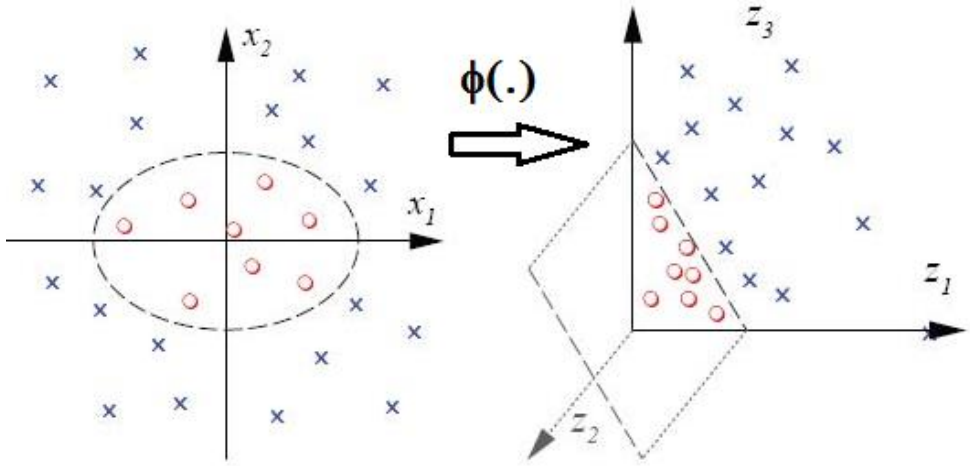

Figura 1 – Mapeamento e Solução Linear.

- Tendo isso em vista, a equação de um classificador linear no espaço de características seria a seguinte:

$$y = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) + b$$

- A equação que obtivemos para os parâmetros do classificador linear de máxima margem seria, então:

$$\mathbf{w}_o = \sum_{i=1}^{N} \lambda_i d_i \boldsymbol{\phi}(\mathbf{x}_i)$$

- Se usarmos esse vetor de pesos na equação da saída do classificador, teremos:

$$y = \sum_{i=1}^{N} \lambda_i d_i \boldsymbol{\phi}^T(\mathbf{x}_i)\, \boldsymbol{\phi}(\mathbf{x}) + b$$

- A equação mostra que não há mudanças essenciais: a saída do classificador ainda é calculada tendo por base produtos escalares, mas são produtos no espaço de características, ou seja, produtos $\boldsymbol{\phi}^T(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x})$.

- Tudo isso é maravilhoso, e resolvemos, sem dúvida, inúmeros problemas *in abstracto*. Há, não obstante, uma pergunta que se deve responder para resolver problemas mais concretos: qual deve ser o mapeamento $\boldsymbol{\phi}(\cdot)$?

1.2. Função de *Kernel* e Truque do *Kernel* (*Kernel Trick*)

- Aqui surge algo equivalente a um *deus ex machina*, com a vantagem de que se tem uma impressão de harmonia, e não de uma trama artificialmente resolvida. Perceba, inicialmente, que **não precisamos diretamente do mapeamento $\boldsymbol{\phi}(\cdot)$**; na verdade, precisamos apenas calcular produtos escalares no espaço que ele gera.
- De maneira bastante direta, a reviravolta no enredo é a seguinte: *produtos escalares no espaço de características podem ser calculados sem que se determine explicitamente o mapeamento* $\boldsymbol{\phi}(\cdot)$. É possível definir *funções de kernel* que generalizam a noção de produto escalar e permitem que efetuemos o cálculo de

produtos escalares no espaço de características *sem o uso de* $\boldsymbol{\phi}(\cdot)$*, ou seja, a partir do espaço original em que jazem os dados*. Isso é o que se chama **truque do kernel** *(kernel trick)*.

- Em outras palavras, lançaremos mão de funções de *kernel* $K(\mathbf{u}, \mathbf{v})$ tais que (BISHOP, 2006):

$$K(\mathbf{u}, \mathbf{v}) = \boldsymbol{\phi}^T(\mathbf{u})\boldsymbol{\phi}(\mathbf{v})$$

- Essas funções, para que possam atuar efetivamente como operadores de produto escalar, devem respeitar a condição de Mercer (CORTES E VAPNIK, 1995):

$$\int \int K(\mathbf{u}, \mathbf{v}) g(\mathbf{u}) g(\mathbf{v}) d\mathbf{u} d\mathbf{v} > 0$$

para toda função $g(\cdot)$ tal que:

$$\int g^2(\mathbf{u}) d\mathbf{u} < \infty$$

- O *kernel* mais utilizado é o *gaussiano*:

$$k(\mathbf{u}, \mathbf{v}) = exp(-\gamma \|\mathbf{u} - \mathbf{v}\|^2)$$

o qual possui um importante hiperparâmetro (γ). Também mencionaremos um exemplo de *kernel* polinomial:

$$k(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^T \mathbf{v} + 1)^d$$

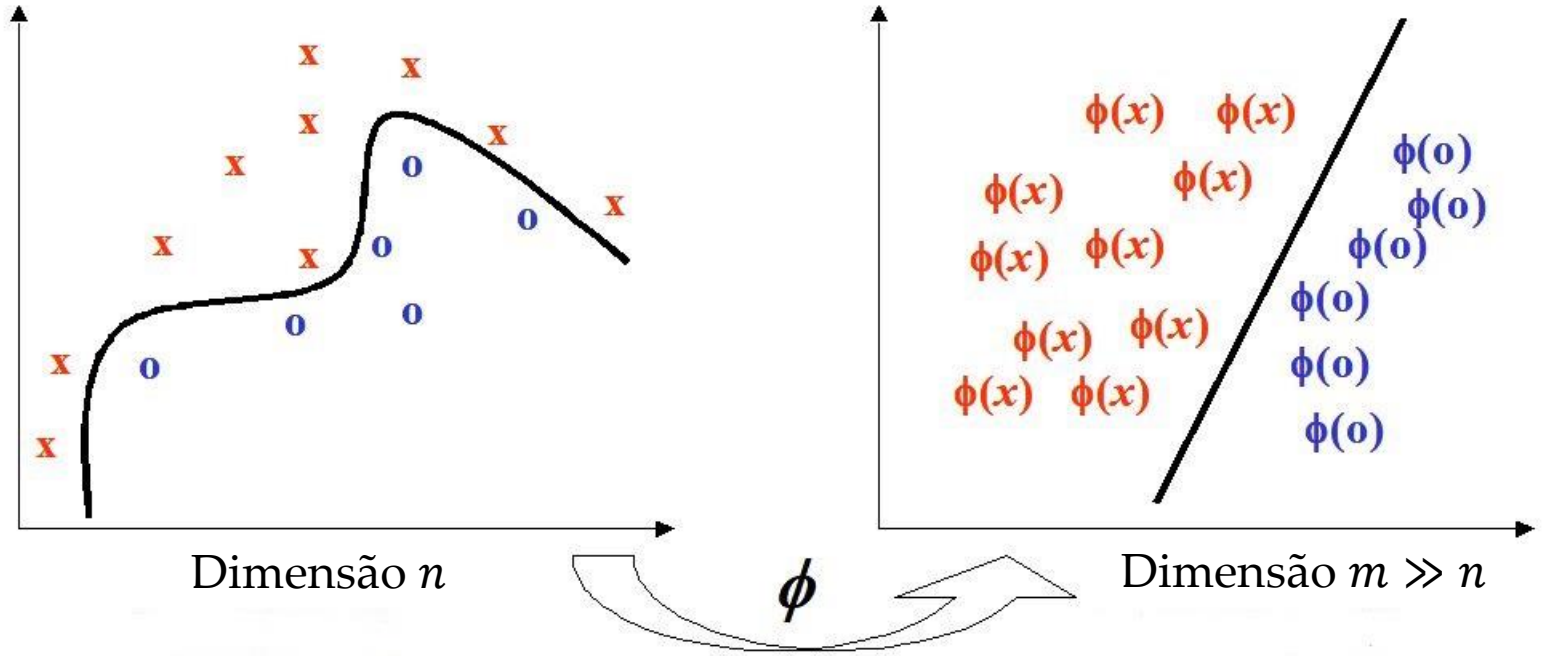

Figura 2 – Por meio do truque do *kernel*, o mapeamento dos dados para o espaço de características é implicitamente realizado.

1.3. Epílogo: Receituário

- Assim como ocorre, em música, na forma sonata, após o desenvolvimento vem a recapitulação. Como a função de *kernel* faz o papel de produto escalar, temos que a saída do classificador será:

$$y = \sum_{i=1}^{N} \lambda_i d_i \boldsymbol{\phi}^T(\mathbf{x}_i)\, \boldsymbol{\phi}(\mathbf{x}) + b = \sum_{i=1}^{N} \lambda_i d_i K(\mathbf{x}_i, \mathbf{x}) + b$$

- Conforme vimos na parte I, o problema dual que se resolve para obter os parâmetros do classificador tem por base a otimização com restrições da seguinte função:

$$L(\boldsymbol{\lambda}) = \boldsymbol{\lambda}^T \mathbf{1}_N - \frac{1}{2}\boldsymbol{\lambda}^T \mathbf{D} \boldsymbol{\lambda}$$

- A informação sobre o conjunto de dados está contida na matriz **D**. Em sua forma original (caso linear), a matriz é composta de elementos do tipo:

$$D_{ij} = d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

No caso não-linear, basta trocar o produto escalar clássico pelo produto gerado a partir da função de *kernel* escolhida:

$$D_{ij} = d_i d_j K(\mathbf{x}_i, \mathbf{x}_j)$$

- Daí em diante, o problema é resolvido virtualmente da mesma forma, bastando, quando necessário, utilizar a nova forma do produto escalar. Cabe frisar que a formulação do problema de margem suave (*soft margin*) também é feita de maneira direta com o uso de funções de *kernel*.
- Os hiperparâmetros associados às funções de *kernel* (e.g. o valor de γ no caso gaussiano) devem ser escolhidos com cuidado. Tipicamente, realiza-se uma busca sobre uma lista de valores tendo por critério norteador o desempenho junto a um conjunto de validação (e.g. num esquema de validação cruzada).

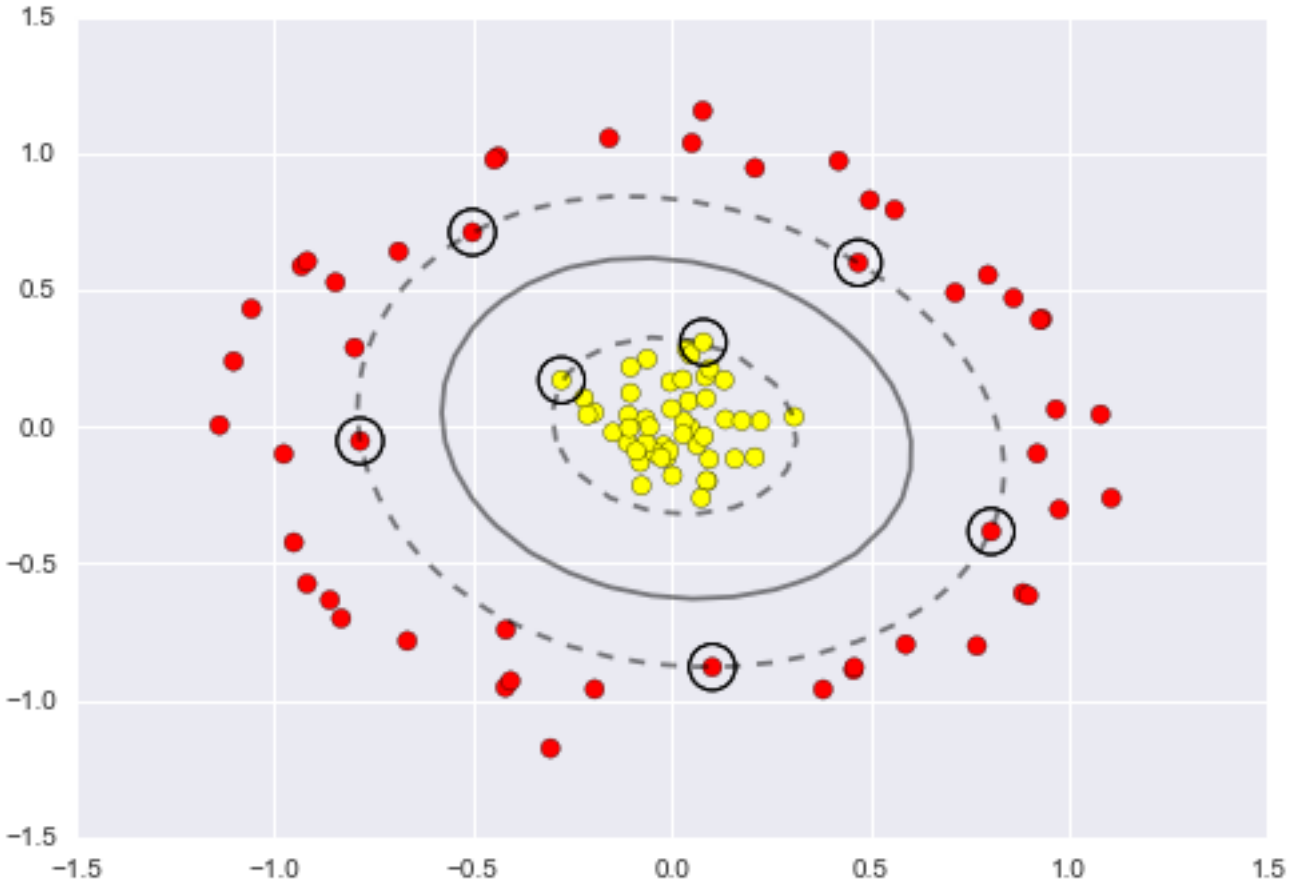

Figura 3 – Visualização da fronteira de decisão (curva sólida em preto) e da margem (curvas em pontilhado) no espaço original dos dados para uma SVM não-linear, isto é, que utiliza uma função de *kernel*. As amostras destacadas de cada classe (pontos circulados) representam os vetores-suporte obtidos no processo de otimização.

2. Referências bibliográficas

BISHOP, C., *Pattern Recognition and Machine Learning*, Springer, 2006.

CORTES, C., VAPNIK, V., "Support Vector Networks", *Machine Learning*, vol. 20, pp. 273 – 297, 1995.