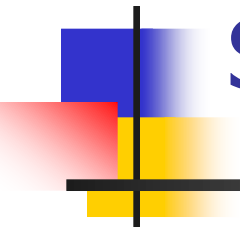


Uma Introdução a SVM
Support Vector Machines

Obs: Baseada nos slides de Martin Law

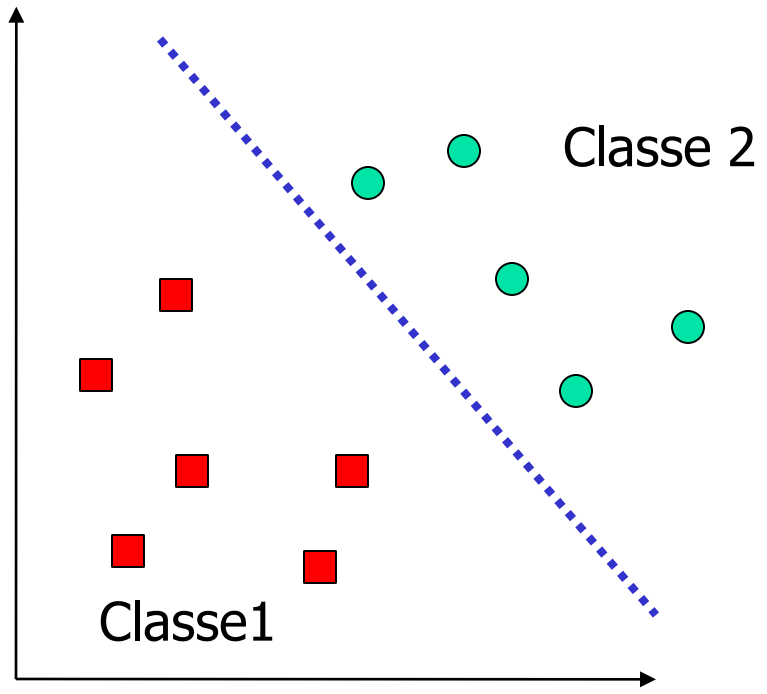
Sumário

- Historia das SVMs
- Duas classes, linearmente separáveis
  - O que é um bom limite para a decisão?
- Duas classes, não linearmente separáveis
- Como lidar com não linearidades: kernel
- Exemplo de SVM
- Conclusão

Historia do SVM

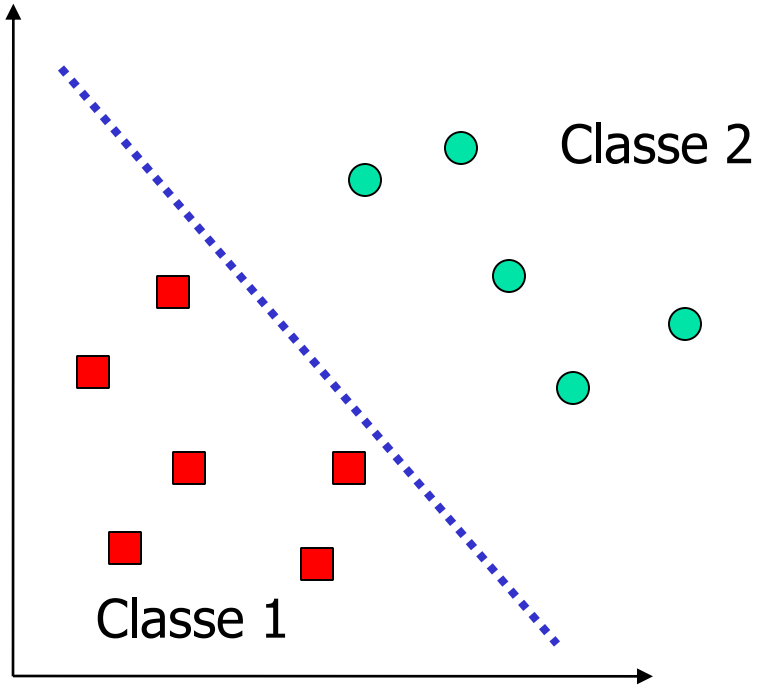
- SVM é um classificador derivado da teoria de aprendizado estatístico por Vapnik e Chervonenkis
- SVM foi introduzida em COLT-92
- SVM se tornou famosa quando , usando mapas de pixels como entrada, ela tive acuracia comparável a uma sofisticada rede neural com características elaboradas na tarefa de reconhecimento de escrita
- Atualmente, as pesquisas em SVM esta relacionada a:
  - Métodos de Kernel, classificadores de máxima margens (large margin classifiers), reprodução de espaços de kernel Hilbert, processos Gaussianos

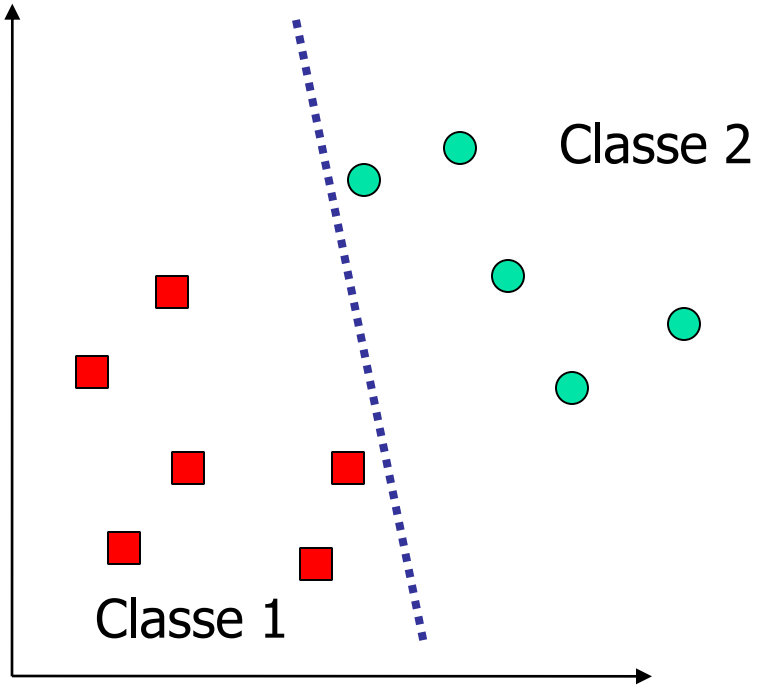
Problemas de dois classes: Linearmente Separáveis



- Diferentes limites de decisão podem separar estas duas classes
- Qual devemos escolher?

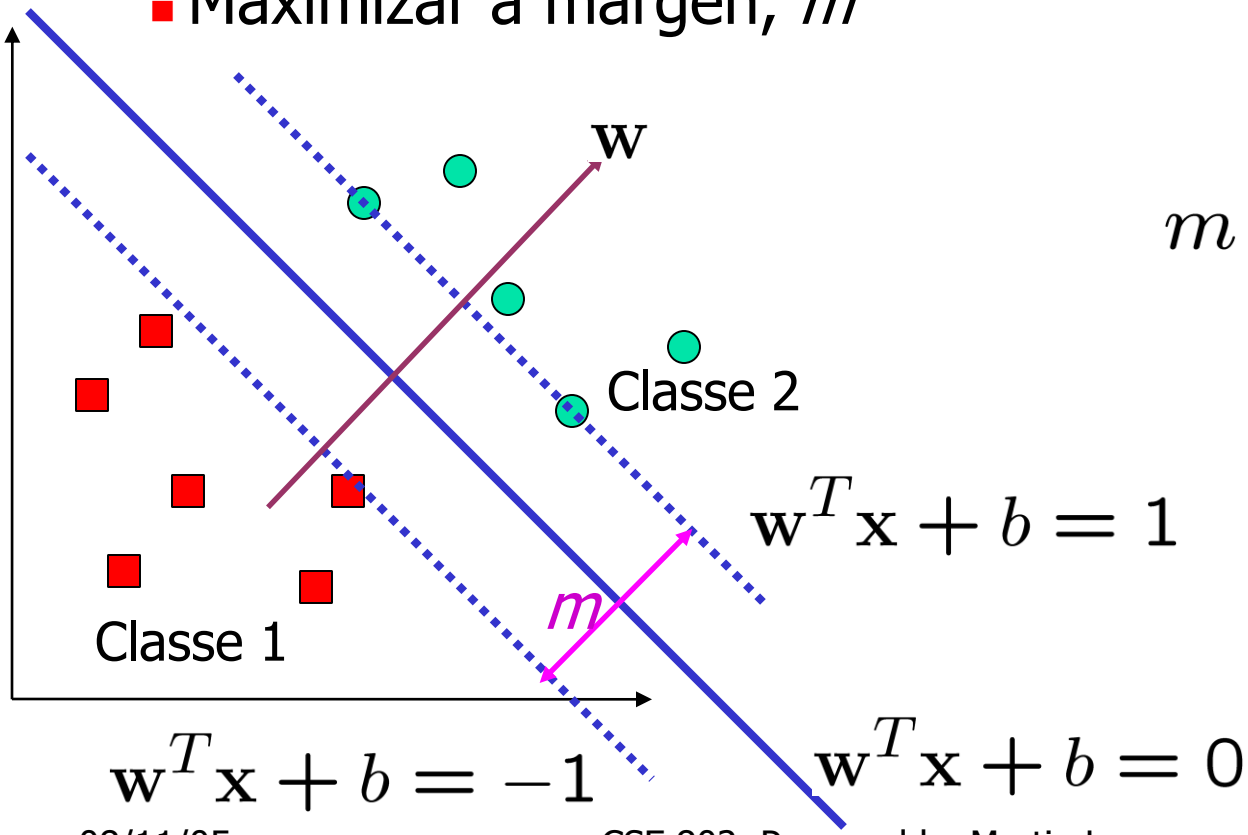
Exemplos





Bons limites de decisão: As Margens devem ser máximizadas

- O limite de decisão deve estar o mais afastado dos dados de ambas classes
  - Máximizar a margen, *m*



$$m = \frac{2}{||\mathbf{w}||}$$

O problema de otimização

- Seja $\{x_1, ..., x_n\}$ um conjunto de dados e seja $y_i \in \{1,-1\}$ a classe da instância x_i
- O limite da decisão deve classificar todos os pontos corretamente $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad \forall i$
- O problema de otimização é

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i$$

Transformação no problema dual

- O problema pode ser transformado no seu dual

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

- Isto é um problema quadratico (QP)
  - Os máximos globais de α_i podem ser sempre encontrados
- **w** podem ser calculados $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$

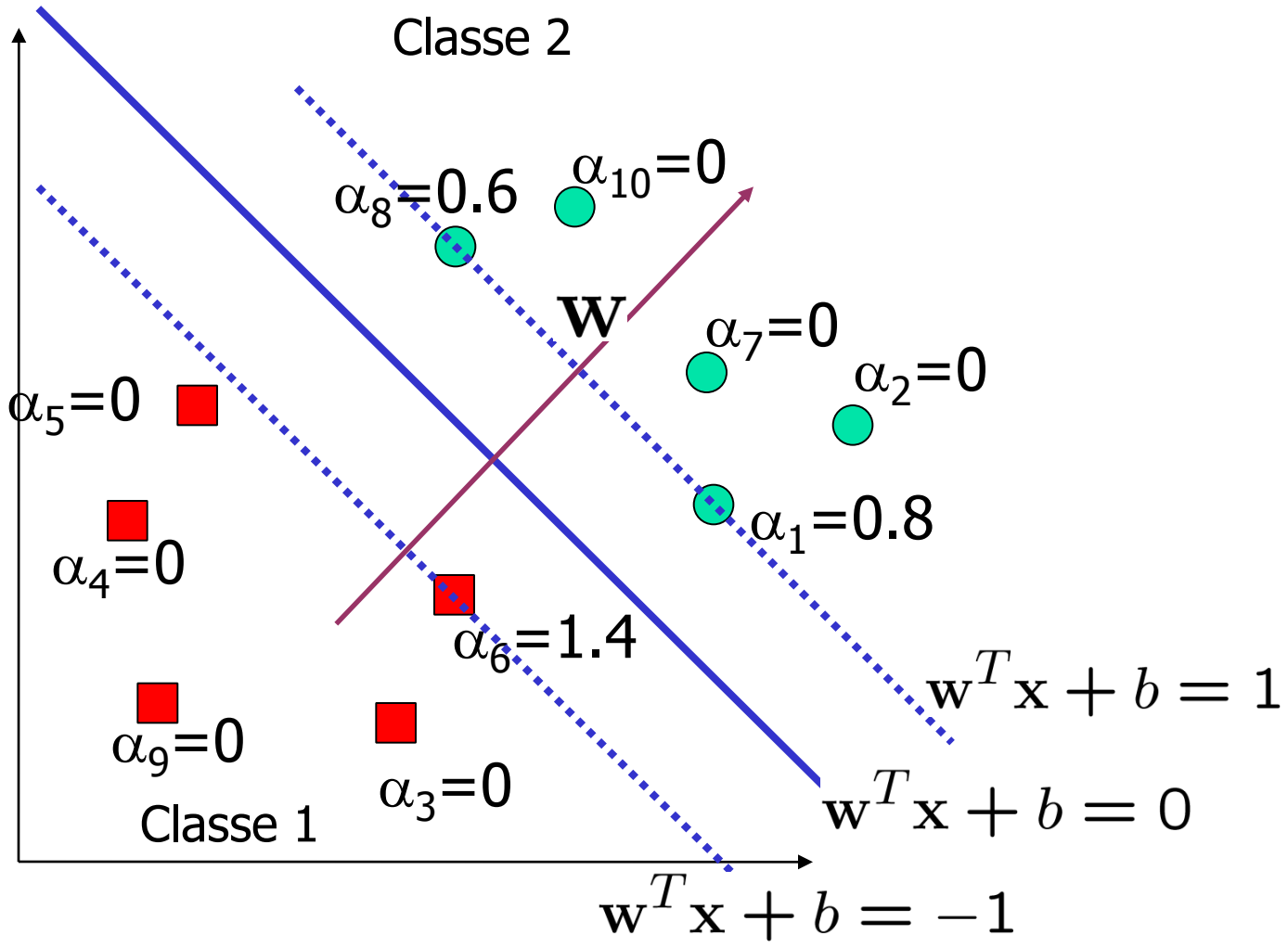
Características das Soluções

- Muitos α_i são zeros
  - **w** é uma combinação linear de um pequeno numero de dados
  - Representação esparsa
- $\mathbf{x}_i$ com não-zero α_i são chamados vetores de suporte (SV).
  - Seja t_j (j=1, ..., s) os índices dos s vetores de suporte

$$\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$$

- Para um novo dato **z**
  - Calcule $\mathbf{w}^T\mathbf{z} + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} (\mathbf{x}_{t_j}^T \mathbf{z}) + b$ e classifique **z** como classe 1 se a soma é positiva

Uma interpretação Geométrica

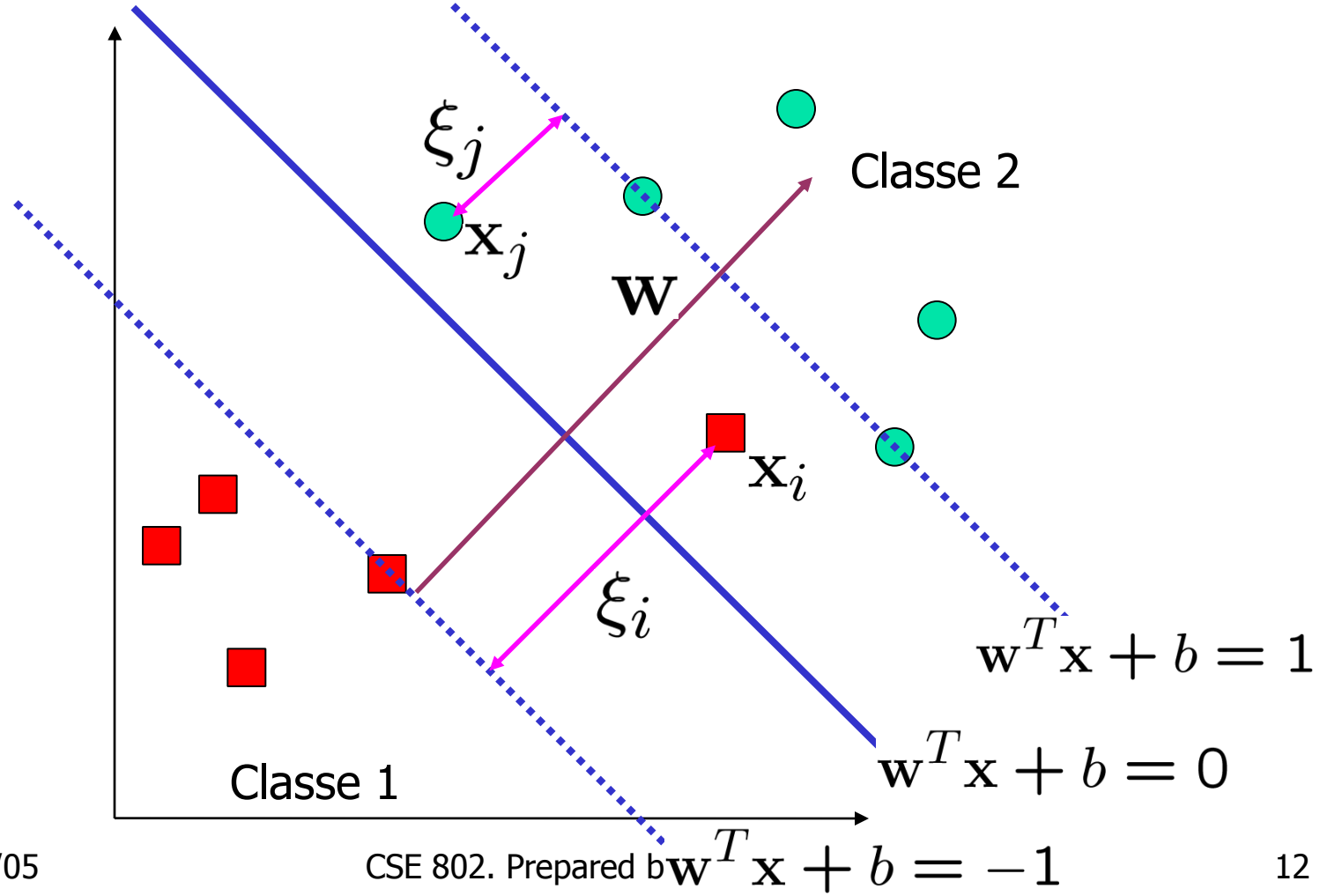


Algumas Observações

- Existem limites teóricos sobre o erro em dados não vistos pela SVM
  - Maior margem, menor o limite
  - Menor número de SV, menor o limite
- Note que ambos treinamento e teste, os dados são somente referenciados por seu produto interno, $\mathbf{x}^T\mathbf{y}$
  - Isto é importante para generalizar o caso não linear

E quando não existe separação linear

- Se permite um erro de classificação ξ_i



Hyperplano de Margem Suave

- Defina ξ_i=0 se não existe erro para x_i
  - ξ_i são somente "variaveis slack variables" na otimização

$$\begin{cases} \mathbf{w}^T\mathbf{x}_i + b \geq 1 - \xi_i & y_i = 1 \\ \mathbf{w}^T\mathbf{x}_i + b \leq -1 + \xi_i & y_i = -1 \\ \xi_i \geq 0 & \forall i \end{cases}$$

  - minimizar $\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$
  - *C* :parametro de balance entre o erro e a margem
- Assim o problema de otimização é

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$
$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

O problema de otimização

- O problema dual é

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

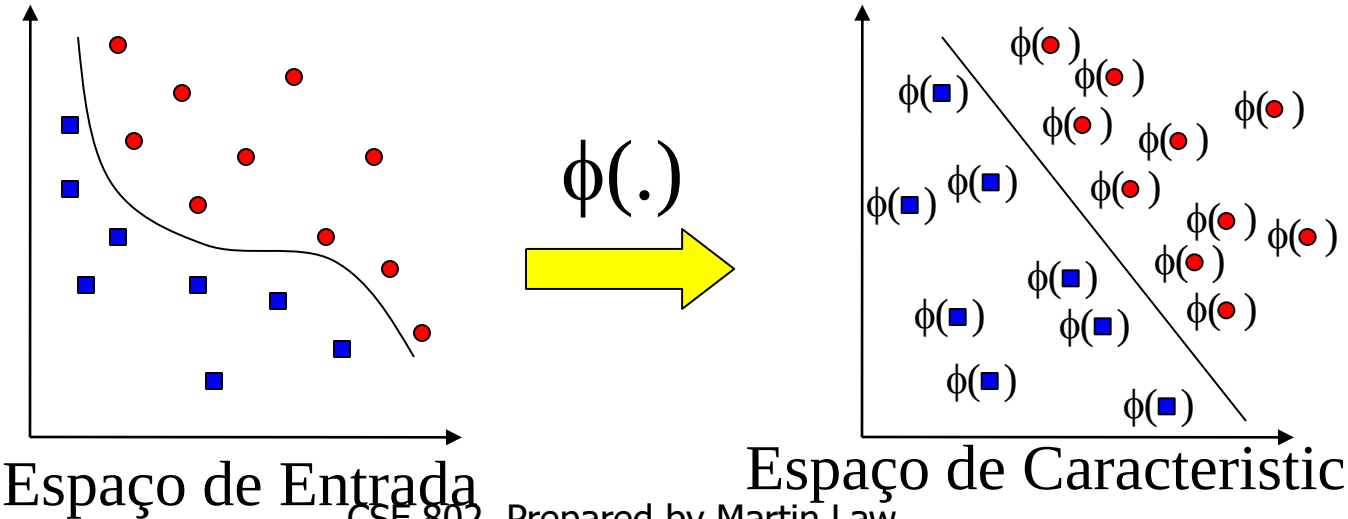
- **w** é calculado com $\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$
- A única diferença é que existe um limite superior C em α_i
- De novo, um QP pode ser usado pra encontrar α_i

Extensão a Limites de Decisão não lineares

- Ideia Chave: transformar $\mathbf{x}_i$ a um espaço de maior dimensão
  - Espaço de Entrada: o espaço $\mathbf{x}_i$
  - Espaço de características: o espaço de $\phi(\mathbf{x}_i)$ após a transformação
- Porque transformar?
  - Operações lineares no espaço de características é equivalente a operações não lineares no espaço de entrada
  - A tarefa de classificação pode ser mais fácil com a transformação apropriada. Exemplo: XOR

Extensão a Limites de Decisão Não Lineares

- Problemas possíveis da transformação
  - Difíceis de calcular e de obter uma boa estimação
- SVM resolve estes dois problemas simultaneamente
- As funções Kernel não precisam ser calculados



Exemplo de Transformação

- Defina a função de kernel $K(\mathbf{x},\mathbf{y})$ como

$$K(\mathbf{x},\mathbf{y}) = (1 + x_1y_1 + x_2y_2)^2$$

- Considere a seguinte transformação

$$\phi(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

$$\phi(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}) = (1, \sqrt{2}y_1, \sqrt{2}y_2, y_1^2, y_2^2, \sqrt{2}y_1y_2)$$

$$\begin{aligned} \langle \phi(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}), \phi(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}) \rangle &= (1 + x_1y_1 + x_2y_2)^2 \\ &= K(\mathbf{x},\mathbf{y}) \end{aligned}$$

- O produto interno pode ser calculado por K sem necessidade de realizar o mapeamento $\phi(.)$

Truque Kernel

- As relações entre a função de Kernel *K* e o mapeamento $\phi(.)$ é

$$K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$$

  - Isto é conhecido pelo nome de truque de Kernel
- Na pratica, pode-se especificar, especificando $\phi(.)$ indiretamente, ao invés de escolher $\phi(.)$
- Intuitivamente, *K* (**x**,**y**) representa nosso desejo de noção de similaridade entre **x** e **y** e isto forma nosso conhecimento a priori
- *K* (**x**,**y**) precisa satisfazer a condição Mercer para que $\phi(.)$ exista

Exemplos de Funções de Kernel

- Kernel polinomial com grau *d*

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + 1)^d$$

- Funções com base Radial σ

$$K(\mathbf{x}, \mathbf{y}) = \exp(-||\mathbf{x} - \mathbf{y}||^2/(2\sigma^2))$$

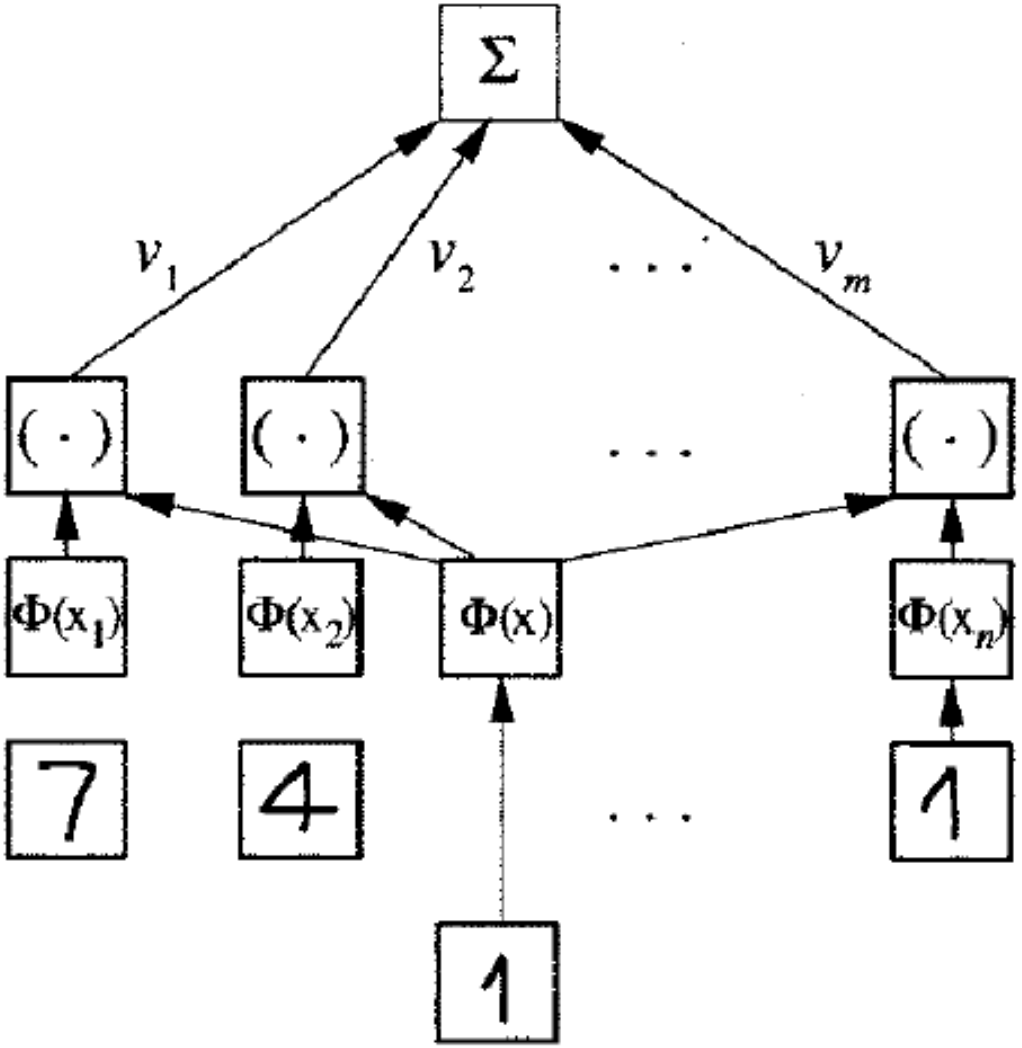
  - Fortemente relacionadas com redes neurais com base radial
  - Sigmoid com os parâmetros κ e θ

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa\mathbf{x}^T\mathbf{y} + \theta)$$

  - Não satisfaz a condição de Mercer para todo κ e θ

- Pesquisa em diferentes funções de kernel é muito ativa

Examplo de Aplicação de SVM : Reconhecimento de Escrita



output $\Sigma\, v_i\, k\,(x,x_i) + b$

weights

dot product $(\Phi(x)\cdot\Phi(x_i)) = k(x,x_i)$

mapped vectors $\Phi(x_i)$, $\Phi(x)$

support vectors $x_1 \ldots x_n$

test vector x

Modificações devido a Funções de Kernel

- Mude todos os produtos internos pela função de kernel
- Para treinamento

Original

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1.i=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

Com kernel

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

Modificações devido a F.K.

- Para testar, o novo data **z** é classificado como da classe 1 se $f \geq 0$, e classe 2 se $f < 0$

Original

$$\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$$

$$f = \mathbf{w}^T \mathbf{z} + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}^T \mathbf{z} + b$$

Com kernel

$$\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \phi(\mathbf{x}_{t_j})$$

$$f = \langle \mathbf{w}, \phi(\mathbf{z}) \rangle + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} K(\mathbf{x}_{t_j}, \mathbf{z}) + b$$

Exemplo

- Suponha que temos 5 1D dados
  - x_1=1, x_2=2, x_3=4, x_4=5, x_5=6, com 1, 2, 6 da classe 1 e 4, 5 da classe 2 $\Rightarrow$ y_1=1, y_2=1, y_3=-1, y_4=-1, y_5=1
- A função de kernel polinomial de grau 2
  - $K(x,y) = (xy+1)^2$
  - C = 100
- Primeiro encontra-se α_i (i=1, …, 5) como

$$\text{max.} \sum_{i=1}^{5} \alpha_i - \frac{1}{2} \sum_{i=1}^{5} \sum_{i=1}^{5} \alpha_i \alpha_j y_i y_j (x_i x_j + 1)^2$$

$$\text{subject to } 100 \geq \alpha_i \geq 0, \sum_{i=1}^{5} \alpha_i y_i = 0$$

Exemplo

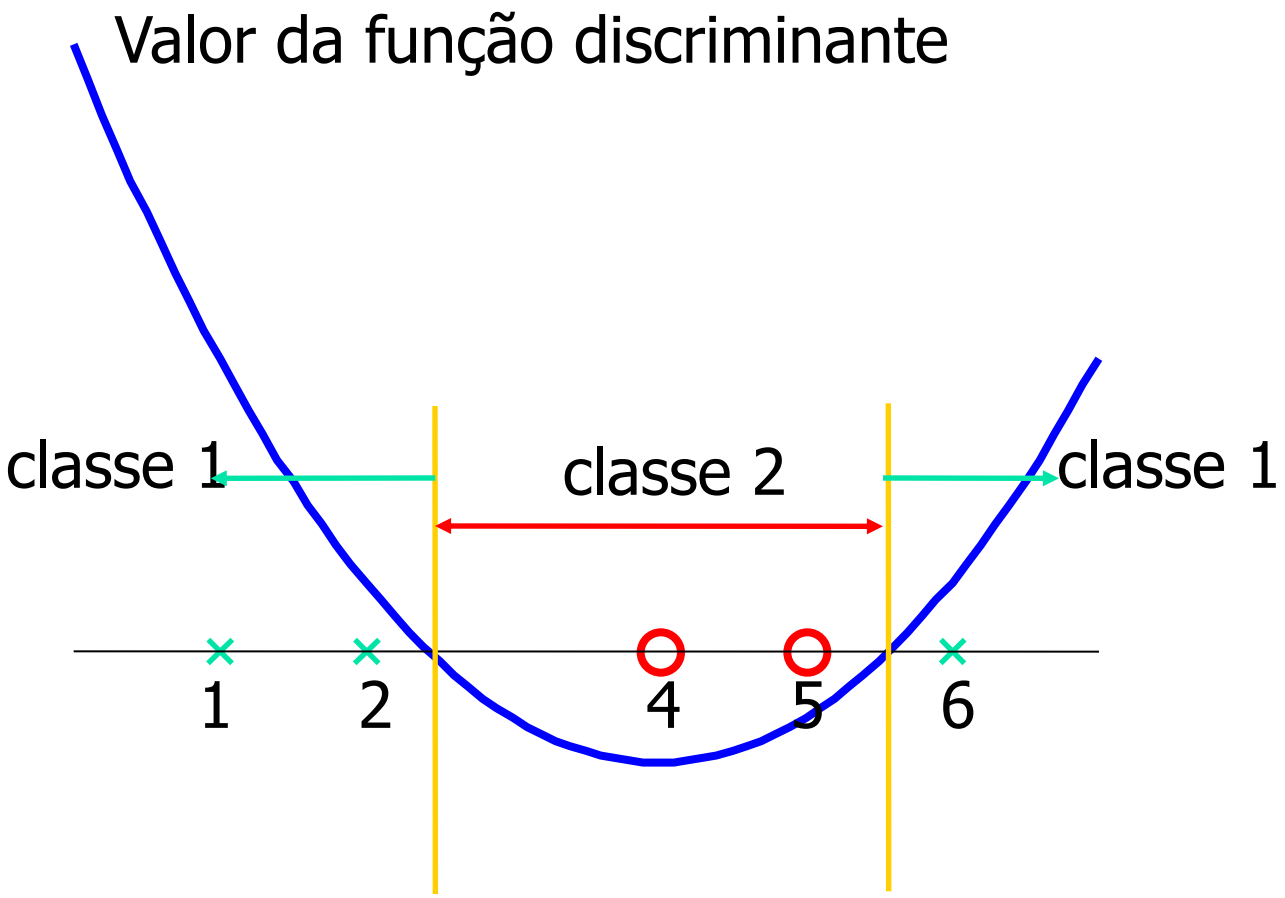
- Usando um QP solver, obtem se
  - α_1=0, α_2=2.5, α_3=0, α_4=7.333, α_5=4.833
  - Note que as restrições são satisfeitas
  - Os vetores de suporte são{x_2=2, x_4=5, x_5=6}
- A função descriminante é

$$f(y) = 2.5(1)(2y+1)^2 + 7.333(-1)(5y+1)^2 + 4.833(1)(6y+1)^2 + b$$
$$= 0.6667x^2 - 5.333x + b$$

- *b* é recuperado por f(2)=1 ou f(5)=-1 ou f(6)=1, como x_2, x_4, x_5 esta em $y_i(\mathbf{w}^T\phi(z) + b) = 1$ b=9

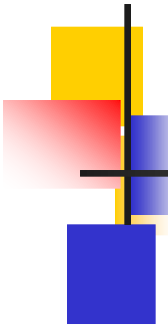
$$\rightarrow \quad f(y) = 0.6667x^2 - 5.333x + 9$$

Exemplo


Valor da função discriminante
classe 1
classe 2
classe 1
1
2
4
5
6

Classificação Multi-classe

- SVM é basicamente um classificador binário
- Pode mudar-se a formulação QP para permitir classificação multi-classe
- Comumente, os dados são divididos em dois partes em diferentes maneiras e são treinadas diferentes SVM uma para cada divisão
- Classificação Multi-classe é realizada combinando a saída de todas as SVM
  - Regra da maioria
  - Código de correção do erro
  - Gráfico acíclico direto

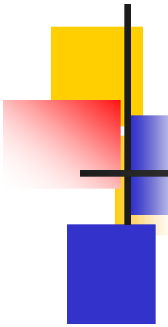


Software

- Uma lista de implementações de SVM
  http://www.kernel-machines.org/software.html
- Diferentes Matlab toolboxes para SVM

Resumo: Passos para a classificação

- Prepare o padrão de entrada
- Selecione a função de kernel
- Selecione o parâmetro da função de kernel e o valor de C
- Execute o algoritmo de treinamento e obtenha α_i
- Os novos dados podem ser classificados usando α_i e os vetores de suporte



Conclusão

- SVM é uma alternativa a redes neurais
- Dois conceitos da SVM: maximize a margem e o truque de kernel
- Diferentes áreas de pesquisa ativas
- Muitas implementações de SVM estão disponíveis

Resources

- http://www.kernel-machines.org/
- http://www.support-vector.net/
- http://www.support-vector.net/icml-tutorial.pdf
- http://www.kernel-machines.org/papers/tutorial-nips.ps.gz
- http://www.clopinet.com/isabelle/Projects/SVM/applist.html