中国科学院大学人工智能学院硕士课《模式识别》 2024.12.25, 教1-107, 怀柔, 北京

课程号： 180206081100M1001H-01

第15章(第2讲)

支持向量机与核方法

Support Vector Machine & Kernel Methods

向 世 明

smxiang@nlpr.ia.ac.cn

https://people.ucas.ac.cn/~xiangshiming

时空数据分析与学习课题组(STDAL)

中科院自动化研究所　多模态人工智能系统国家重点实验室

助教：　杨　奇　(yangqi2021@ia.ac.cn)

　　　　张　涛　(zhangtao2021@ia.ac.cn)

上次课核心知识点

- **VC维**
  - 假设空间H的VC维是能被H打散的最大示例集的大小。
  - VC(H)=d表明存在大小为d的示例集能被假设空间H打散。
  - 通常按如下方式来计算VC维：
    - 若存在大小为d的示例集能被H打散，但不存在任何大小为d+1示例集能被H打散，则H的VC维是d。

The Probabilistic Guarantee：

置信水平: $\log(1/\delta)$

$$E_{test} \le E_{train} + \left(\frac{h + h\log(2N/h) - \log(p/4)}{N} \right)^{\frac{1}{2}}$$

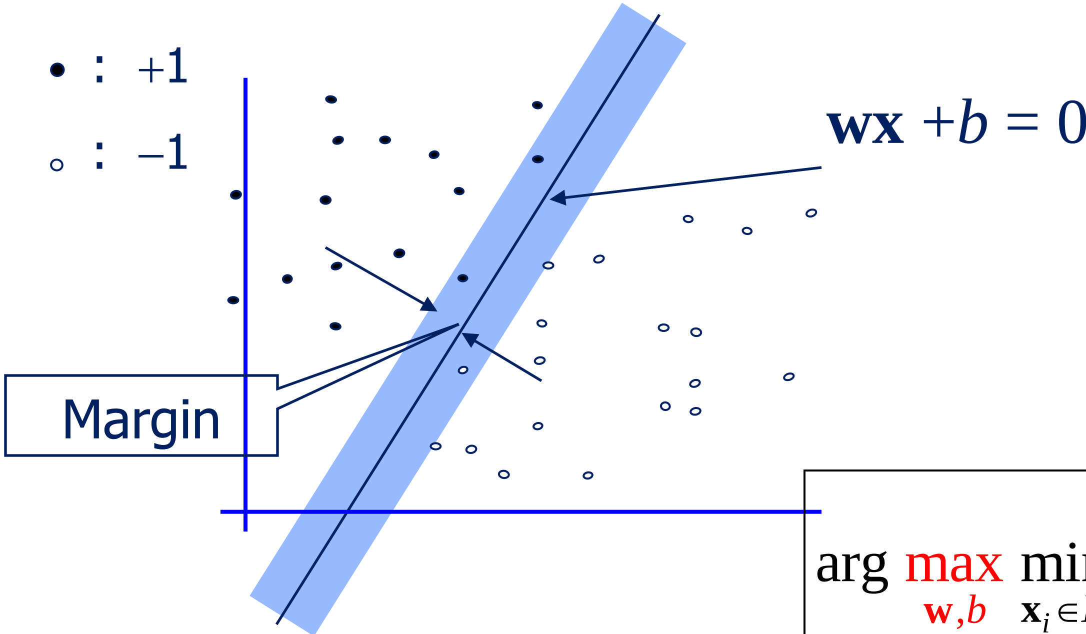
where N = size of training set

h = VC dimension of the model

p = upper bound on probability that this bound fails

此边界失败的概率上限

Estimate Margin (Method 1)



$$\mathbf{wx}_i + b \geq 0 \text{ iff } y_i = 1$$

$$\mathbf{wx}_i + b \leq 0 \text{ iff } y_i = -1$$

⇩

$$y_i(\mathbf{wx}_i + b) \geq 0$$

Strategy:

$$\forall \mathbf{x}_i \in D: \ |b + \mathbf{x}_i \cdot \mathbf{w}| \geq 1$$

$$\arg \max_{\mathbf{w},b} \min_{\mathbf{x}_i \in D} \frac{|b + \mathbf{x}_i \cdot \mathbf{w}|}{\sqrt{\sum_{i=1}^{d} w_i^2}}$$

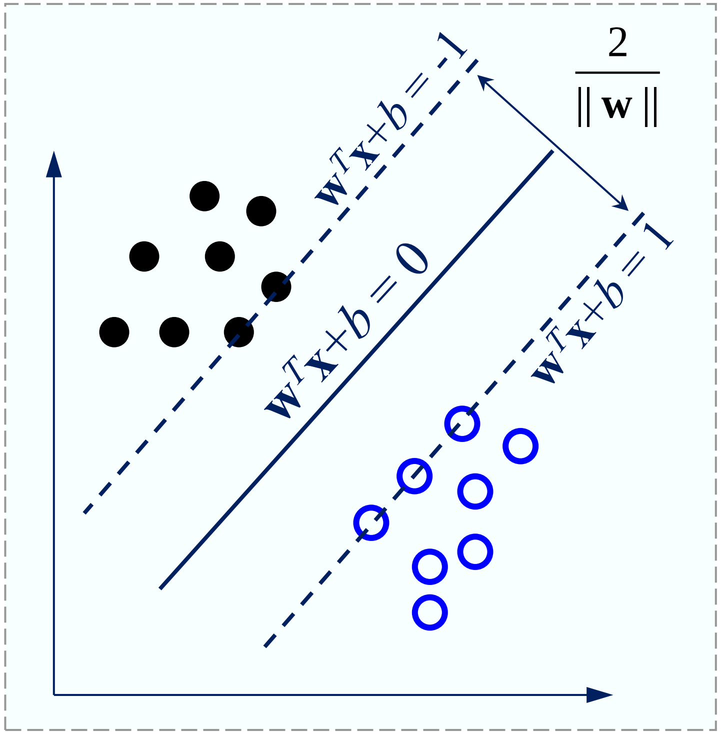
$$\text{subject to } \forall \mathbf{x}_i \in D: y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 0$$

⇩

$$\arg \min_{\mathbf{w},b} \sum_{i=1}^{d} w_i^2$$

$$\text{subject to } \forall \mathbf{x}_i \in D: y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1$$

Estimate Margin (Method 1)



总是可以做到的（其一：数值上）

Strategy:

$$\forall \mathbf{x}_i \in D:\ \left|b+\mathbf{x}_i\cdot\mathbf{w}\right|\geq 1$$

$$\arg\max_{\mathbf{w},b}\ \min_{\mathbf{x}_i\in D}\frac{\left|b+\mathbf{x}_i\cdot\mathbf{w}\right|}{\sqrt{\sum_{i=1}^{d}w_i^2}}$$

$$\text{subject to}\ \ \forall \mathbf{x}_i\in D: y_i\left(\mathbf{x}_i\cdot\mathbf{w}+b\right)\geq 0$$

⇩

$$\arg\min_{\mathbf{w},b}\sum_{i=1}^{d}w_i^2$$

$$\text{subject to}\ \ \forall \mathbf{x}_i\in D: y_i\left(\mathbf{x}_i\cdot\mathbf{w}+b\right)\geq 1$$

- ✓ 让负类样本点 $\mathbf{w}^T\mathbf{x}+\mathrm{b}\leq -1$
- ✓ 让正类样本点 $\mathbf{w}^T\mathbf{x}+\mathrm{b}\geq +1$
- ✓ 注意到负类标签 y_i=-1，正类标签 y_i=+1，综合起来，有：

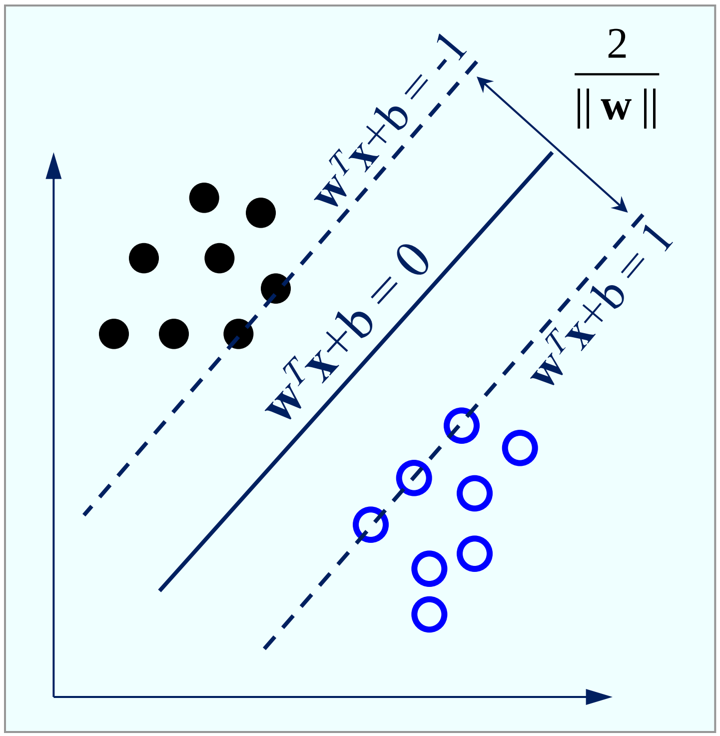
$$y_i\,(\mathbf{w}^T\mathbf{x}+\mathrm{b})\geq 1$$

（其二，考虑类别时的约束统一表示）

线性可分支持向量机

- **学习模型**

线性可分情形



给定训练集：

$T=\{(\mathbf{x}_1,y_1),\ \ldots,\ (\mathbf{x}_n,y_n)\},y_i\in\{+1,\text{-}1\}$

任务：估计最大间隔分类超平面

$$\min_{\mathbf{w},b}\ \frac{1}{2}\|\mathbf{w}\|^2$$

$$s.t.\quad y_i(\mathbf{w}^T\mathbf{x}_i+b)-1\geq 0,$$
$$i=1,2,\cdots,n$$

分类超平面： $\mathbf{w}^T\mathbf{x}+b=0$

分类决策函数： $f(\mathbf{x})=sign(\mathbf{w}^T\mathbf{x}+b)$

线性不可分-学习模型（支持向量机）

Describe the Theory　　Describe the Mistake

体现了表达能力　　体现了经验风险

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$s.t. \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i,$$

$$\xi_i \geq 0,$$

$$i = 1, 2, \cdots, n$$

C : tradeoff parameter between error and margin; chosen by the user; large C means a higher penalty to errors

目标函数第一项表示使margin尽量大，第二项表示使误差分类点的个数尽量小。

- **复习KKT条件**（数学知识点，不要求）：

原始问题

$$\min_{\mathbf{x}\in R^d} \quad f(\mathbf{x})$$

$$s.t. \quad c_i(\mathbf{x}) \le 0, \quad i=1,2,\ldots,k$$

$$h_j(\mathbf{x}) = 0, \quad j=1,2,\ldots,l$$

KKT条件

$$\nabla_{\mathbf{x}} L(\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\beta}) = 0$$

$$\nabla_{\boldsymbol{\alpha}} L(\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\beta}) = 0$$

$$\nabla_{\boldsymbol{\beta}} L(\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\beta}) = 0$$

$$\alpha_i c_i(\mathbf{x}) = 0, \quad i=1,2,..,k$$

$$c_i(\mathbf{x}) \le 0, \quad i=1,2,..,k$$

$$\alpha_i \ge 0, \quad i=1,2,..,k$$

$$h_j(\mathbf{x}) = 0, \quad j=1,2,..,l$$

$$L(\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\beta}) = f(\mathbf{x}) + \sum_{i=1}^{k} \alpha_i c_i(\mathbf{x}) + \sum_{j=1}^{l} \beta_j h_j(\mathbf{x})$$

广义拉格朗日函数

支持向量机(对偶)

- **对偶算法（线性可分情形）**
  - ✓ 在约束最优化问题中，常利用拉格朗日对偶性将原始问题转化为对偶问题进行求解
  - ✓ 对偶算法往往容易求解
  - ✓ 对偶算法可以推广到核学习

$$
\begin{aligned}
\min_{\mathbf{x}\in R^d} \quad & f(\mathbf{x}) \\
s.t. \quad & c_i(\mathbf{x}) \le 0, \quad i=1,2,\dots,k \\
& h_j(\mathbf{x}) = 0, \; j=1,2,\dots,l
\end{aligned}
$$

广义拉格朗日函数

$$L(\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\beta}) = f(\mathbf{x}) + \sum_{i=1}^{k}\alpha_i c_i(\mathbf{x}) + \sum_{j=1}^{l}\beta_j h_j(\mathbf{x})$$

数学知识点

（拉格朗日对偶性）

$$
\begin{aligned}
\min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\
s.t. \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \ge 0, \\
& i=1,2,\cdots,n
\end{aligned}
$$

原始问题

$$
\begin{aligned}
\max_{\boldsymbol{\alpha}} \min_{\mathbf{w},b} \quad & L(\mathbf{w},b,\boldsymbol{\alpha}) \\
s.t. \quad & \alpha_i \ge 0, \quad i=1,2,\dots,n
\end{aligned}
$$

$$L(\mathbf{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n}\alpha_i y_i(\mathbf{w}^T\mathbf{x}_i + b) + \sum_{i=1}^{n}\alpha_i$$

对偶问题

证明见：李航：统计学习方法，清华大学出版社, 2012 (第7章) （本课程不要求）

支持向量机(对偶)

- **对偶问题求解（线性可分）**
  - (1) 求 $\min_{\mathbf{w},b} L(\mathbf{w},b,\boldsymbol{\alpha})$

$$\nabla_{\mathbf{w}} L(\mathbf{w},b,\boldsymbol{\alpha}) = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = \mathbf{0} \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$\nabla_{b} L(\mathbf{w},b,\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\Downarrow$$

$$\min_{\mathbf{w},b} L(\mathbf{w},b,\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \underline{(\mathbf{x}_i \cdot \mathbf{x}_j)} + \sum_{i=1}^{n} \alpha_i$$

支持向量机(对偶)

– (2) 求对偶问题，即求 $\min_{\mathbf{w},b} L(\mathbf{w},b,\boldsymbol{\alpha})$ 对 $\boldsymbol{\alpha}$ 的极大

$$\max_{\boldsymbol{\alpha}} \quad -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)+\sum_{i=1}^{n}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\alpha_i \geq 0, \quad i=1,2,\ldots,n$$

⇓

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)-\sum_{i=1}^{n}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\alpha_i \geq 0, \quad i=1,2,\ldots,n$$

- ✓ This is a convex quadratic programming (QP) problem
- ✓ Global maximum of α_i can always be found
  - ➢ well established tools for solving this optimization problem

软间隔最大化（线性不可分）：

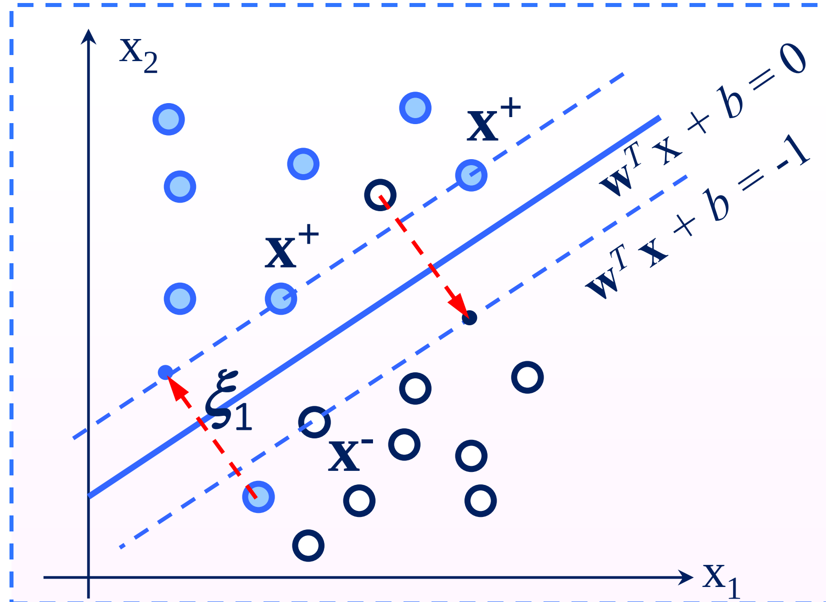
$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$s.t. \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \ge 1 - \xi_i,$$

$$\xi_i \ge 0,$$

$$i = 1, 2, \cdots, n$$

原始问题



$\Downarrow$（广义拉格朗日函数）$\Downarrow$

$$L(\mathbf{w}, b, \xi, \boldsymbol{\alpha}, \mu) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i y_i(\mathbf{w}^T\mathbf{x}_i + b + \xi_i) + \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\mu_i\xi_i$$

$\Downarrow$

拉格朗日对偶

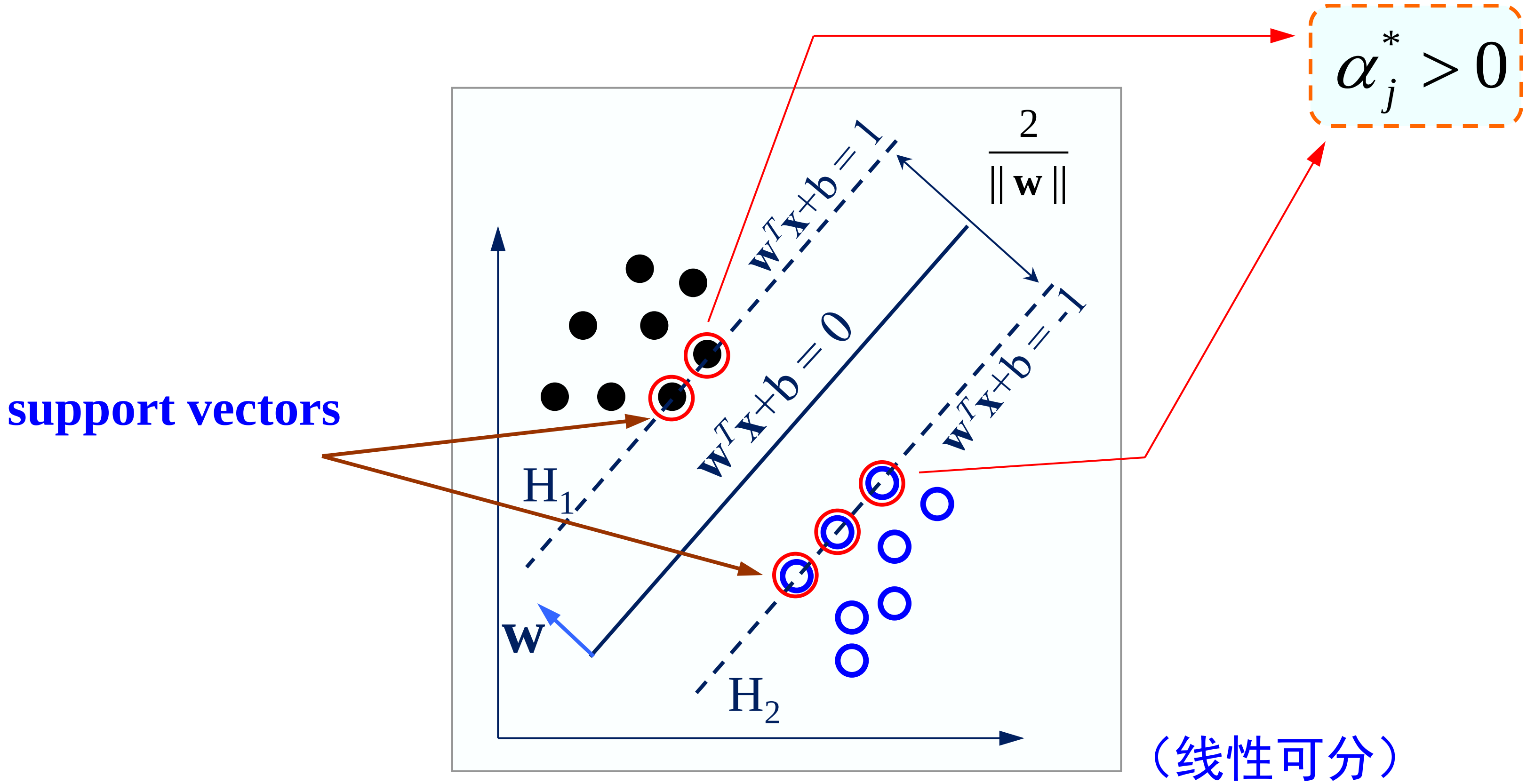
$$\max_{\substack{\alpha\ge 0\\ \mu \ge 0}} \min_{w,b,\xi} L(w, b, \xi, \alpha, \mu)$$

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i \cdot \mathbf{x}_j) - \sum_{i=1}^{n}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{n}\alpha_i y_i = 0;$$

对偶问题 $\quad 0 \le \alpha_i \le C, \quad i = 1, 2, \ldots, n$

✓ 使等式成立的点为**支持向量**： $y_i(\mathbf{w}^T\mathbf{x}_i+b)=1$



（线性可分）

所有样本中，“支持向量”到分类面的几何距离最小。

软间隔支持向量：注意 $\alpha_j^* > 0$ 的样本点均称为支持向量！

图中，第一个点到其正确边界的距离为：$\dfrac{\xi_1}{\|\mathbf{w}\|}$，其它类推。

1： $\alpha^* = C$， $\xi_1 > 1$

2： $\alpha^* = C$， $\xi_2 > 1$

3： $\alpha^* = C$， $0 < \xi_3 < 1$

4： $\alpha^* = C$， $\xi_4 > 1$

5： $\alpha^* = C$， $0 < \xi_5 < 1$

6： $0 < \alpha^* < C$， $\xi_6 = 0$

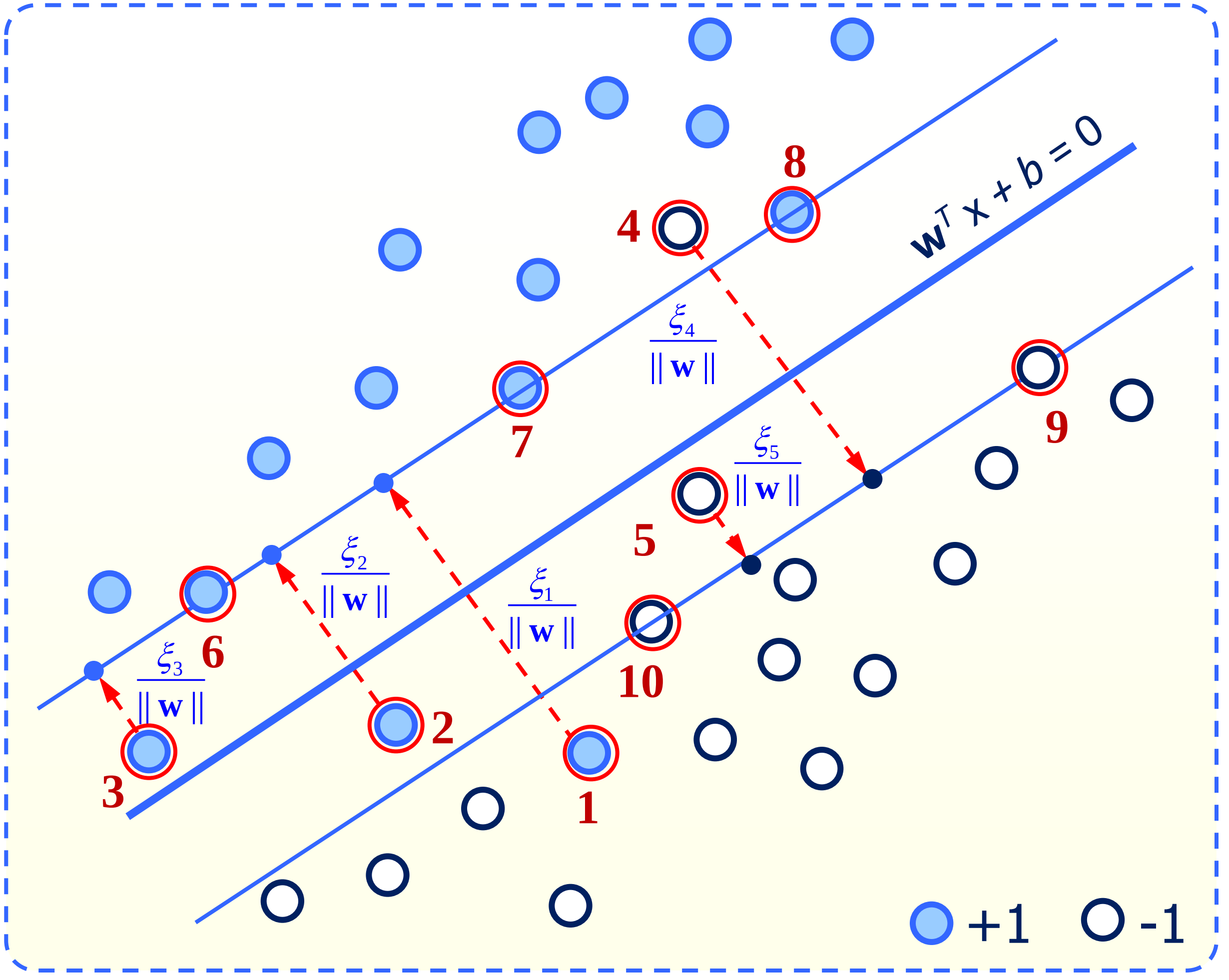
7： $0 < \alpha^* < C$， $\xi_7 = 0$

8： $0 < \alpha^* < C$， $\xi_8 = 0$

9： $0 < \alpha^* < C$， $\xi_9 = 0$

10： $0 < \alpha^* < C$， $\xi_{10} = 0$

10个支持向量



图中带红色圆圈的均表示支持向量

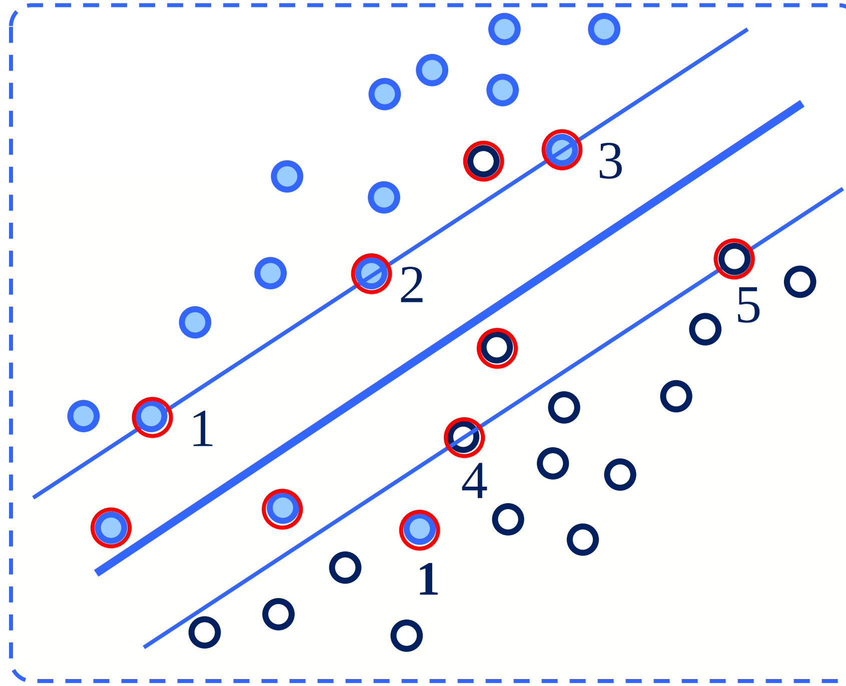
15.3.3 支持向量机解的存在性

- **定理3**
  - 设 $\boldsymbol{\alpha}^* = [\alpha_1^*, \alpha_1^*, \cdots, \alpha_n^*]^T \in R^n$ 是对偶问题的解，则至少存在一个下标 j，有 $0 < \alpha_j^* < C$ ，可按下式求得原始问题的最优解：

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i, \quad b^* = y_j - \sum_{i=1}^{n} \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}_j)$$

- 分类超平面： $\mathbf{w}^* \cdot \mathbf{x} + b^* = 0$
- 分类决策函数： $f(\mathbf{x}) = sign(\mathbf{w}^* \cdot \mathbf{x} + b^*) = sign\left(\sum_{i=1}^{n} \alpha_i^* y_i (\mathbf{x} \cdot \mathbf{x}_i) + b^* \right)$

15.3.3 支持向量机解的存在性



- 偏置b的确定
  - 不唯一

$$b^* = y_j - \sum_{i=1}^{n} y_i \alpha_i^* (\mathbf{x}_i \cdot \mathbf{x}_j), \quad 0 < \alpha_j^* < C$$

在所有符合条件的样本上计算一个b^*，然后取平均：

编程技巧： $$b^* = \frac{1}{\left|\{\alpha_k^* : 0 < \alpha_k^* < C\}\right|} \sum_{0<\alpha_k^*<C} (y_k - \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i \cdot \mathbf{x}_j)$$

$|\{\cdot\}|$: 表示集合的基，也就是集合元素的个数

KSVM：

- 从对偶问题直接实现SVM核化—训练

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)-\sum_{i=1}^{n}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$C \ge \alpha_i \ge 0, \quad i=1,2,\ldots,n$$

$$\Downarrow$$

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

$$s.t. \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$C \ge \alpha_i \ge 0, \quad i=1,2,\ldots,n$$

KSVM：

- 预测（对新数据）

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{n} \alpha_i^* y_i (\mathbf{x} \cdot \mathbf{x}_i) + b^*\right), \quad b^* = y_j - \sum_{i=1}^{n} \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}_j)$$

$$\Downarrow$$

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{n} \alpha_i^* y_i K(\mathbf{x} \cdot \mathbf{x}_i) + b^*\right), \quad b^* = y_j - \sum_{i=1}^{n} \alpha_i^* y_i K(\mathbf{x}_i \cdot \mathbf{x}_j)$$

15.7 Model Selection

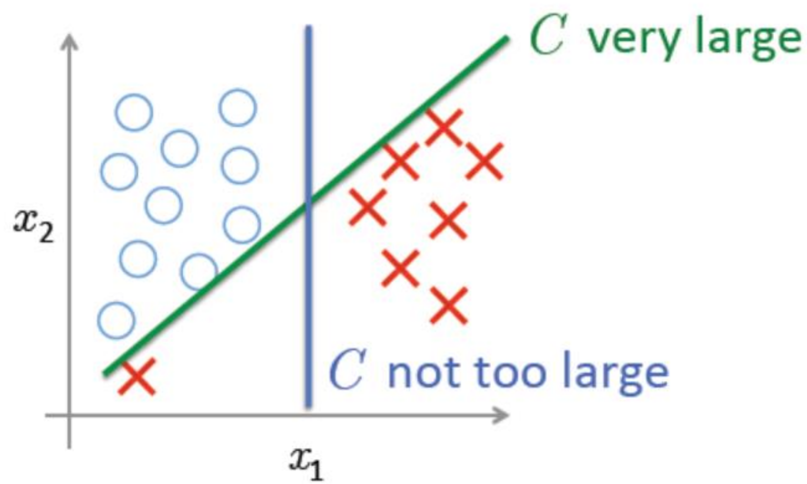
15.7 模型选择

The “C” Problem：

- “C” plays a major role in controlling “over-fitting”
- Finding the “Right” value for “C” is one of the major problems of SVM

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\begin{aligned} s.t. \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \\ & i = 1, 2, \cdots, n \end{aligned}$$

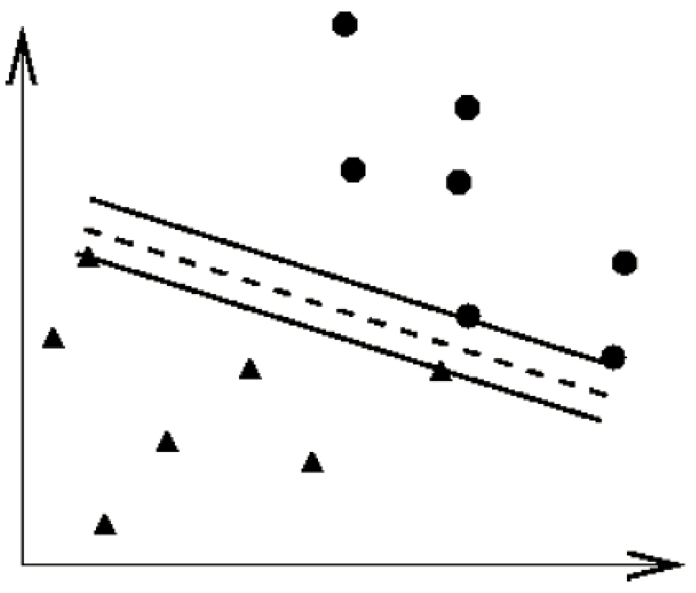


15.7 模型选择

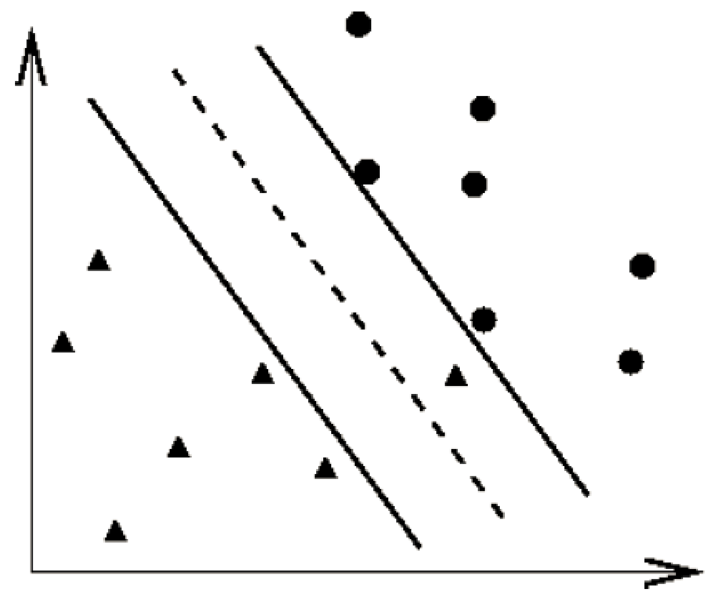
$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\begin{aligned} s.t. \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \ge 1 - \xi_i, \\ & \xi_i \ge 0, \\ & i = 1, 2, \cdots, n \end{aligned}$$

不同*C*值对分类面的影响：



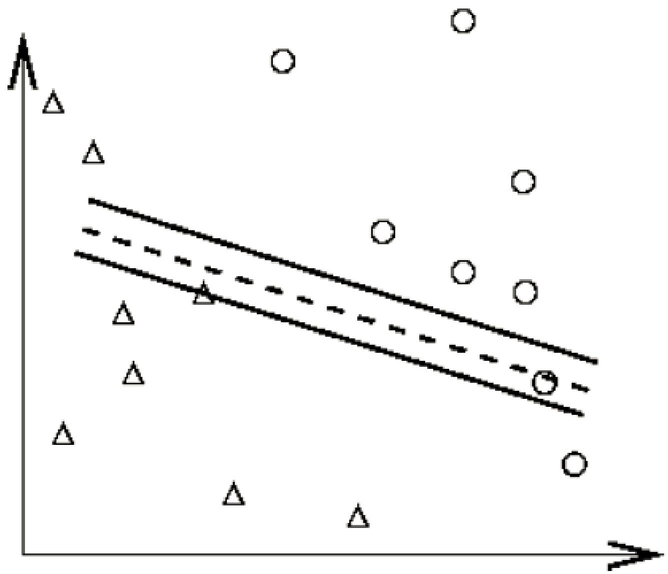
C值较大，更加关心错分样本，倾向于产生没有错分样本的分界面



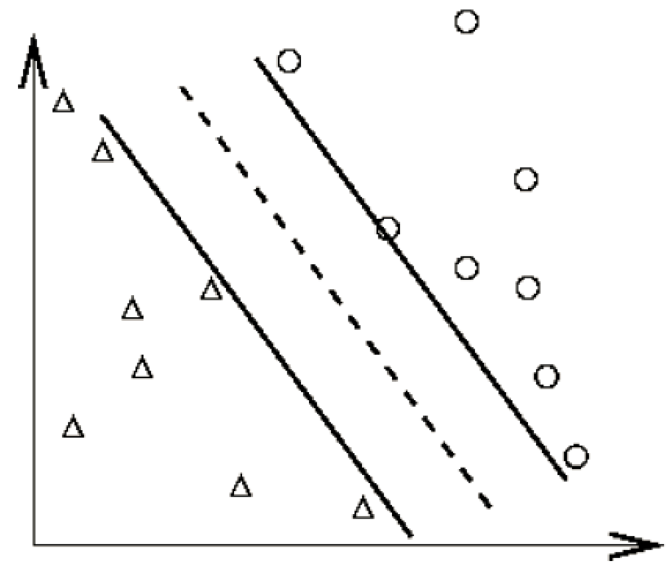
C值较小，更加关心分类间隔，倾向于产生大间隔的分界面

15.7 模型选择

通过选择合适的C值，适当地平衡分类间隔，能减少过拟合风险。



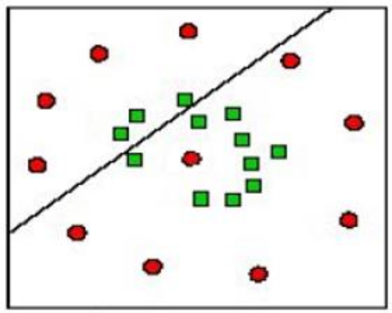
过拟合 (overfitting)



15.7 模型选择

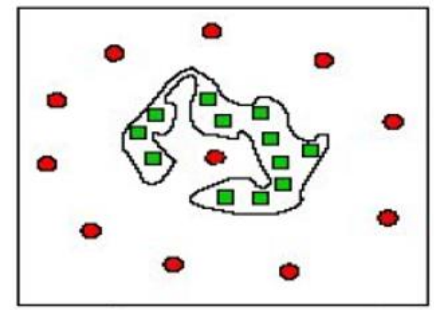
Overfitting and Underfitting：

Under-Fitting

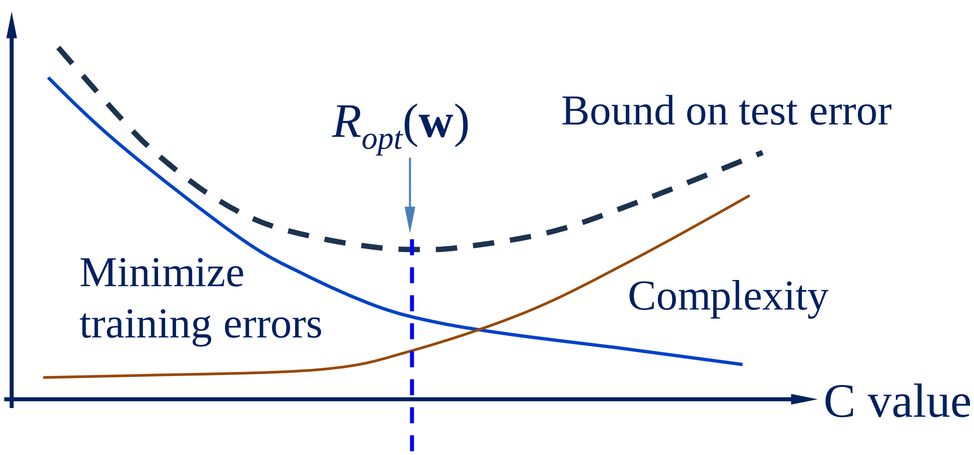


Too much simple

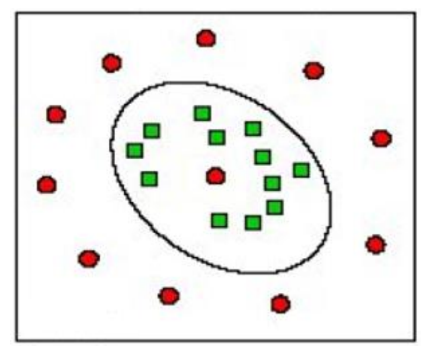
Over-Fitting



Too much complicated



Under-fitting　Best trade-off　Over-fitting



Trade-off

15.7 模型选择

C and Kernel:

- In practice, a Gaussian radial basis or a low degree polynomial kernel is a good start
- Checking which set of parameters (such as C or σ if we choose RBF kernel) are the most appropriate by Cross-Validation (K- fold): （K-折交叉验证）
  - divide randomly all the available training examples into K equal-sized subsets
  - use all but one subset to train the SVM with the chosen parameters
  - use the held-out subset to measure classification error
  - repeat above two steps for each subset
  - average the results to get an estimate of the generalization error of the SVM classifier

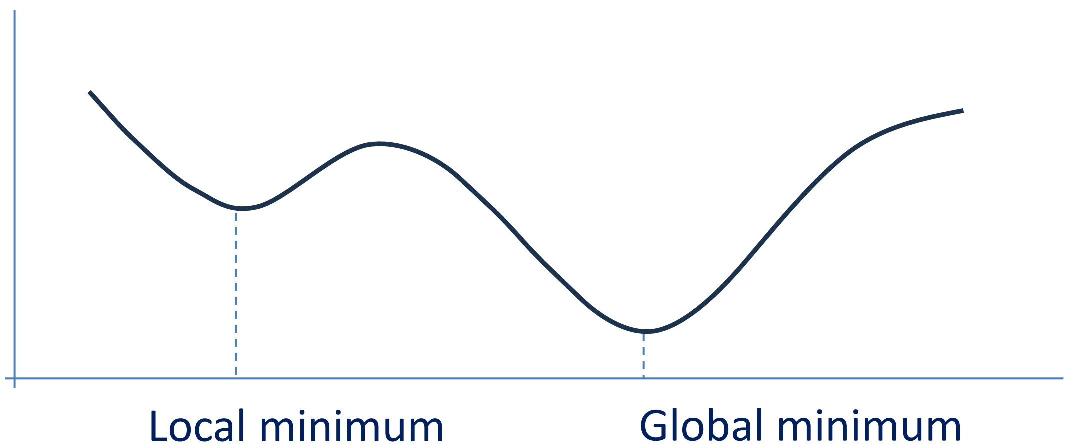
Why cross-validation?

15.8 Optimization

15.8.1 回顾最优化方法

Local and Global Optimal

- Unique solution?
- Depend on initialization?



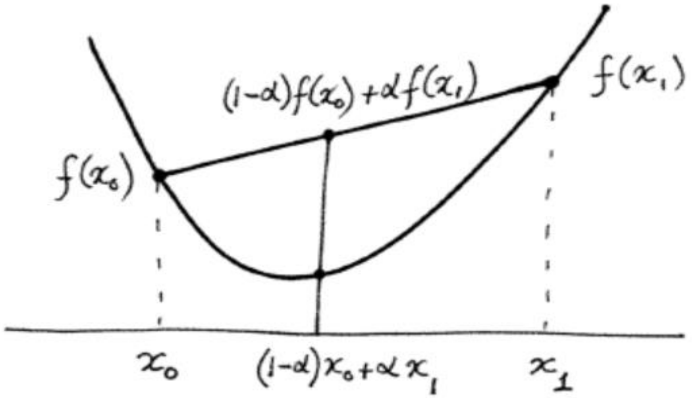
- If the cost function is convex, then a locally optimal point is globally optimal

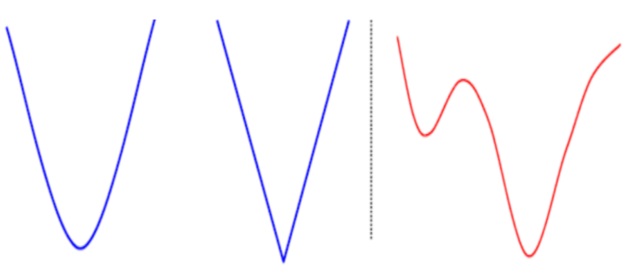
Convex Function:

Give a domain D in R^d, a convex function f: $D \rightarrow R$ is one that satisfies, for any $\mathbf{x}_0$ and $\mathbf{x}_1$ in D:

$$f\left((1-\alpha)\mathbf{x}_0+\alpha\mathbf{x}_1\right)\leq f\left((1-\alpha)\mathbf{x}_0\right)+f\left(\alpha\mathbf{x}_1\right)$$

Line jointing $(\mathbf{x}_0, f(\mathbf{x}_0))$ and $(\mathbf{x}_1, f(\mathbf{x}_1))$ lies above the function (curve or surface)





convex non convex

$$\min_{\mathbf{w},b} \sum_{i=1}^{n}[1-y_i(\mathbf{w}^T\mathbf{x}_i+b)]_+ + \lambda\|\mathbf{w}\|^2$$

SVM is convex: A non-negative sum of convex functions is convex.

Quadratic Programming:

- Many approaches have been developed

  https://en.wikipedia.org/wiki/Quadratic_programming

Name	Brief info
AIMMS	A software system for modeling and solving optimization and scheduling-type problems
ALGLIB	Dual licensed (GPL/proprietary) numerical library (C++, .NET).
AMPL	A popular modeling language for large-scale mathematical optimization.
APMonitor	Modeling and optimization suite for LP, QP, NLP, MILP, MINLP, and DAE systems in MATLAB and Python.
Artelys Knitro	An Integrated Package for Nonlinear Optimization
CGAL	An open source computational geometry package which includes a quadratic programming solver.
CPLEX	Popular solver with an API (C, C++, Java, .Net, Python, Matlab and R). Free for academics.
Excel Solver Function	A nonlinear solver adjusted to spreadsheets in which function evaluations are based on the recalculating cells. Basic version available as a standard add-on for Excel.
GAMS	A high-level modeling system for mathematical optimization
GNU Octave	A free (its licence is GPLv3) general-purpose and matrix-oriented programming-language for numerical computing, similar to MATLAB. Quadratic programming in GNU Octave is available via its qp command
HiGHS	Open-source software for solving linear programming (LP), mixed-integer programming (MIP), and convex quadratic programming (QP) models
IMSL	A set of mathematical and statistical functions that programmers can embed into their software applications.

… …

Quadratic Programming

- Most are "interior-point" methods: 内点法
  - 二次规划的内点法是一种用于解决具有等式和不等式约束的优化问题的方法。
  - 内点法通过将约束条件纳入目标函数，逐步求解，从而避免了对可行域边界的处理。
- For SVM, sequential minimal optimization (SMO，序列最小最优化算法, 1998, by Platt) seems to be the most popular：
  - A QP with two variables is trivial to solve （求解起来很简单）
  - Each iteration of SMO picks a pair of (α_i, α_j) and solve the QP with these two variables; repeat until convergence
- In practice, we can just regard the QP solver as a "blackbox" without bothering how it works
  - considering BP in deep learning

15.8.2 Speed up the training for SVM

- **Hint 1: Only SVs will influence the decision boundary**

$$\max_{\boldsymbol{\alpha}} \quad -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i \cdot \mathbf{x}_j) + \sum_{i=1}^{n}\alpha_i$$

$$s.t. \quad \boxed{\sum_{i=1}^{n}\alpha_i y_i = 0} \dashrightarrow \alpha_1 = -y_1\sum_{i=2}^{n}\alpha_i y_i$$

$$C \geq \alpha_i \geq 0, \quad i = 1, 2, \ldots, n$$

如果只有(α_1,α_2)两个变量待求，在α_2确定的情况下，可得α_1

- ✓ Training SVM only on part data in each iteration
- ✓ keeping remaining the SVs and repeating training by adding new data sequentially until SVs don’t change
- ✓ 整个SMO算法包含两个部分（Chunk strategy，组块策略）：
  - 求解两个变量的二次规划的解析解
  - 选择变量的启发式方法

15.8.2 Speed up the training for SVM

- **Hint 2: When the optimum is achieved, KKT conditions are satisfied**

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$
$$s.t. \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \ge 1 - \xi_i,$$
$$\xi_i \ge 0,$$
$$i = 1, 2, \cdots, n$$

$$\begin{cases} \alpha_i = 0 & \Rightarrow y_i f(x_i) \ge 1 \Rightarrow \textit{Samples outside the boundary} \\ 0 < \alpha_i < C & \Rightarrow y_i f(x_i) = 1 \Rightarrow \textit{Samples on the boundary} \\ \alpha_i = C & \Rightarrow y_i f(x_i) \le 1 \Rightarrow \textit{Samples within the boundary} \end{cases}$$

- Sequential Minimal Optimization (SMO)：序列最小最优化算法
  - Each time, choose two samples violating the KKT condition and updating the weights until all the points satisfied KKT condition (每次选择两个)
  - J. C. Platt, *Fast Training of Support Vector Machines Using Sequential Minimal Optimization*, Advances in kernel methods: support vector learning, 1999.

SMO:

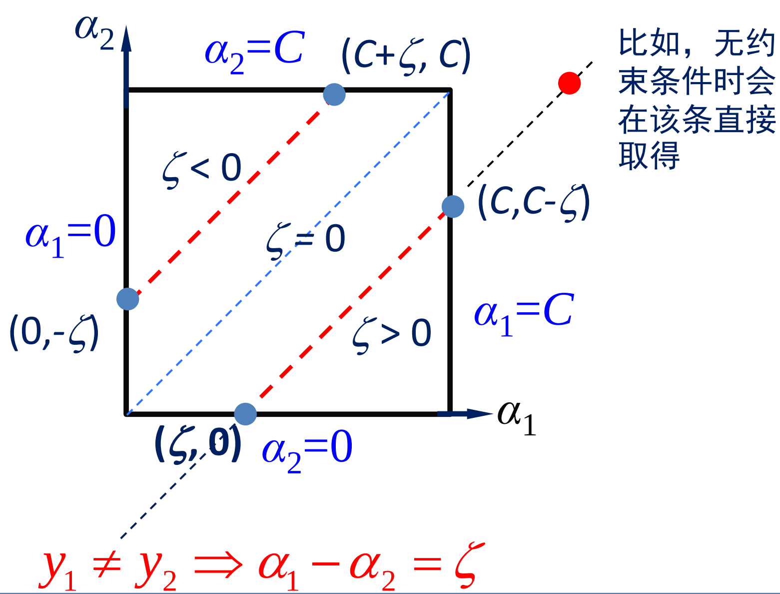
- Constraints on the Lagrange Multipliers (select two):

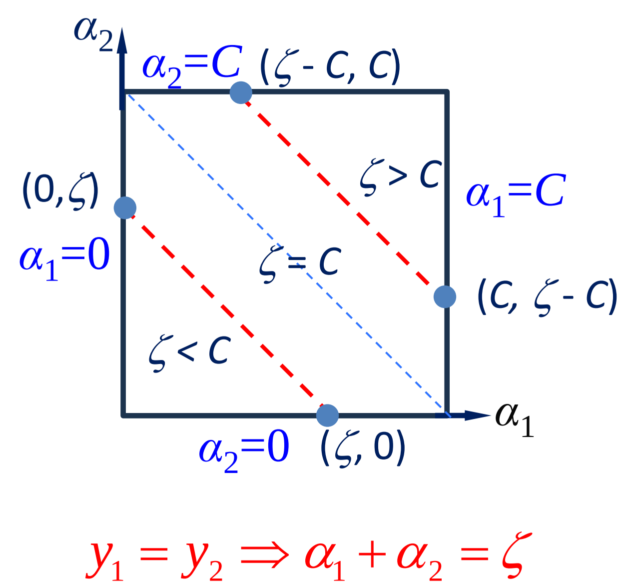
$$\min_{\alpha_1,\alpha_2} \quad g(\alpha_1,\alpha_2)=\frac{1}{2}K_{11}\alpha_1^2+\frac{1}{2}K_{22}\alpha_2^2+y_1y_2K_{12}\alpha_1\alpha_2$$

$$-(\alpha_1+\alpha_2)+y_1\alpha_1\sum_{i=3}^{n}\alpha_i y_i K_{i1}+y_2\alpha_2\sum_{i=3}^{n}\alpha_i y_i K_{i2}+const$$

$$s.t. \qquad y_1\alpha_1+y_2\alpha_2=-\sum_{i=3}^{n}\alpha_i y_i=\zeta$$

$$C\geq\alpha_i\geq 0, \quad i=1,2$$





细节求解：李航, 统计机器学习, 清华大学出版社（第一版）， P. 124-131

15.8.2 Speed up the training for SVM

SMO: 选取两个Langrange乘子变量的启发式规则：

- 第一个Langrange乘子α_1的选择：
  - 任何违反KKT条件的乘子(或样本)都是合法的第一个Langrange乘子。
  - 第一个Langrange乘子的选择构成SMO算法的外层循环。
    - 检查违反KKT条件最严重的 (within ε ,e.g. 10^{-3}).
    - The outer loop then goes back and iterates over the entire training set.
- 第二个Langrange乘子α_2的选择：
  - 与第一个乘子的结合，应该使第二个乘子的迭代步长较大（即使α_2的变化最大）
- 很多文献改进方法：如何选择两个乘子变量
  - Maximal violating pair
  - Second order information

15.9 Multi-Class SVM

15.9.1 Multi-class Extension

- SVMs can only handle two-class outputs.（支持向量机只能处理两类分类问题）
- What can be done?
  - One answer: with output C labels, learn C SVM's：一对多
    - SVM 1 learns "Output==1" vs "Output != 1"
    - SVM 2 learns "Output==2" vs "Output != 2"
    - :
    - SVM C learns "Output==C" vs "Output != C"
- Then to predict the output for a new input, just predict with each SVM and find out which one puts the prediction the furthest into the positive region.

15.9.1 Multi-class Extension

Two families of methods for Multi-Class SVM:

- **Multiple Binary Problem**
  - One-vs-All:
    - C quadratic programming problems are solved
  - One-vs-One:
    - $C(C\text{-}1)/2$ quadratic programming
  - DAGSVM:
    - $C(C\text{-}1)/2$ quadratic programming
  - Half-vs-Half:
    - 多个类别（比如一半）组成一组，然后构建一棵DAG
  - ECOC （Error-correcting output codes, 纠错输出码）
    - 引入停用类
  - LatticeSVM:
    - Z. Liu, L. Jin, LatticeSVM—A New Method for Multi-Class Support Vector Machines, *IJCNN 2008*.

15.9.1 Multi-class Extension

Two families of methods for Multi-Class SVM:

- **Single Machine**
  - Consider all classes at once, result in a much larger optimization problem in one step.
    - ✓ K. Crammer, Y. Singer, On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines, *JMLR 2001*.
    - ✓ J. Weston, C. Watkins, Multi-Class Support Vector Machines, *Technical Report 1998*.
- C.-W. Hsu, C. Lin, A Comparison of Methods for Multiclass Support Vector Machines, IEEE Trans. Neural Networks, 2002.

15.9.2 DAGSVM

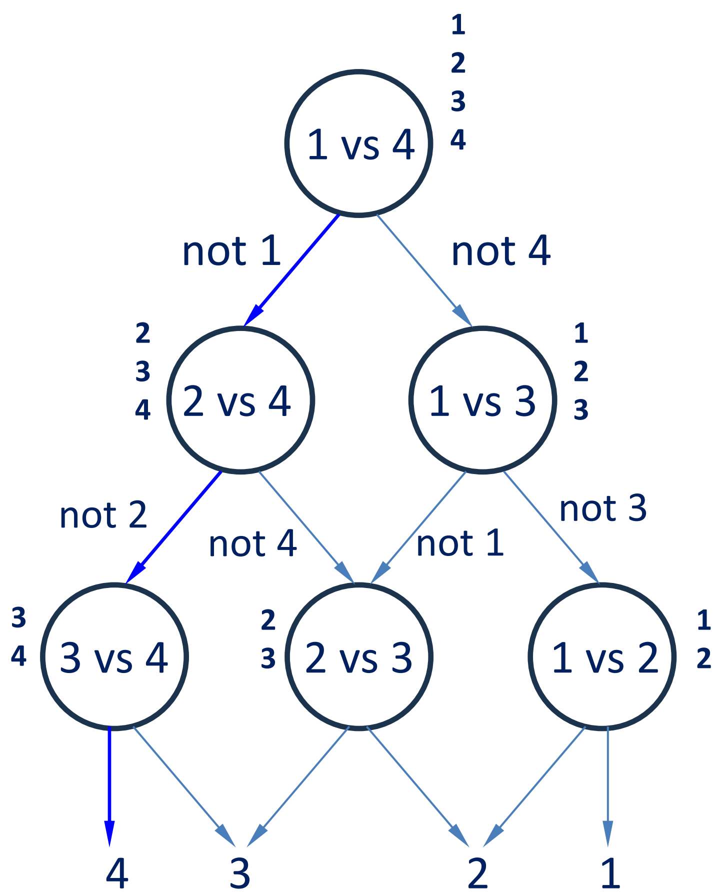
DAGSVM（Directed Acyclic Graph Support Vector Machine）通过构建一个有向无环图（DAG）来表示多个分类器，每个节点代表一个二分类器，节点之间的边表示类别的决策边界。这种方法可以有效地将多类分类问题转化为一系列的二分类问题，从而简化问题的复杂度。

- 构建DAG图：对于有C个类别的分类问题，构建一个有$C(C-1)/2$个节点的DAG图。每个节点代表一个二分类问题，节点之间的边表示类别之间的决策边界。核心：1 VS 1
- 训练二分类器：在每个节点上训练一个二分类器，将当前节点代表的两个类别分开。
- 预测：在进行预测时，从根节点开始，根据每个节点的分类结果逐步向下决策，直到到达叶子节点，最终确定样本的类别。
- The choice of the class order in the list is arbitrary.

J.C. Platt et al, Large Margin DAGs for Multiclass Classification, *NIPS 2000*.

15.9.2 DAGSVM

DAGSVM：通过构建一个有向无环图（DAG）来表示多个分类器，每个节点代表一个二分类器，节点之间的边表示类别的决策边界。



- ✓ 在训练阶段，仍然要构建 $C(C-1)/2$个SVM
- ✓ 在测试阶段，则不必运行 $C(C-1)/2$个SVM，沿树进行，节省预测时间

J.C. Platt et al, Large Margin DAGs for Multiclass Classification, *NIPS 2000*.

15.9.2 DAGSVM

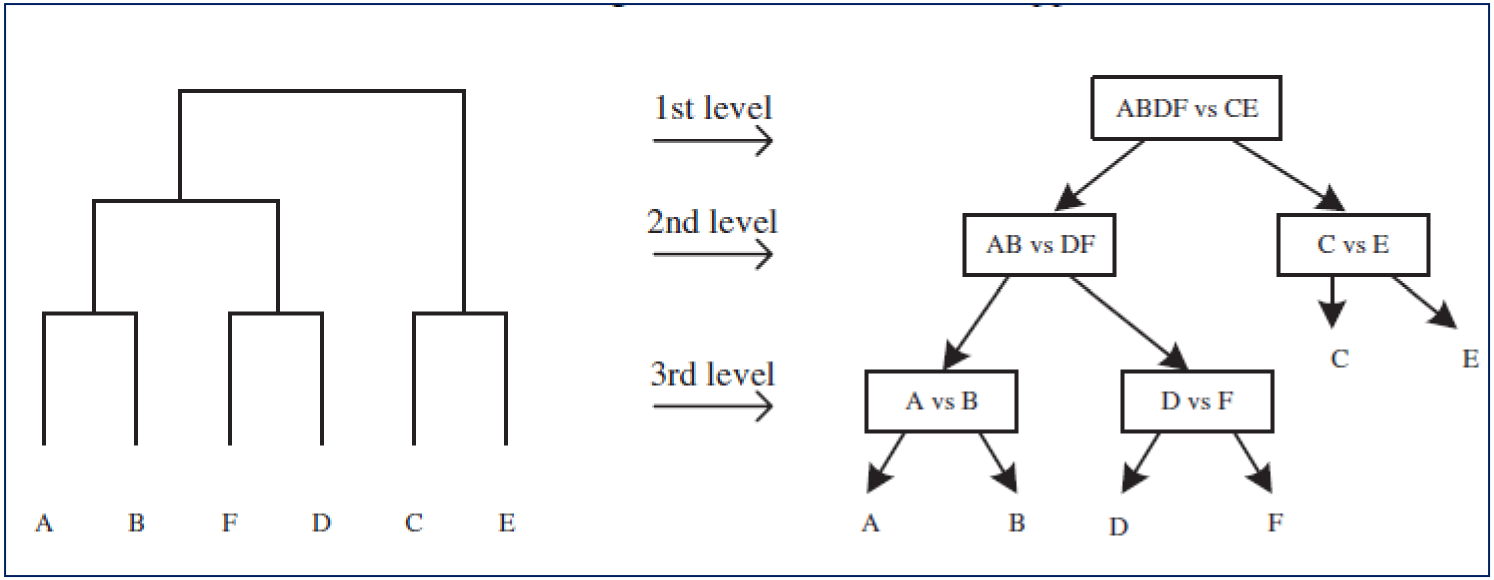
DAGSVM 通过构建一个有向无环图（DAG）来表示多个分类器，每个节点代表一个二分类器，节点之间的边表示类别的决策边界。

Algorithm:

1. Create a list of class labels $L=(1,2,\ldots,M)$ (arbitrary order)
2. Evaluates the sample with the binary SVM that corresponds to the first and last element in list L, the loser class index is eliminated from the list.（失败的类别逐渐被过滤掉）
3. After M-1 evaluations, the last label is the answer.

15.9.3 Half-vs-Half

可以利用K-means或层级聚类方法来对类别进行分组



- ✓ H. Lei, V. Govindaraju, Half-Against-Half Multi-Class Support Vector Machines, *MCS 2005*.
- ✓ V. Vural, J.G. Dy, A Hierarchical Method for Multi-Class Support Vector Machines, *ICML 2004*.

15.9.3 Half-vs-Half

- How to divide the data into two subsets hierarchically?
  - ✓ Two subsets have the largest margin.
  - ✓ Two subsets have minimum number of SVs
  - ✓ k-means division
  - ✓ Spherical shells
  - ✓ Balanced subsets
  - ✓ Hierarchy clustering

常用的二分组方法

- Testing speed: worst case: M-1 evaluations
  - ✓ 1-v-1: M(M-1)/2
  - ✓ 1-v-a: M
  - ✓ DAGSVM: M-1

15.9.4 ECOC

- 类别划分通过编码矩阵（coding matrix）进行来指定。编码矩阵有多种形式。常见的有二元码、三元码。
  - 二元码：将每个类别，要么指定为正类，要么指定为负类；
  - 三元码：对于一个类别，既可以充当正类，也可以充当负类，还可以充当“停用类”。

15.9.4 ECOC

- **ECOC二元码示意图（以四类分类问题为例）**

四类分类问题

分类器:	f_1	f_2	f_3	f_4	f_5	海明距离	欧氏距离
C_1	-1	+1	-1	+1	+1	3	$2\sqrt{3}$
C_2	+1	-1	-1	+1	-1	4	4
C_3	-1	+1	+1	-1	+1	1	2
C_4	-1	-1	+1	+1	-1	2	$2\sqrt{2}$
测试示例	-1	-1	+1	-1	+1		

15.9.4 ECOC

- **ECOC三元码示意图** (0:表示不考虑此类):

四类样例

分类器:	f_1	f_2	f_3	f_4	f_5	f_6	f_7	海明距离	欧氏距离
C_1	-1	-1	+1	+1	-1	+1	+1	4	4
C_2	-1	0	0	0	+1	-1	0	4	2
C_3	+1	+1	-1	-1	-1	+1	-1	5	$2\sqrt{5}$
C_4	-1	+1	0	+1	-1	0	+1	3	$\sqrt{10}$
测试示例	-1	+1	+1	-1	+1	-1	+1		

15.9.4 ECOC

- **ECOC为什么会纠错**
  - 在测试阶段，ECOC编码对分类器的错误有一定的容忍和修正能力。
    - 假设在上述的二元码示例中，正确的预测编码应该为(-1,+1,+1,-1,+1)，即属于C_3类。但多分类器的实际输出为(-1,-1,+1,-1,+1)。最后仍被分C_3类。
  - 对同一个学习任务，ECOC编码越长，纠错能力就越强。
  - 对同等长度的编码，理论上讲，任意两个类别之间的编码越远，则纠错能力越强。
  - 在编码长度较小时，可以据此设计最优编码。但对于长编码，最优编码设计是一个NP难问题。
  - 但并不是编码的理论性能越好，分类性能就越好，因为机器学习问题涉及很多因素。
  - 编码越长，则分类器越多，训练时间越长。

15.9.5 其它方法（略）

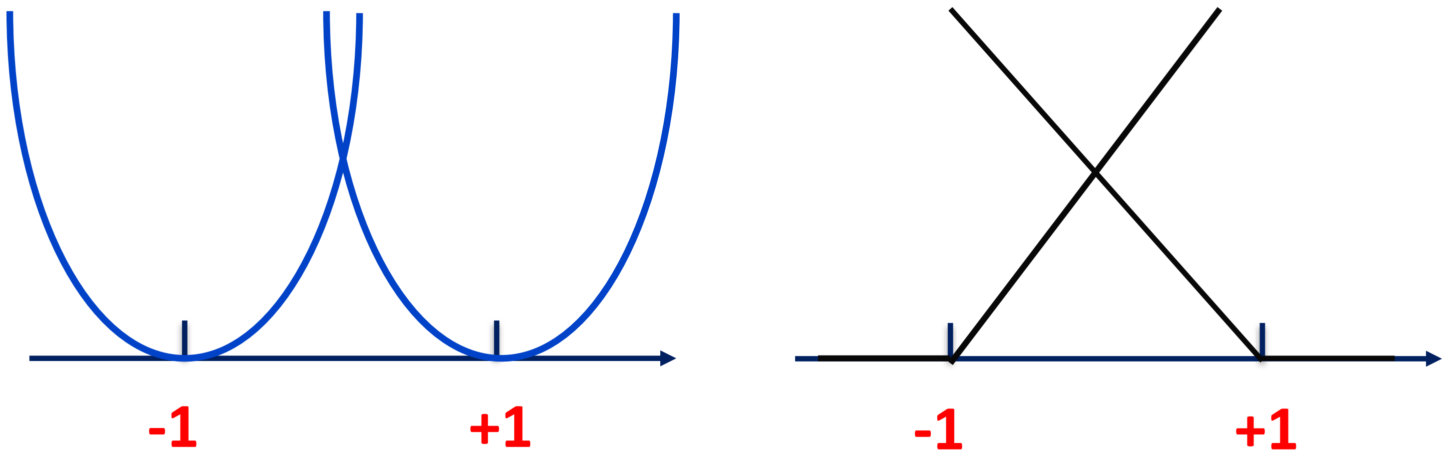
- ✓ LatticeSVM：通过引入“格子算法”，解决了传统SVM在多类分类问题上的不足。
  - LatticeSVM通过静态候选技术和格子结构，有效地处理了多类问题，并在存储空间和运算速度上具有明显优势
  - Z. Liu, L. Jin, LatticeSVM—A New Method for Multi-Class Support Vector Machines, *IJCNN 2008.*
  - … …

15.10 支持向量机与判别最小二乘法

A more natural way to solve multi-class problems is to construct a decision function by considering all classes at once. Result in a much larger optimization problem in one step.

15.10.1 基础知识回顾

Least Square Loss vs Hinge Loss：



15.10.1 基础知识回顾

Linear Regression：

- Data set $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^m$
- Destination set $\{\mathbf{y}_i\}_{i=1}^n \subset \mathbb{R}^c$
- Linear regression can be defined as:

$$\min_{\mathbf{W},\mathbf{b}} \sum_{i=1}^{n} \left\| \mathbf{W}^T \mathbf{x}_i + \mathbf{b} - \mathbf{y}_i \right\|_2^2 + \lambda \left\| \mathbf{W} \right\|_F^2$$

$$\mathbf{W} \in \mathbb{R}^{m\times c} \quad \mathbf{b} \in \mathbb{R}^c$$

- Vector L_2 norm $||\cdot||_2$
- Matrix Frobenius norm $||\cdot||_F$
- The simplest (multi-class) classifier !

15.10.1 基础知识回顾

Different Loss Functions：

- Data: $\{x_i\}_{i=1}^N \quad \{t_i\}_{i=1}^N \quad t_i \in \{1, -1\}$
- Classifier: $y = \text{sign}(w^\top \phi(x) + b).$
- Margin variable: $z = ty$
  - **Large z → small loss**
- Loss Functions:
  - 0-1 loss: $l_{mer}(z_i) = ||(\text{-}z_i)_+||_0$
  - Hinge loss: $l_{hinge} = \{(1 - z_i)_+\}^p$ (p=1 or 2)
  - Logistic regression: $l_{log}=\log_2[1+\exp(\text{-}\ z_i)]$
  - Least squares: $l_{ls}=(1\text{-}z_i)^2$
  - Adaboost: $l_{exp} = \exp(\text{-}z_i).$

15.10.2 Discriminative LSR

- Data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n \quad y_i \in \{1, 2, \ldots, c\}$
- Class $\mathbf{f}_j = [0, \ldots, 0, 1, 0, \ldots, 0]^T \in \mathbb{R}^c$

- Training sample fitting

$$\mathbf{f}_{y_i} \approx \mathbf{W}^T \mathbf{x}_i + \mathbf{t}, \qquad i = 1, 2, \ldots, n$$

- Let

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times m}$$

$$\mathbf{Y} = [\mathbf{f}_{y_1}, \mathbf{f}_{y_2}, \ldots, \mathbf{f}_{y_n}]^T \in \mathbb{R}^{n \times c}$$

- Training sample fitting

$$\mathbf{X}\mathbf{W} + \mathbf{e}_n \mathbf{t}^T \approx \mathbf{Y}$$

Shiming Xiang et al. Discriminative Least Squares Regression for Multiclass Classification and Feature Selection, IEEE T-NNLS, 2012.

15.10.2 Discriminative LSR

$$\mathbf{XW} + \mathbf{e}_n \mathbf{t}^T \approx \mathbf{Y}$$

- Each column of **Y** → binary regression with +1/0
- How to enlarge the margin?
- Define a new matrix $\mathbf{B} \in \mathbb{R}^{n \times c}$

$$B_{ij} = \begin{cases} +1, & \text{if } y_i = j \\ -1, & \text{otherwise.} \end{cases}$$

- Each element in B corresponds to a dragging direction
- Target re-definition

$$\mathbf{M} \in \mathbb{R}^{n \times c}$$

$$\mathbf{XW} + \mathbf{e}_n \mathbf{t}^T - (\mathbf{Y} + \mathbf{B} \odot \mathbf{M}) \approx \mathbf{0}$$

15.10.2 Discriminative LSR

- Optimize three parts of parameters

$$\min_{\mathbf{W},\mathbf{t},\mathbf{M}} \quad ||\mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{t}^T - \mathbf{Y} - \mathbf{B} \odot \mathbf{M}||_F^2 + \lambda||\mathbf{W}||_F^2$$
$$\text{s.t.} \quad \mathbf{M} \geq \mathbf{0}$$

- OPT1:
  - Fix **M** and solve **W**, **t** → unconstrained convex QP→解析解
- OPT2:
  - Fix **W**, **t** and solve **M** → very simple elementwise solution

15.10.2 Discriminative LSR

OPT1: Fix M

$$\min_{\mathbf{W},\mathbf{t},\mathbf{M}} \; ||\mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{t}^T - \mathbf{Y} - \mathbf{B} \odot \mathbf{M}||_F^2 + \lambda||\mathbf{W}||_F^2$$
$$\text{s.t.} \quad \mathbf{M} \geq \mathbf{0}$$

- Fix M, and let $\mathbf{R} = \mathbf{Y} + \mathbf{B} \odot \mathbf{M} \in \mathbb{R}^{n \times c}$
- We have:

$$\mathbf{W} = (\mathbf{X}^T\mathbf{H}\mathbf{X} + \lambda\mathbf{I}_m)^{-1}\mathbf{X}^T\mathbf{H}\mathbf{R}$$

$$\mathbf{t} = \frac{(\mathbf{R}^T\mathbf{e}_n - \mathbf{W}^T\mathbf{X}^T\mathbf{e}_n)}{n}$$

15.10.2 Discriminative LSR

OPT1: Fix M

$$\min_{\mathbf{W},\mathbf{t},\mathbf{M}} \quad ||\mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{t}^T - \mathbf{Y} - \mathbf{B} \odot \mathbf{M}||_F^2 + \lambda||\mathbf{W}||_F^2$$
$$\text{s.t.} \quad \mathbf{M} \geq \mathbf{0}$$

$$\frac{\partial g(\mathbf{W},\mathbf{t})}{\partial \mathbf{t}} = \mathbf{0} \Rightarrow \mathbf{W}^T\mathbf{X}^T\mathbf{e}_n + \mathbf{t}\mathbf{e}_n^T\mathbf{e}_n - \mathbf{R}^T\mathbf{e}_n = \mathbf{0}$$

$$\Rightarrow \mathbf{t} = \frac{(\mathbf{R}^T\mathbf{e}_n - \mathbf{W}^T\mathbf{X}^T\mathbf{e}_n)}{n}.$$

$$\frac{\partial g(\mathbf{W},\mathbf{t})}{\partial \mathbf{W}} = \mathbf{0}$$

$$\Rightarrow \mathbf{X}^T\left(\mathbf{X}\mathbf{W} + \frac{1}{n}\mathbf{e}_n\mathbf{e}_n^T\mathbf{R} - \frac{1}{n}\mathbf{e}_n\mathbf{e}_n^T\mathbf{X}\mathbf{W} - \mathbf{R}\right) + \lambda\mathbf{W} = \mathbf{0}$$

$$\Rightarrow \mathbf{X}^T\left(\mathbf{I}_n - \frac{1}{n}\mathbf{e}_n\mathbf{e}_n^T\right)\mathbf{X}\mathbf{W} - \mathbf{X}^T\left(\mathbf{I}_n - \frac{1}{n}\mathbf{e}_n\mathbf{e}_n^T\right)\mathbf{R} + \lambda\mathbf{W} = \mathbf{0}$$

$$\Rightarrow \mathbf{W} = (\mathbf{X}^T\mathbf{H}\mathbf{X} + \lambda\mathbf{I}_m)^{-1}\mathbf{X}^T\mathbf{H}\mathbf{R}.$$

15.10.2 Discriminative LSR

OPT2: Fix W and t

$$\min_{\mathbf{W},\mathbf{t},\mathbf{M}} \quad ||\mathbf{XW}+\mathbf{e}_n\mathbf{t}^T-\mathbf{Y}-\mathbf{B}\odot\mathbf{M}||_F^2+\lambda||\mathbf{W}||_F^2$$
$$\text{s.t.} \quad \mathbf{M}\geq\mathbf{0}$$

$$\mathbf{P}=\mathbf{XW}+\mathbf{e}_n\mathbf{t}^T-\mathbf{Y}$$

$$\min_{\mathbf{M}} \quad ||\mathbf{P}-\mathbf{B}\odot\mathbf{M}||_F^2, \quad \text{s.t.} \quad \mathbf{M}\geq\mathbf{0}.$$

$$\min_{M_{ij}} \quad (P_{ij}-B_{ij}M_{ij})^2, \quad \text{s.t.} \quad M_{ij}\geq 0$$

$$M_{ij}=\max(B_{ij}P_{ij},0).$$

$$\mathbf{M}=\max(\mathbf{B}\odot\mathbf{P},\mathbf{0}).$$

15.10.2 Discriminative LSR

Algorithm of DLSR

$$\min_{\mathbf{W},\mathbf{t},\mathbf{M}} \ ||\mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{t}^T - \mathbf{Y} - \mathbf{B} \odot \mathbf{M}||_F^2 + \lambda||\mathbf{W}||_F^2 \quad \text{s.t.} \ \ \mathbf{M} \geq \mathbf{0}$$

Algorithm 1 *DLSR*

Input: n data points $\{\mathbf{x}_i\}_{i=1}^n$ in $\mathbb{R}^m$, and their corresponding class labels $\{y_i\}_{i=1}^n \subset \{1, 2, \ldots, c\}$; parameter λ in (7); and maximum number of iterations T.

Allocate $\mathbf{M}$, $\mathbf{W}$, $\mathbf{W}_0$, $\mathbf{t}$, and $\mathbf{t}_0$.
$\mathbf{M} = \mathbf{0}$, $\mathbf{W}_0 = \mathbf{0}$, and $\mathbf{t}_0 = \mathbf{0}$.
Construct $\mathbf{X}$ and $\mathbf{Y}$ in (4), and $\mathbf{B}$ according to (5).
Let $\mathbf{U} = (\mathbf{X}^T\mathbf{H}\mathbf{X} + \lambda\mathbf{I}_m)^{-1}\mathbf{X}^T\mathbf{H}$.
Let $k = 1$.
while $k < T$ **do**
  $\mathbf{R} = \mathbf{Y} + \mathbf{B} \odot \mathbf{M}$.
  $\mathbf{W} = \mathbf{U}\mathbf{R}, \quad \mathbf{t} = \frac{1}{n}\mathbf{R}^T\mathbf{e}_n - \frac{1}{n}\mathbf{W}^T\mathbf{X}^T\mathbf{e}_n$.
  $\mathbf{P} = \mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{t}^T - \mathbf{Y}$.
  $\mathbf{M} = \max(\mathbf{B} \odot \mathbf{P}, \mathbf{0})$.
  if $(||\mathbf{W} - \mathbf{W}_0||_F^2 + ||\mathbf{t} - \mathbf{t}_0||_2^2) < 10^{-4}$, **then**
    Stop.
  end if
  $\mathbf{W}_0 = \mathbf{W}, \quad \mathbf{t}_0 = \mathbf{t}, \quad k = k + 1$.
end while
Output $\mathbf{W}$ and $\mathbf{t}$.

15.10.2 Discriminative LSR

Re-think DLSR

$$\mathbf{W} \in \mathbb{R}^{m \times c} \qquad B_{ij} = \begin{cases} +1, & \text{if } y_i = j \\ -1, & \text{otherwise.} \end{cases}$$

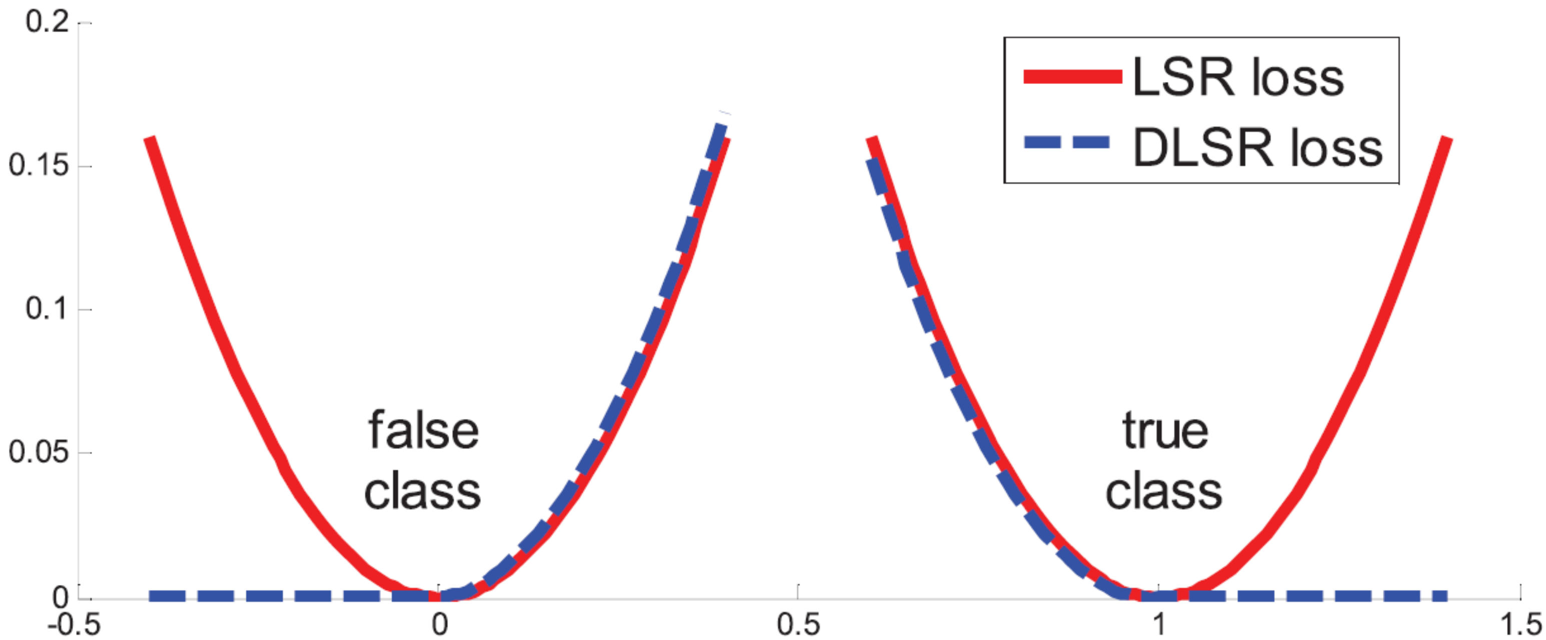
$$\mathbf{M} \geq \mathbf{0}$$

$$\boxed{\begin{aligned} \min_{\mathbf{W},\mathbf{t},\mathbf{M}} \quad & \|\mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{t}^T - \mathbf{Y} - \mathbf{B} \odot \mathbf{M}\|_F^2 + \lambda\|\mathbf{W}\|_F^2 \\ \text{s.t.} \quad & \mathbf{M} \geq \mathbf{0} \end{aligned}}$$

- Another formalization of **one-vs-all SVM** with **squared hinge loss** !

15.10.2 Discriminative LSR

Re-think DLSR



15.10.3 Retargeted LSR

- LSR: use exact 0-1 as target
- DLSR: use relaxed 0-1 as target
- ReLSR: use any (learned) number as target
  - Totally learn the regression targets from data
  - With flexible large margin constraints

$\mathbf{y}$	regression result	margin≥ 1	LSR	DLSR	ReLSR
$[1,0,0]$	$[1.5,0,0]$	Yes	loss=0.25	loss=0.00 target=$[1.5,0,0]$	loss=0.00 target=$[1.5,0,0]$
$[1,0,0]$	$[1,-0.5,-0.5]$	Yes	loss=0.50	loss=0.00 target=$[1,-0.5,-0.5]$	loss=0.00 target=$[1,-0.5,-0.5]$
$[0,1,0]$	$[0.5,1.5,0.5]$	Yes	loss=0.75	loss=0.50 target=$[0,1.5,0]$	loss=0.00 target=$[0.5,1.5,0.5]$
$[0,1,0]$	$[-0.5,1.5,0.5]$	Yes	loss=0.75	loss=0.25 target=$[-0.5,1.5,0]$	loss=0.00 target=$[-0.5,1.5,0.5]$
$[0,0,1]$	$[0.2,0.2,0.8]$	No	loss=0.12	loss=0.12 target=$[0,0,1]$	loss=0.11 target=$[0.0667,0.0667,1.0667]$
$[0,0,1]$	$[-0.2,0.2,0.6]$	No	loss=0.24	loss=0.20 target=$[-0.2,0,1]$	loss=0.18 target=$[-0.2,-0.1,0.9]$

Xu-Yao Zhang et al. Retargeted Least Squares Regression Algorithm. IEEE Transactions on Neural Network and Learning Systems (T-NNLS), 2015.

15.10.3 Retargeted LSR

- Target Matrix $\mathbf{T} \in \mathbb{R}^{n\times c}$

$$\min_{\mathbf{W},\mathbf{b},\mathbf{T}} \|\mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{b}^\top - \mathbf{T}\|_F^2 + \beta\|\mathbf{W}\|_F^2$$
$$\text{s.t.} \quad T_{i,y_i} - \max_{j\neq y_i} T_{i,j} \geq 1, \quad i = 1, 2, \ldots, n.$$

- The constraint flexibility: LSR < DLSR < ReLSR
- LSR/DLSR → decomposed into c independent sub-problems (one-against-rest)
- Because of **T**, ReLSR is a single and compact machine for multiclass classification

15.10.3 Retargeted LSR

OPT1: Regression

$$\boxed{\begin{aligned} &\min_{\mathbf{W},\mathbf{b},\mathbf{T}} \|\mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{b}^\top - \mathbf{T}\|_F^2 + \beta\|\mathbf{W}\|_F^2 \\ &\text{s.t.} \quad T_{i,y_i} - \max_{j \neq y_i} T_{i,j} \geq 1, \quad i = 1, 2, \ldots, n. \end{aligned}}$$

$$\text{Regression:} \quad \min_{\mathbf{W},\mathbf{b}} \|\mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{b}^\top - \mathbf{T}\|_F^2 + \beta\|\mathbf{W}\|_F^2$$

$$\mathbf{W} = (\mathbf{X}^\top\mathbf{H}\mathbf{X} + \beta\mathbf{I}_d)^{-1}\mathbf{X}^\top\mathbf{H}\mathbf{T}, \ \ \mathbf{b} = \frac{(\mathbf{T} - \mathbf{X}\mathbf{W})^\top\mathbf{e}_n}{n}$$

15.10.3 Retargeted LSR

OPT2: Retargeting

$$\boxed{\begin{aligned} &\min_{\mathbf{W},\mathbf{b},\mathbf{T}} \|\mathbf{X}\mathbf{W} + \mathbf{e}_n \mathbf{b}^\top - \mathbf{T}\|_F^2 + \beta \|\mathbf{W}\|_F^2 \\ &\text{s.t.} \quad T_{i,y_i} - \max_{j \neq y_i} T_{i,j} \geq 1, \quad i = 1, 2, \ldots, n. \end{aligned}}$$

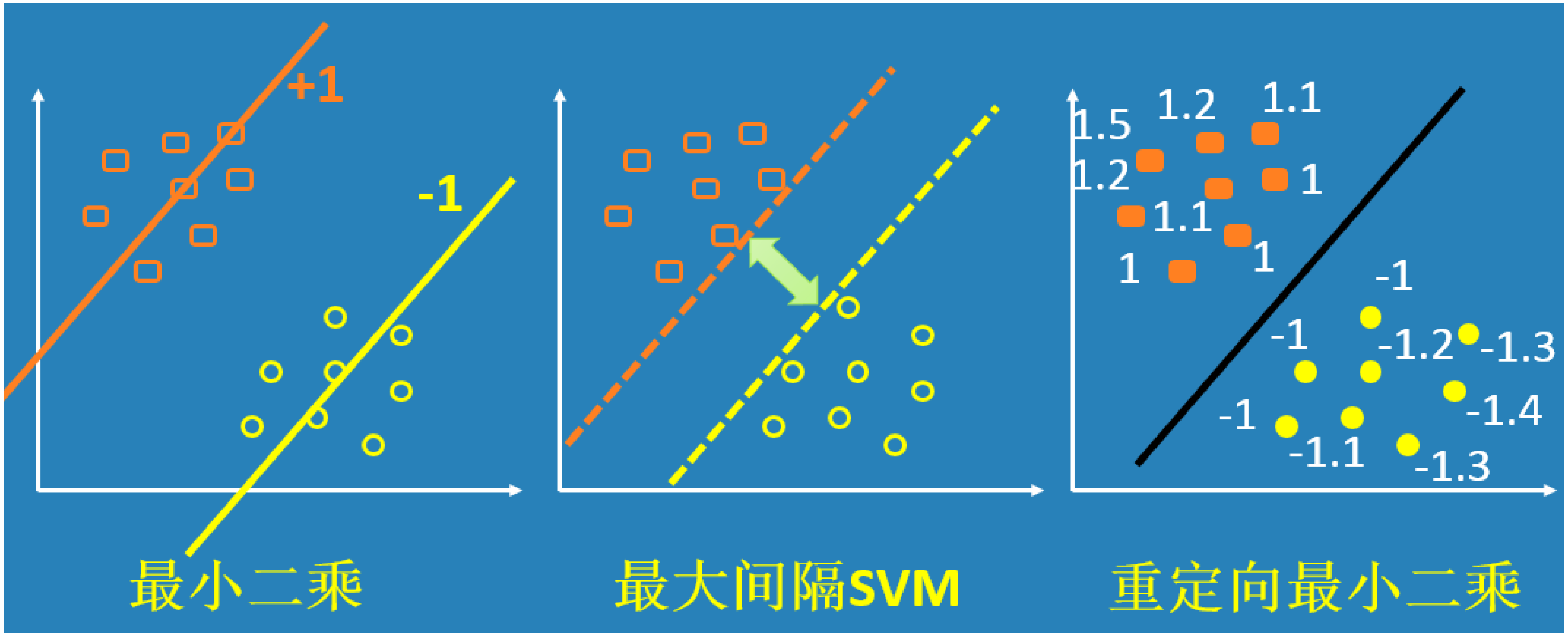
$$\text{Retargeting:} \quad \min_{\mathbf{T}} \|\mathbf{X}\mathbf{W} + \mathbf{e}_n \mathbf{b}^\top - \mathbf{T}\|_F^2 = \|\mathbf{R} - \mathbf{T}\|_F^2$$

$$\text{s.t.} \quad T_{i,y_i} - \max_{j \neq y_i} T_{i,j} \geq 1, \quad i = 1, 2, \ldots, n.$$

$$\min_{\mathbf{t}} \|\mathbf{r} - \mathbf{t}\|_2^2 = \sum_{i=1}^{c} (r_i - t_i)^2 \quad \text{s.t.} \quad t_k - \max_{i \neq k} t_i \geq 1.$$

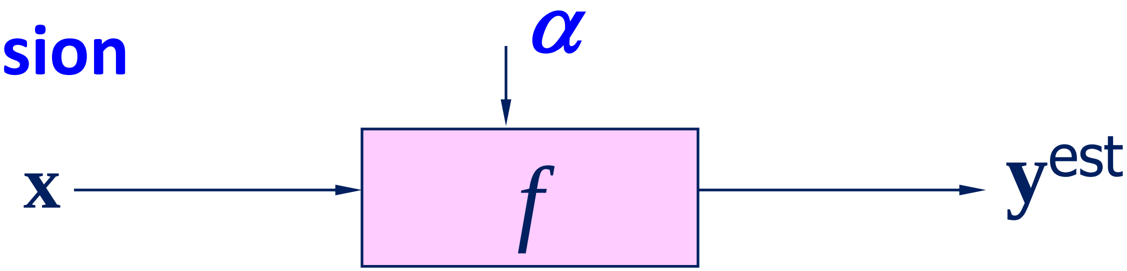
15.10.3 Retargeted LSR

From Regression to Classification

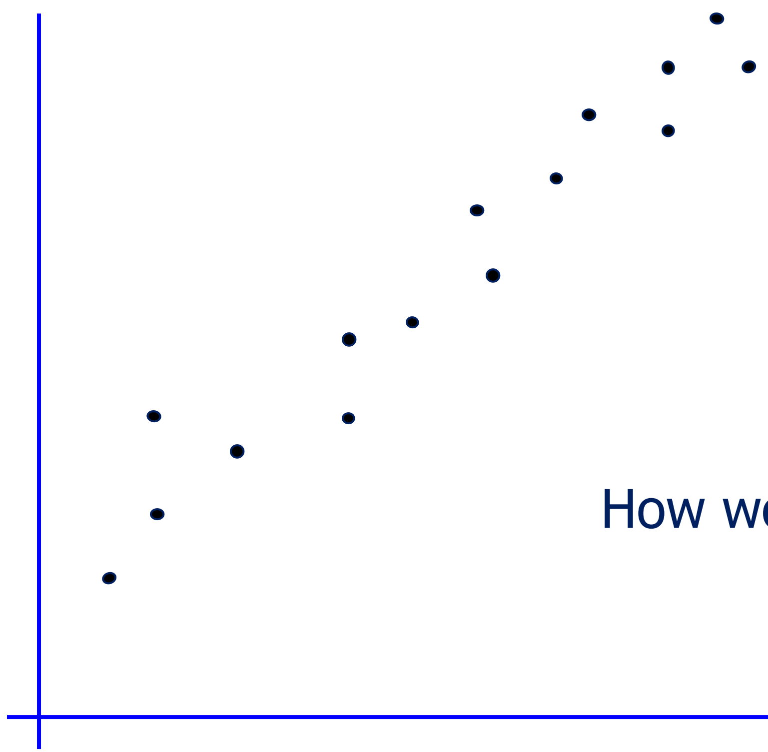


15.11 Support Vector Regression

15.11.1 Linear Regression

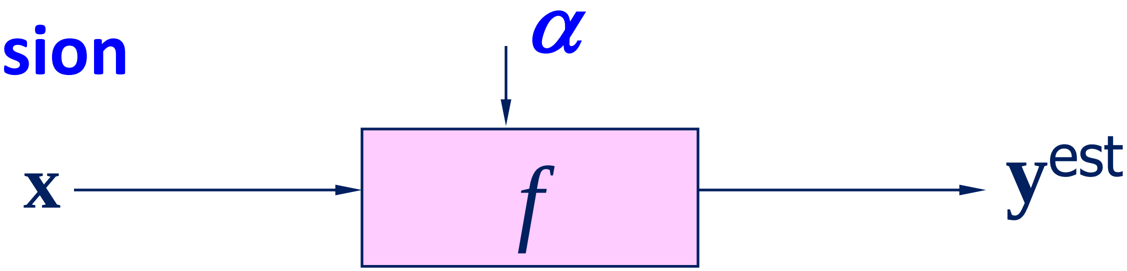


$$f(\mathbf{x},\mathbf{w},\mathbf{b}) = \mathbf{w}\cdot \mathbf{x} + \mathbf{b}$$

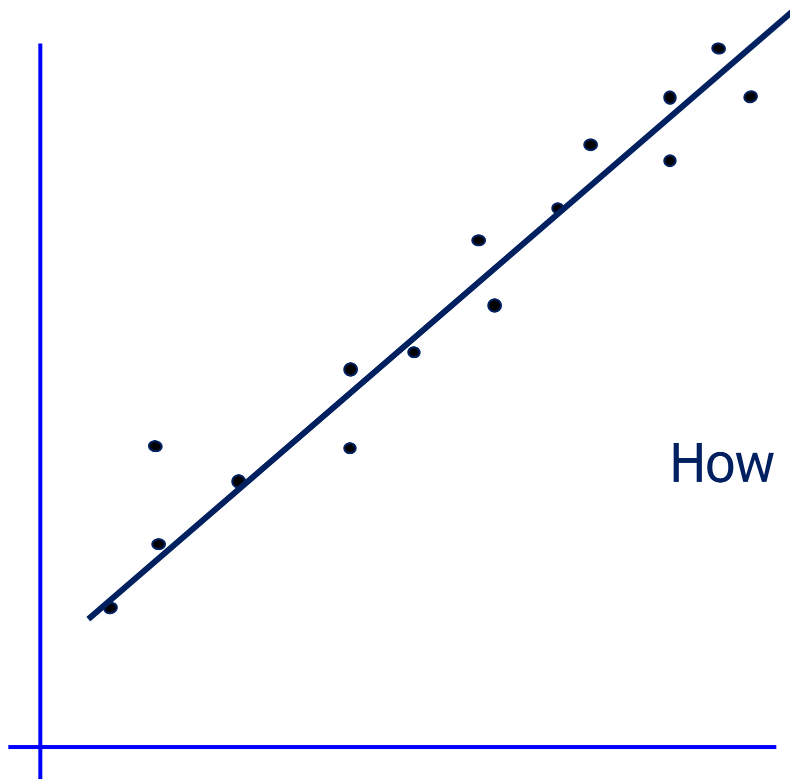


How would you fit this data?

15.11.1 Linear Regression

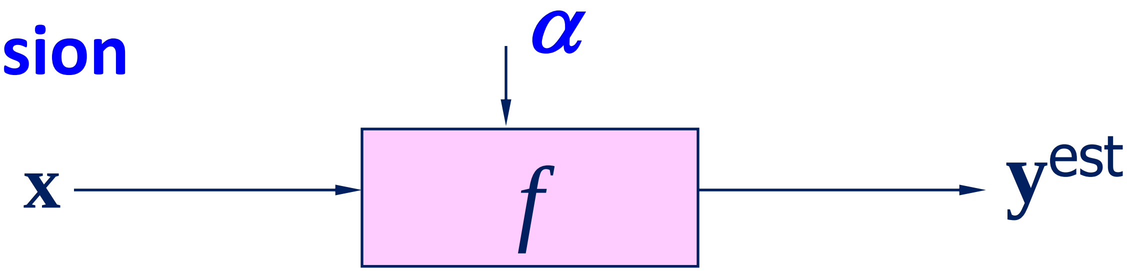


$$f(\mathbf{x},\mathbf{w},\mathbf{b}) = \mathbf{w}\cdot \mathbf{x} + \mathbf{b}$$

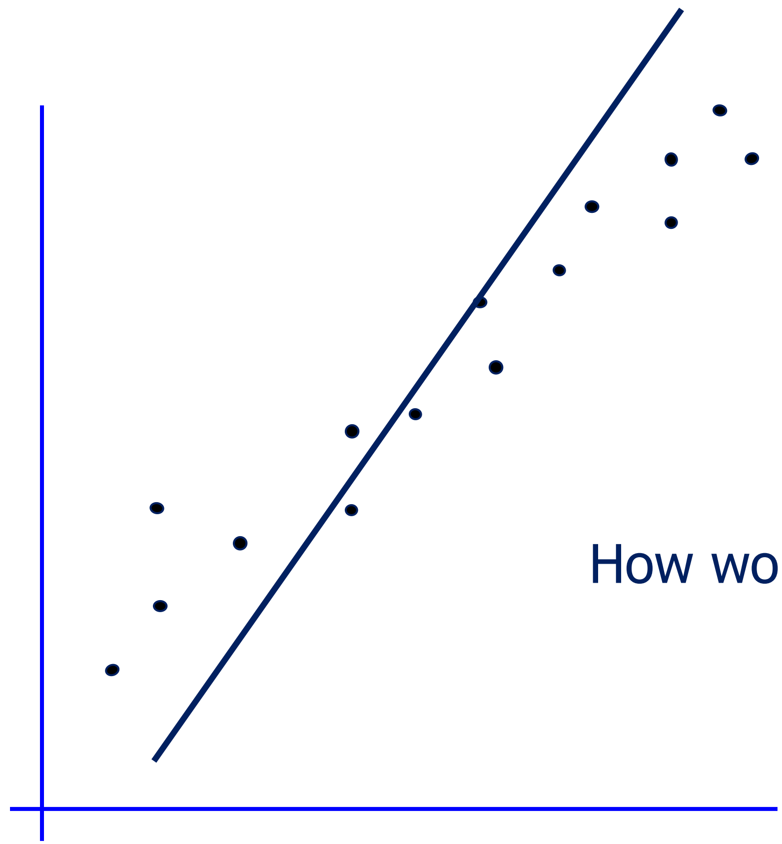


How would you fit this data?

15.11.1 Linear Regression

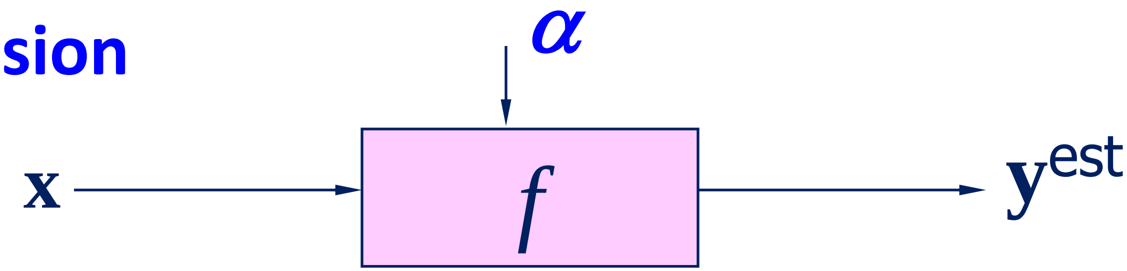


$$f(\mathbf{x},\mathbf{w},\mathbf{b}) = \mathbf{w}\cdot \mathbf{x} + \mathbf{b}$$

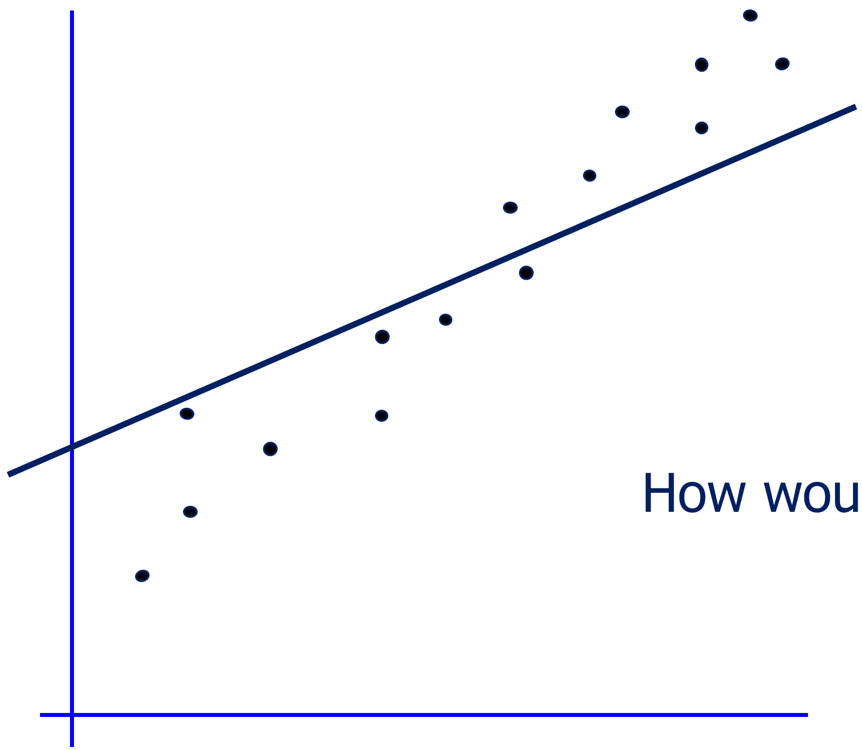


How would you fit this data?

15.11.1 Linear Regression

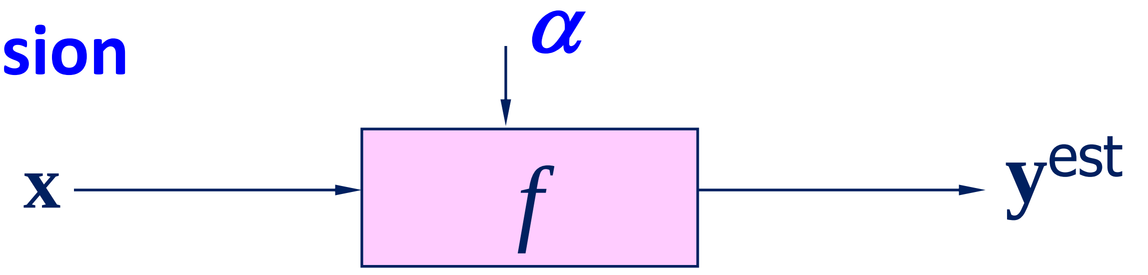


$$f(\mathbf{x},\mathbf{w},\mathbf{b}) = \mathbf{w}\cdot \mathbf{x} + \mathbf{b}$$

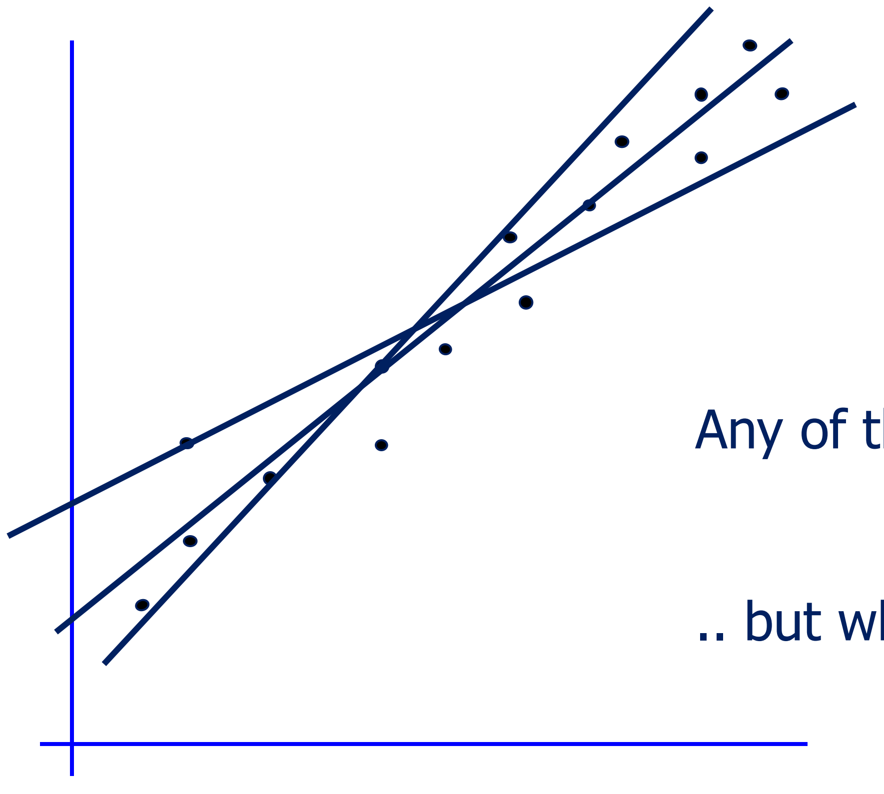


How would you fit this data?

15.11.1 Linear Regression



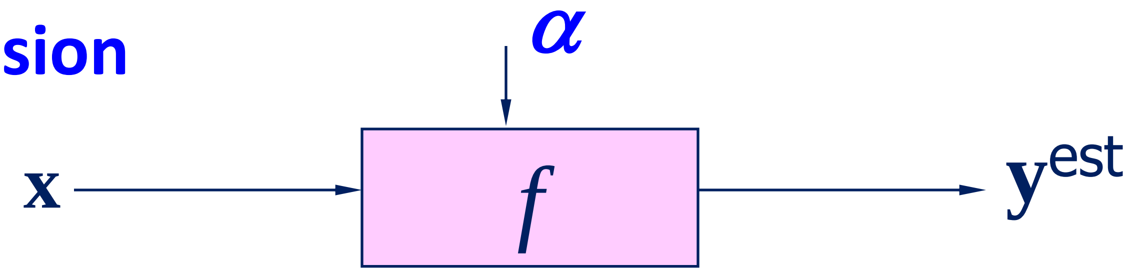
$$f(\mathbf{x},\mathbf{w},\mathbf{b}) = \mathbf{w}\cdot \mathbf{x} + \mathbf{b}$$



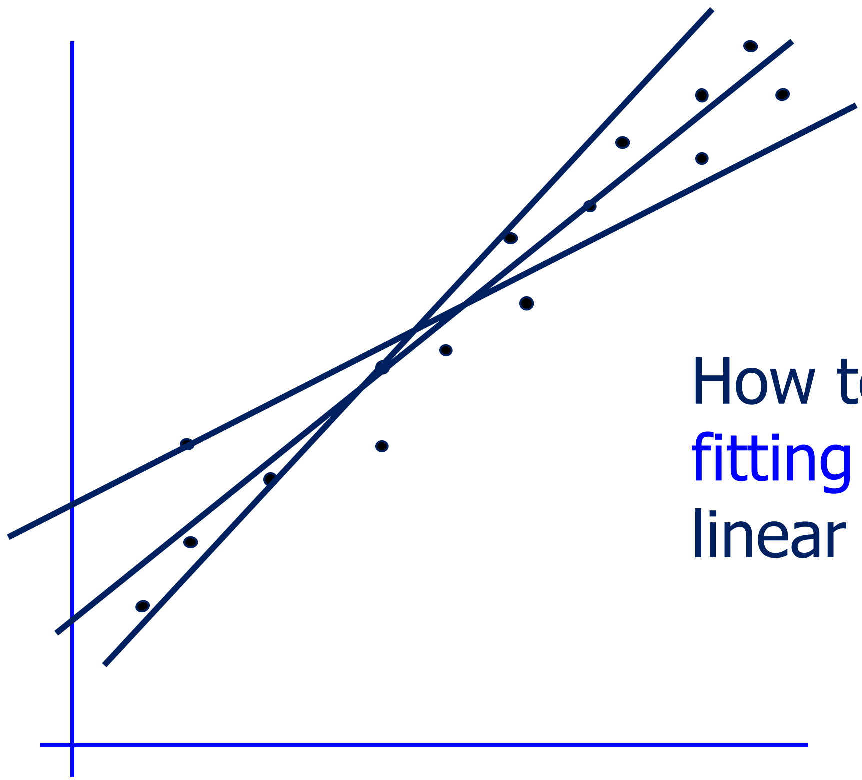
Any of these would be fine..

.. but which is best?

15.11.1 Linear Regression

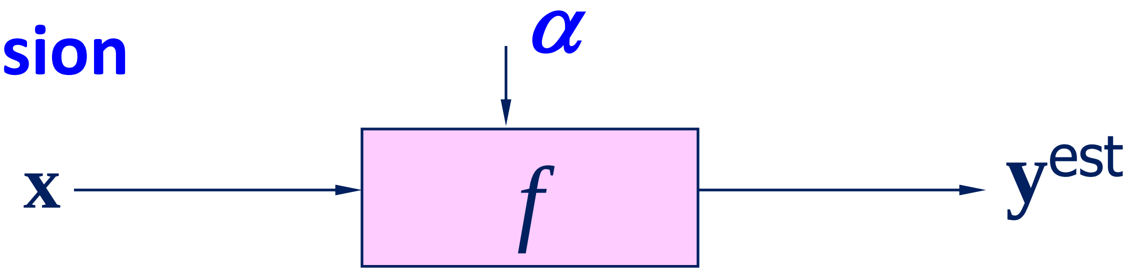


$$f(\mathbf{x},\mathbf{w},\mathbf{b}) = \mathbf{w}\cdot \mathbf{x} + \mathbf{b}$$

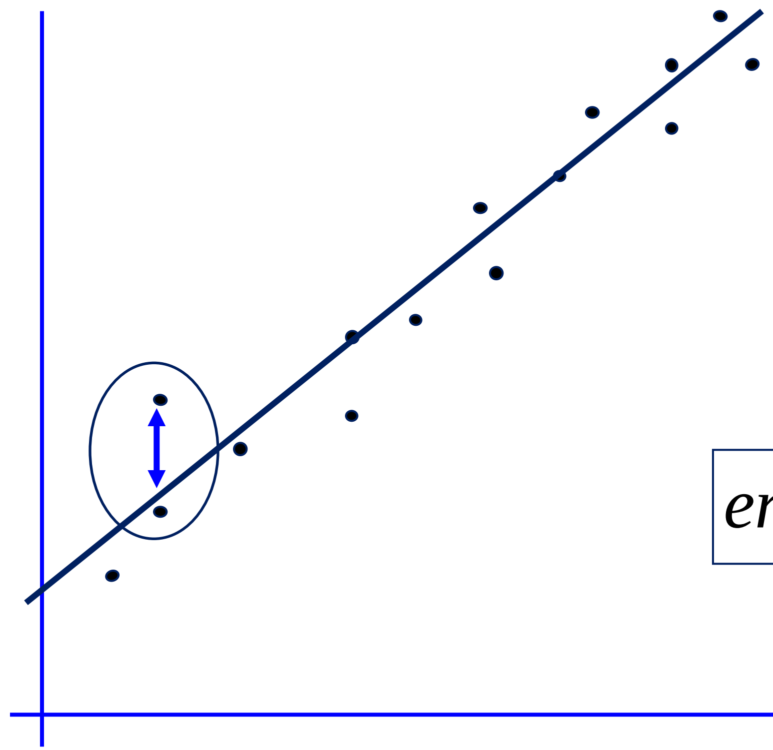


How to define the fitting error of a linear regression ?

15.11.1 Linear Regression



$$f(\mathbf{x},\mathbf{w},\mathbf{b}) = \mathbf{w}\cdot \mathbf{x} + \mathbf{b}$$



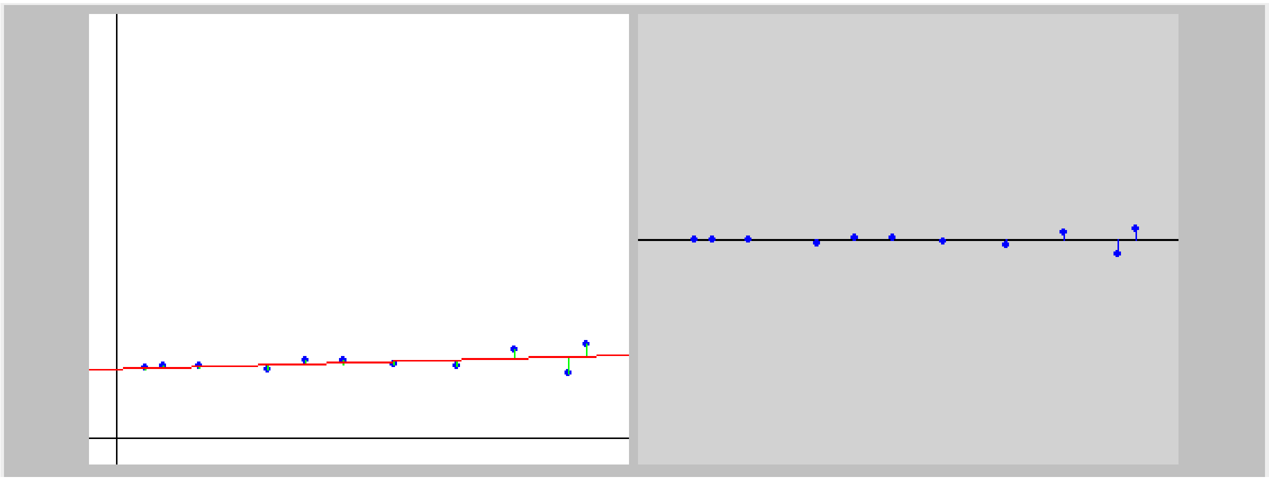
How to define the fitting error of a linear regression ?

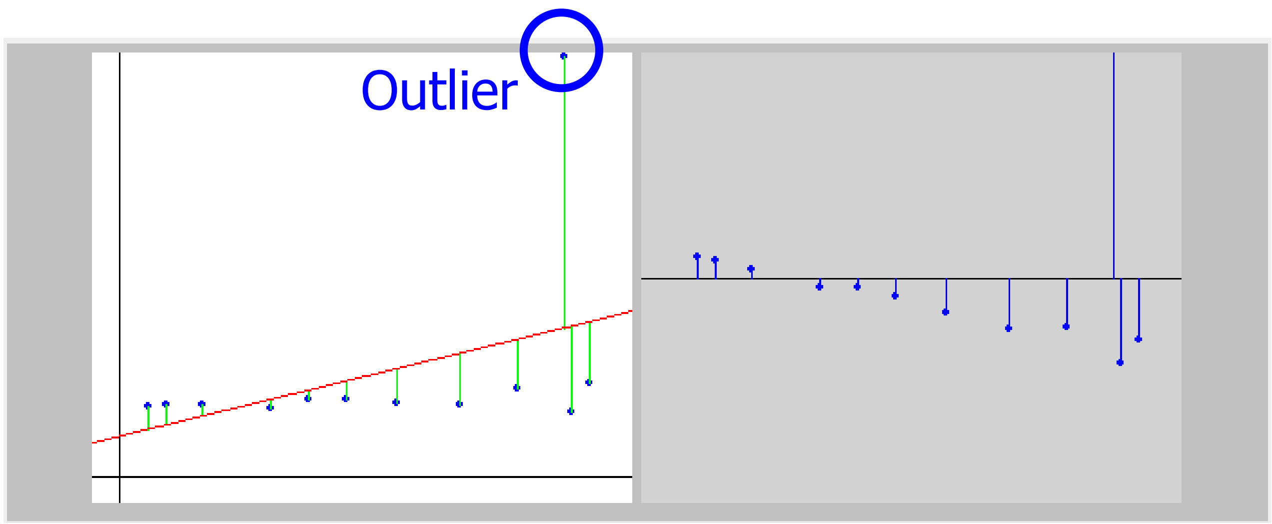
$$\boxed{err_i = (\mathbf{w}\cdot\mathbf{x}_i - b - y_i)^2}$$

Squared-Loss

15.11.1 Linear Regression

Sensitive to Outliers：

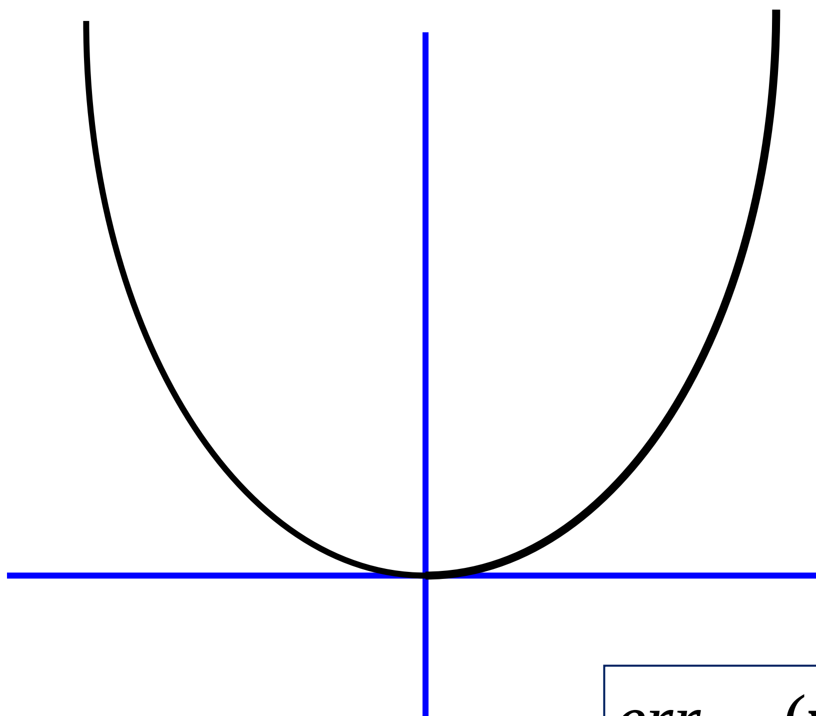




15.11.1 Linear Regression

Why?

- Squared-Loss Function: Fitting Error Grows Quadratically

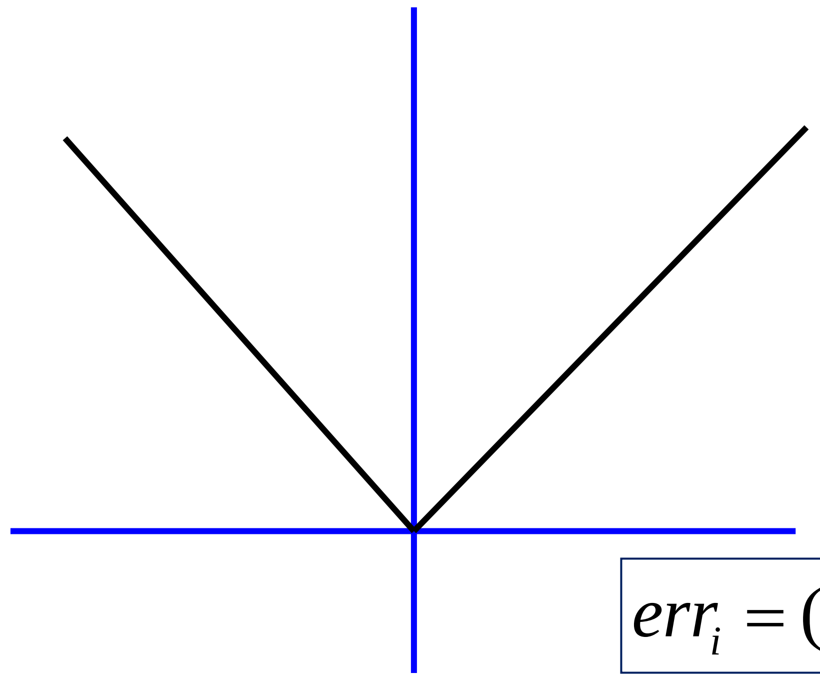


$$err_i = (\mathbf{w} \cdot \mathbf{x}_i - b - y_i)^2$$

15.11.1 Linear Regression

How about Linear-Loss ?

- Linear-Loss Function: Fitting Error Grows Linearly



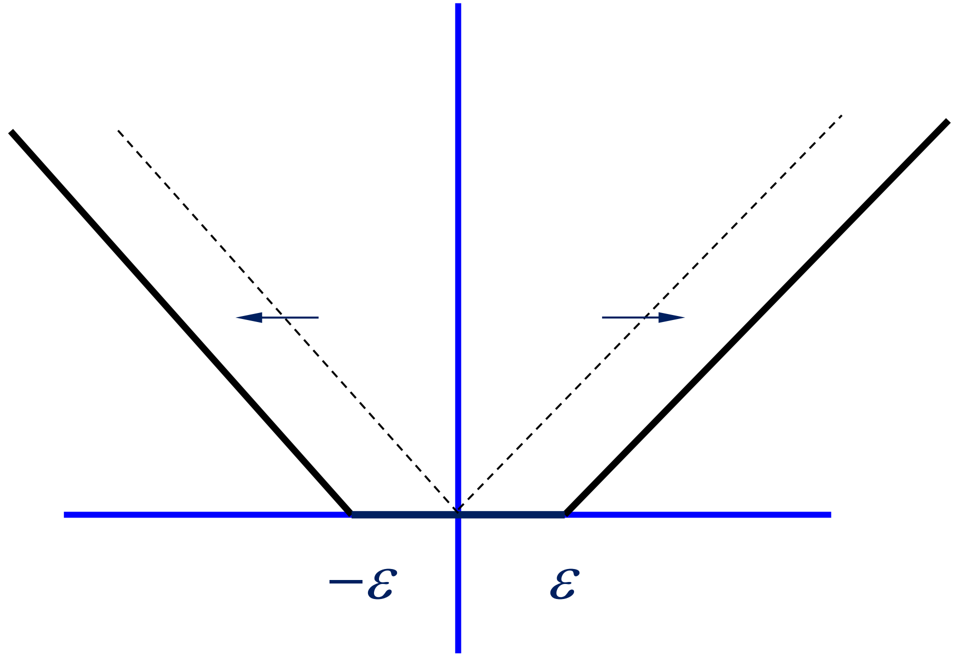
$$err_i = (\mathbf{w} \cdot \mathbf{x}_i - b - y_i)^2$$

15.11.2 支持向量机回归

Actually

- Support vector regression（ SVR） uses the Loss Function below

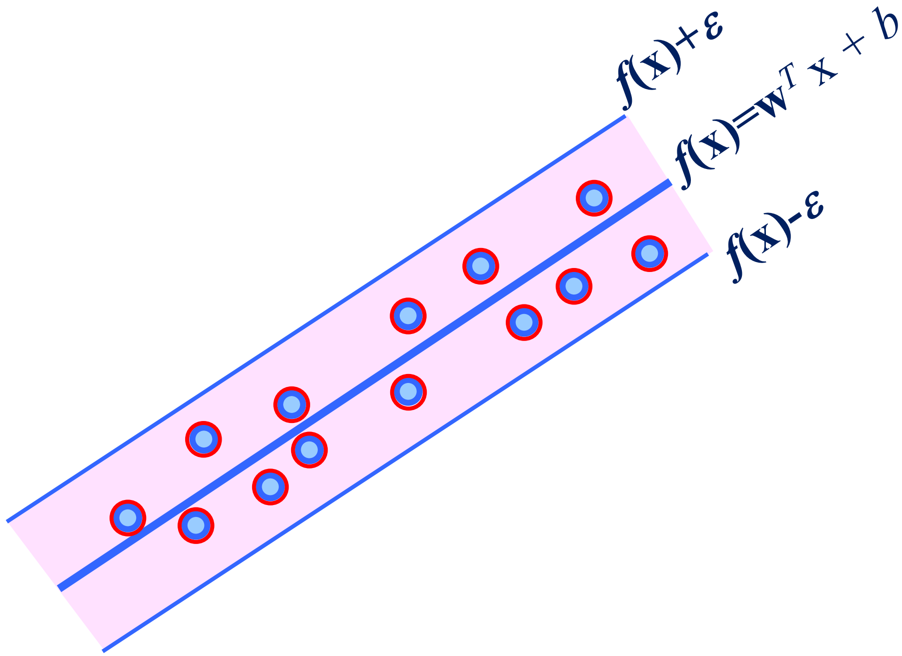
ε-insensitive loss function



回归差异比较小时，比如小于ε时，直接令损失等于0

15.11.2 支持向量机回归

- 基本思想
  - 假设*f*(**x**)与*y*之间可以有 ε 容忍偏差，在这个偏差内，我们都能接收。



如果所有的样本点都在一个宽度为2ε的管道内，我们得到了一个很好的回归！

15.11.2 支持向量机回归

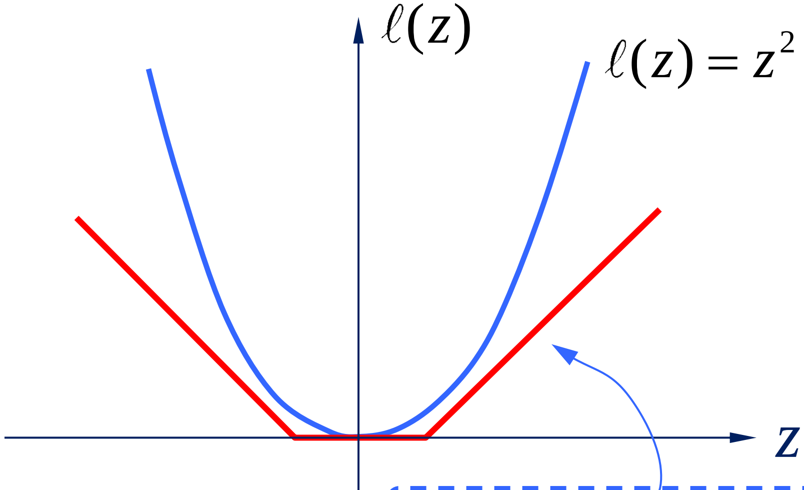
- **学习模型**

ε-insensitive loss function

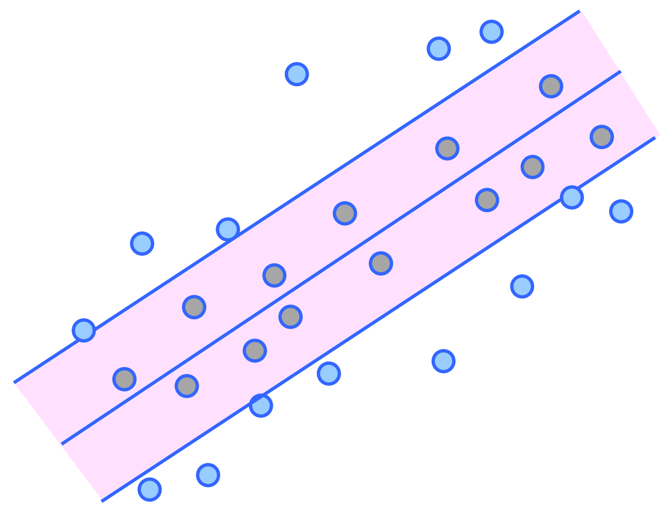
ε-不敏感损失函数

$$\min_{\mathbf{w},\mathbf{b}} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n} \ell_{\varepsilon}\left(f(\mathbf{x}_i) - y_i\right),$$

$$\ell_{\varepsilon}(z) = \begin{cases} 0, & \text{if } |z| \le \varepsilon \\ |z| - \varepsilon, & \text{otherwise} \end{cases}$$

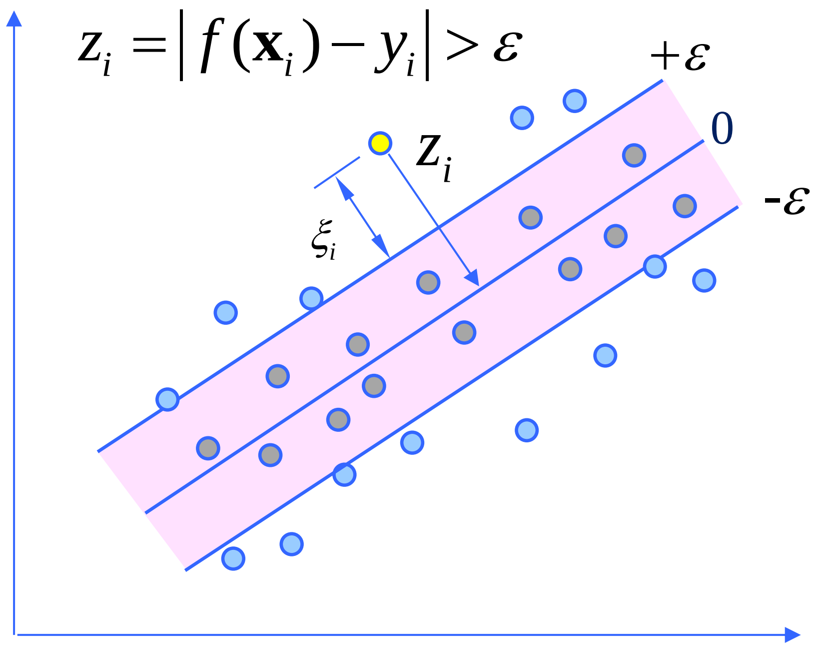


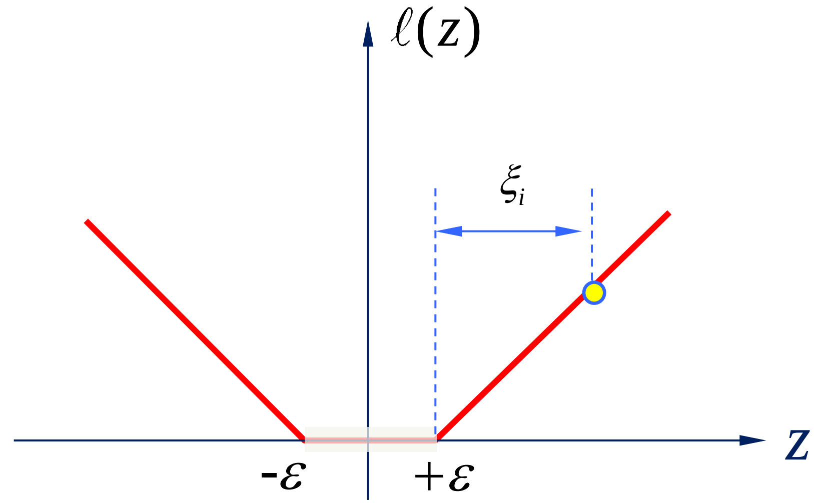
$$\ell_{\varepsilon}(z) = \begin{cases} 0, & \text{if } |z| \le \varepsilon \\ |z| - \varepsilon, & \text{otherwise} \end{cases}$$



15.11.2 支持向量机回归

- **学习模型** $\min_{\mathbf{w},\mathbf{b}} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\ell_\varepsilon\left(f(\mathbf{x}_i)-y_i\right),$





$$\ell_\varepsilon(z) = \begin{cases} 0, & \text{if } |z| \le \varepsilon \\ |z| - \varepsilon, & \text{otherwise} \end{cases}$$

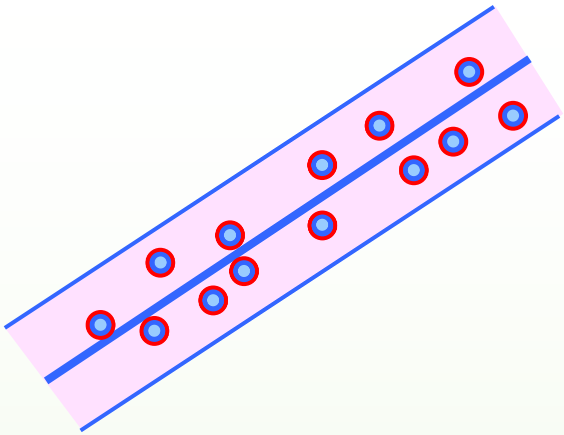
ε-不敏感损失函数

对应的损失为：$|f(\mathbf{x}_i) - y_i| - \varepsilon$

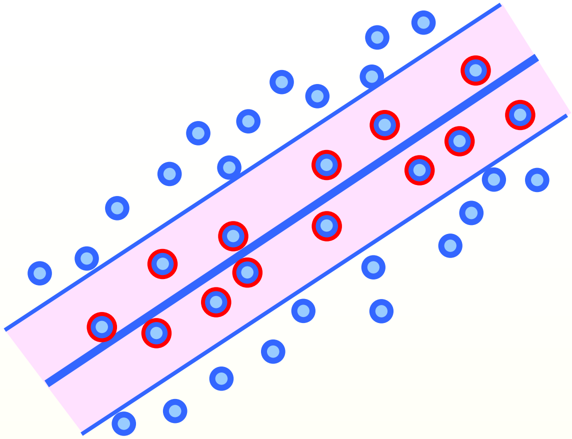
15.11.2 支持向量机回归

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2$$
$$s.t. \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \geq 0,$$
$$i = 1, 2, \cdots, n$$

- 学习模型（从支持向量机的角度）



$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|_2^2$$
$$s.t. \quad |f(\mathbf{x}_i) - y_i| \leq \varepsilon,$$
$$i = 1, 2, \ldots n$$



$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i$$
$$s.t. \quad |f(\mathbf{x}_i) - y_i| - \xi_i \leq \varepsilon,$$
$$\xi_i \geq 0,$$
$$i = 1, 2, \ldots n$$

15.11.2 支持向量机回归

学习模型：

$$
\begin{aligned}
\min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 \\
s.t. \quad & |f(\mathbf{x}_i) - y_i| - \xi_i \le \varepsilon, \\
& \xi_i \ge 0, \\
& i = 1, 2, \dots n
\end{aligned}
$$

$\Rightarrow$

$$
\begin{aligned}
\min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i \\
s.t. \quad & -\varepsilon - \xi_i \le f(\mathbf{x}_i) - y_i \le \varepsilon + \xi_i, \\
& \xi_i \ge 0, \\
& i = 1, 2, \dots n
\end{aligned}
$$

15.11.2 支持向量机回归

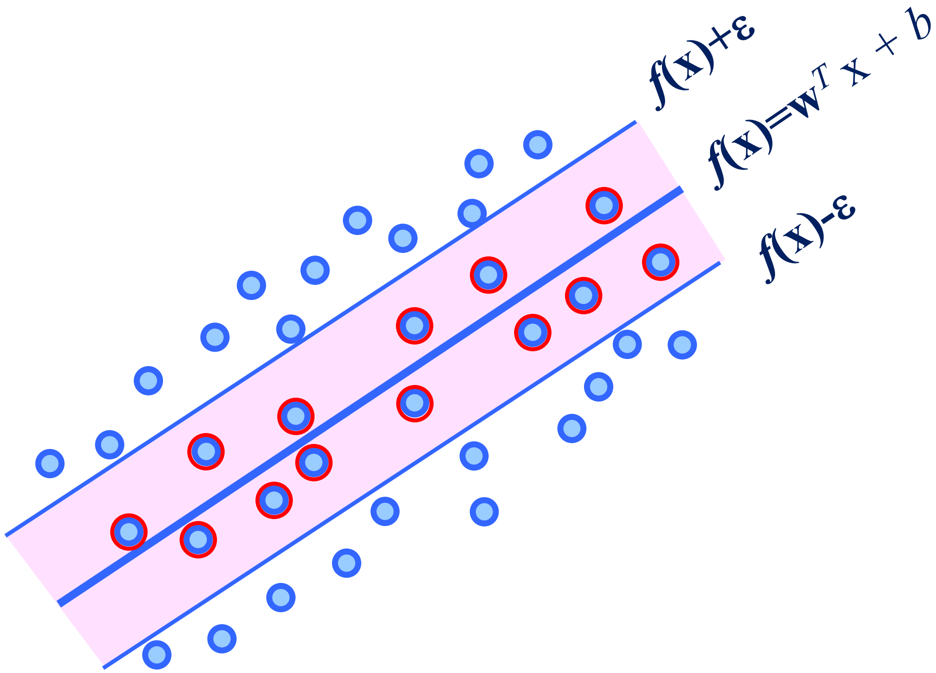
学习模型（松弛模型）

$$
\begin{aligned}
\min_{\mathbf{w},b,\xi_i,\hat{\xi}_i} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\left(\xi_i + \hat{\xi}_i\right) \\
s.t. \quad & f(\mathbf{x}_i) - y_i \le \varepsilon + \xi_i, \\
& y_i - f(\mathbf{x}_i) \le \varepsilon + \hat{\xi}_i, \\
& \xi_i \ge 0, \\
& \hat{\xi}_i \ge 0, \\
& i = 1, 2, \cdots, n
\end{aligned}
$$

Similar to SVM, this can be solved as a quadratic programming problem

15.11.2 支持向量机回归

- Support vector regression, SVR: 假设$f(\mathbf{x})$与y之间可以有 ε 容忍偏差。

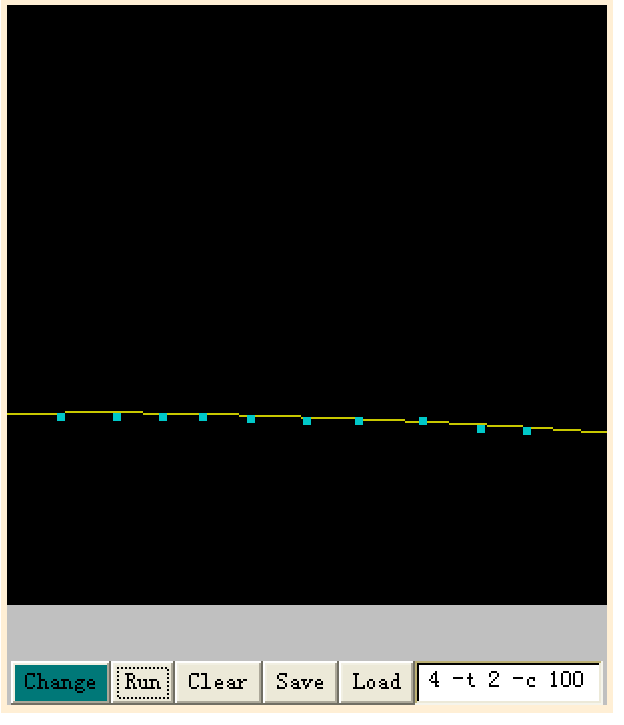


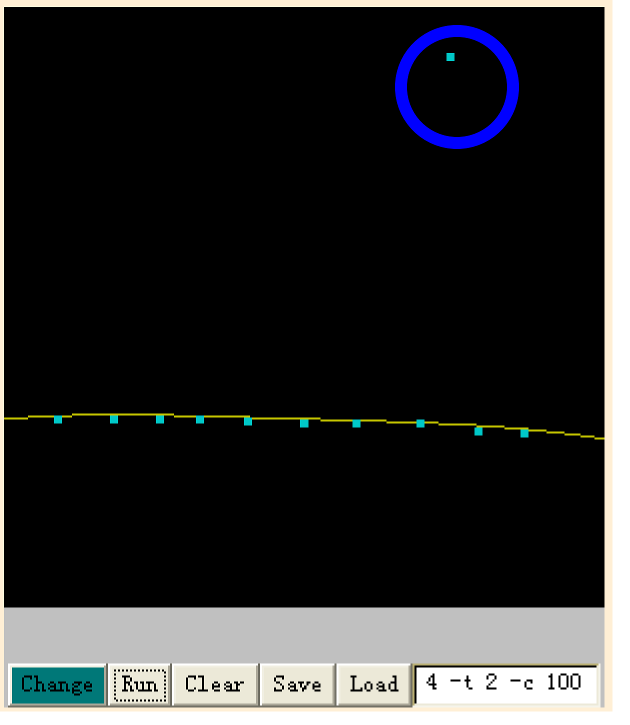
在所有样本点中，只有分布在“管壁”和“管壁”之外的样本点决定管道的位置。这一部分训练样本称为“支持向量”。

15.11.2 支持向量机回归

Demo

- Less Sensitive to Outlier

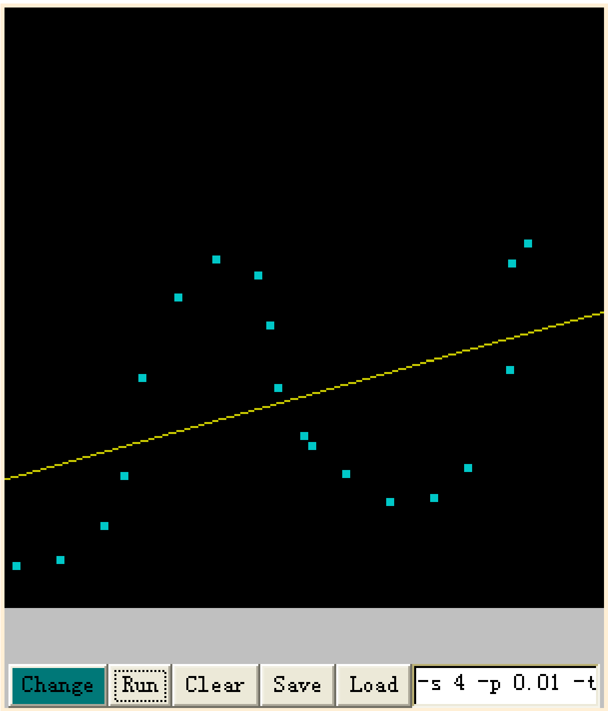


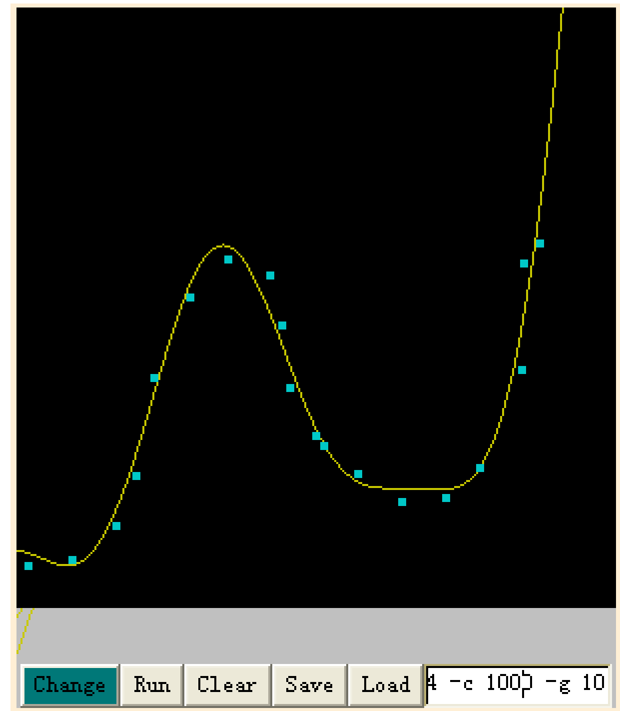


15.11.2 支持向量机回归

Again, Extend to Non-Linear Case

- Similar with SVM





15.12 SVM for Ranking

15.12 支持向量机排序

- **Learning to rank (L2R)**
  - 排序一直是信息检索的核心问题之一，Learning to Rank(简称LTR)用机器学习的思想来解决排序问题。
  - L2R有三种主要的方法：PointWise，PairWise，ListWise。
  - Ranking SVM算法是PointWise方法的一种，由R. Herbrich等人在2000提出。
  - RankSVM的基本思想是，将排序问题转化为pairwise的分类问题，然后使用SVM分类模型进行学习并求解。

15.12 支持向量机排序

- **将排序问题转化为分类问题**
  - 比如，以文档查询为背景“query-doc pair”。
  - 记一个文档的特征为$\mathbf{x}$，我们的目的是需要找到一个排序函数$f(\mathbf{x})$，根据$f(\mathbf{x})$的大小来决定排序顺序。即如果$f(\mathbf{x}_i) > f(\mathbf{x}_j)$，则 $\mathbf{x}_i$ 应该排在 $\mathbf{x}_j$ 的前面，反之亦然：

$$\mathbf{x}_i \succ \mathbf{x}_j \quad \Leftrightarrow \quad f(\mathbf{x}_i) > f(\mathbf{x}_j)$$

  - 理论上，$f(\mathbf{x})$可以是任意函数。
  - 为了简单起见，假设其为线性函数：$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$
  - 由于排序不受参数 b 的影响，所以可以令：$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$

15.12 支持向量机排序

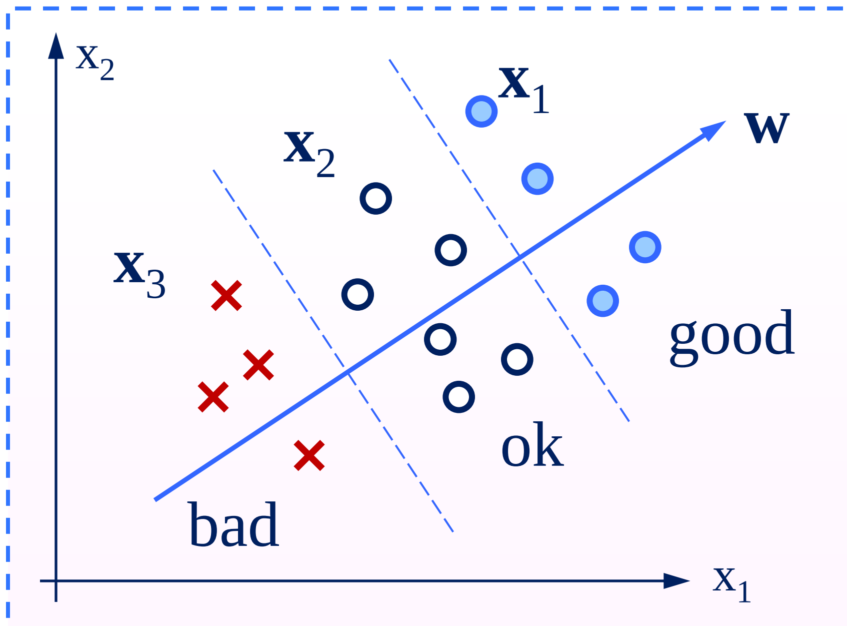
- **为什么可以转换为两类分类问题？**
  - 首先，对于任意两个数据点 $\mathbf{x}_i$ 和 $\mathbf{x}_j$，若 $f(\mathbf{x})$是线性函数，则如下关系成立：

$$f(\mathbf{x}_i) > f(\mathbf{x}_j) \quad \Leftrightarrow \quad \mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j) > 0$$

  - 然后，可以对$\mathbf{x}_i$和$\mathbf{x}_j$的差值向量引入两类分类问题，按如下方式进行标签赋值：

$$y = \begin{cases} +1, & \text{if } x_i \succ x_j \\ -1, & \text{if } x_i \prec x_j \end{cases}, \qquad \mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j) > 0 \quad \Leftrightarrow \quad y = +1$$

- **SVM模型解决排序问题**
  - 将排序问题转化为分类问题之后，可使用Linear SVM或 kernel SVM解决排序问题。



上图展示了一组查询，给出了所召回的文档，其中文档的相关程度等级分为三档(good，ok，bad)。权重向量w对应排序函数，可以对“查询-返回”对进行打分和排序。

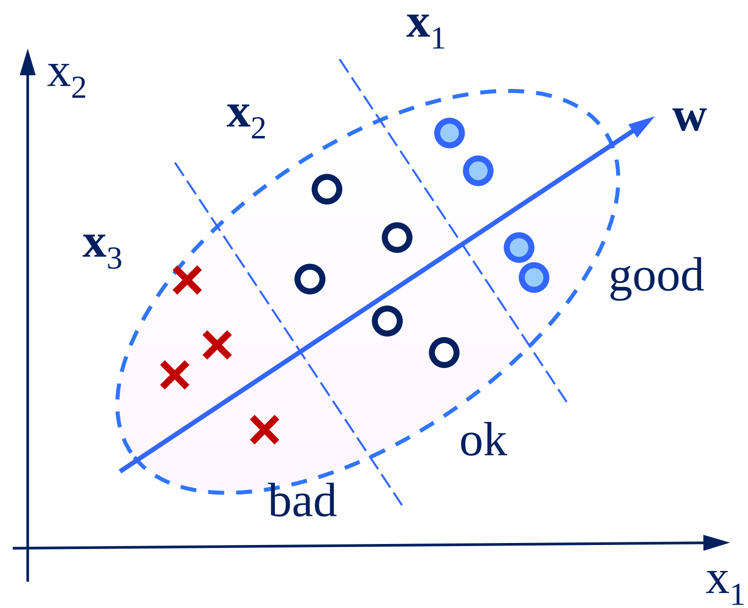
15.12 支持向量机排序

为SVM准备样本：

- ✓ 给定一个查询及其反馈，可对样本进行组合，形成新数据点：$\mathbf{x}_1-\mathbf{x}_2$，$\mathbf{x}_1-\mathbf{x}_3$，$\mathbf{x}_2-\mathbf{x}_3$。其label也会被重新赋值，比如将$\mathbf{x}_1-\mathbf{x}_2$，$\mathbf{x}_1-\mathbf{x}_3$，$\mathbf{x}_2-\mathbf{x}_3$的label赋值为正类。
- ✓ 为了构造分类问题，还需负样本。可以使用其反方向向量作为负样本：$\mathbf{x}_2-\mathbf{x}_1$，$\mathbf{x}_3-\mathbf{x}_1$，$\mathbf{x}_3-\mathbf{x}_2$。
- ✓ 需要指出的是，在组合形成新样本时，不能使用在原始排序问题中处于相同相似度等级的两个数据点，也不能使用处于不同query下的两个数据点来组合新样本。

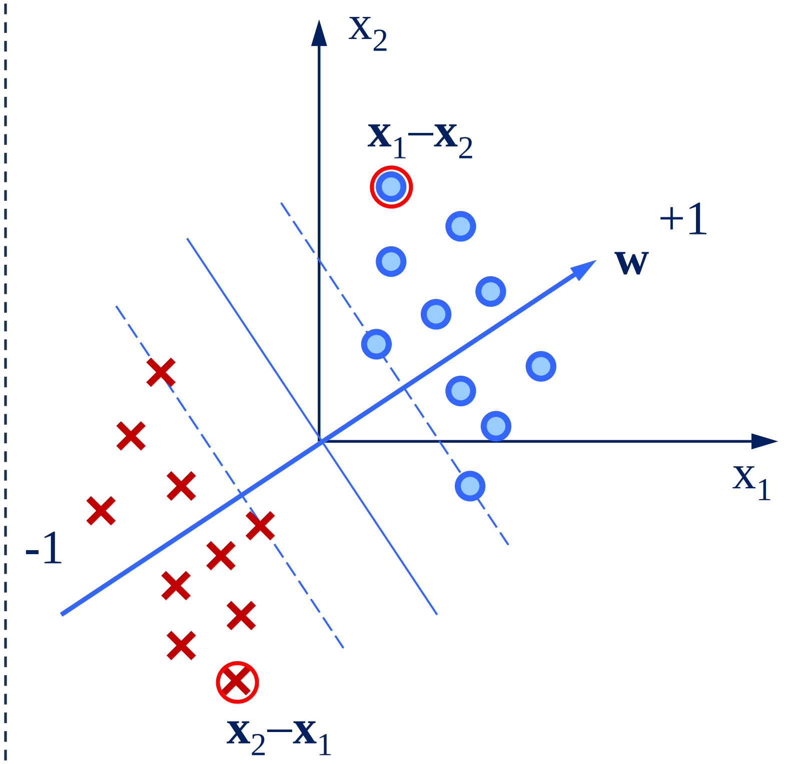
15.12 支持向量机排序

为SVM准备样本：



正样本： $\mathbf{x}_1-\mathbf{x}_2$， $\mathbf{x}_1-\mathbf{x}_3$， $\mathbf{x}_2-\mathbf{x}_3$

负样本： $\mathbf{x}_2-\mathbf{x}_1$， $\mathbf{x}_3-\mathbf{x}_1$， $\mathbf{x}_3-\mathbf{x}_2$



15.12 支持向量机排序

- **学习模型**
  - 转化为分类问题后，便可以采用SVM的通用方式进行求解。学习模型如下：

$$
\begin{aligned}
\min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\
s.t. \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \\
& \xi_i \geq 0, \\
& i = 1, 2, \cdots, n
\end{aligned}
$$

支持向量机分类

$$
\begin{aligned}
\min_{\mathbf{w},\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\
s.t. \quad & y_i\left(\mathbf{w}^T\left(\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\right)\right) \geq 1 - \xi_i, \\
& \xi_i \geq 0, \\
& i = 1, 2, \cdots, n
\end{aligned}
$$

支持向量机排序

15.12 支持向量机排序

- **学习模型**
  - 类似地，采用合页损失函数：

$$\min_{\mathbf{w},b} \sum_{i=1}^{n} \left[1 - y_i \left(\mathbf{w}^T \left(\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)} \right) \right) \right]_+ + \lambda \| \mathbf{w} \|_2^2$$

$$\text{where } [z]_+ = \begin{cases} z, & \text{if } z > 0 \\ 0, & \text{otherwise} \end{cases}$$

15.12 支持向量机排序

- **对偶学习模型**

支持向量机分类

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) - \sum_{i=1}^{n}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{n}\alpha_i y_i = 0;$$

$$0 \le \alpha_i \le C, \quad i = 1, 2, \ldots, n$$

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \left(\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\right)\cdot\left(\mathbf{x}_j^{(1)} - \mathbf{x}_j^{(2)}\right) - \sum_{i=1}^{n}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{n}\alpha_i y_i = 0;$$

$$0 \le \alpha_i \le C, \quad i = 1, 2, \ldots, n$$

支持向量机排序

15.12 支持向量机排序

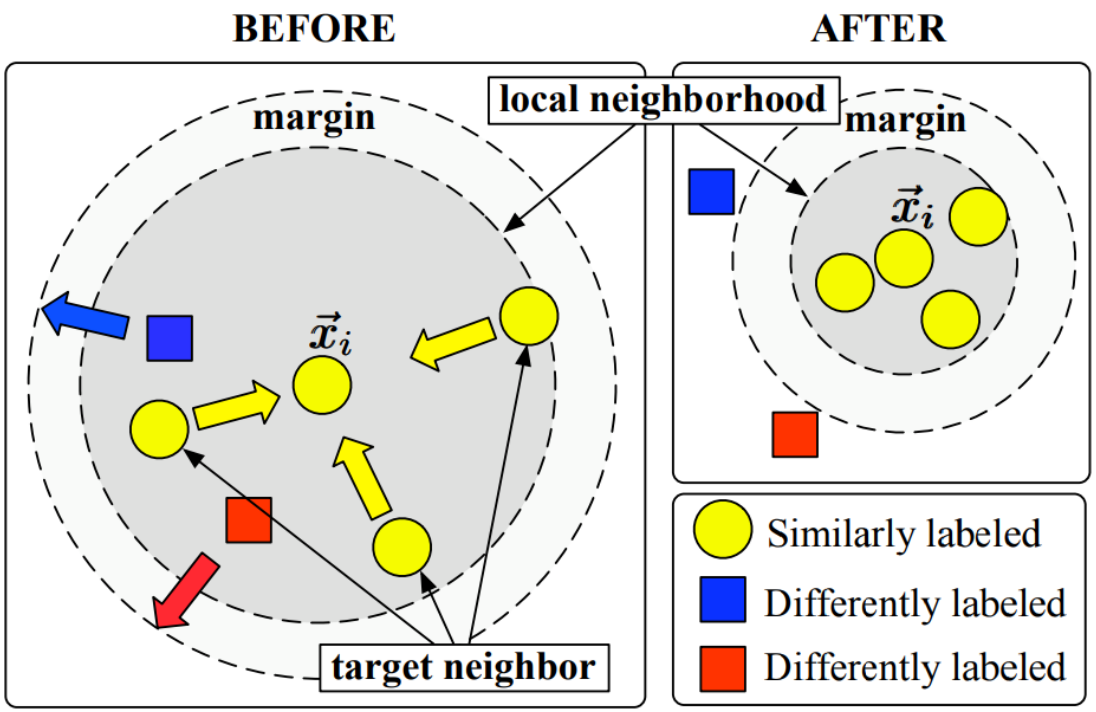
- **其它改进**
  - Hinge Loss 是 0-1 损 失 。 损 失 函 数 的 优 化 目 标 可 以 进 一 步 与 Information Retrieval 的Evaluation常用指标建立紧密联系。
  - **更复杂的方法：采用**Ordinal Regression方法来对此问题进行建模。
  - 还要可以将对偶问题转化为核学习技巧。

15.13 Large Margin Nearest Neighbor (LMNN)

15.13 Large Margin Nearest Neighbor (LMNN)

- Learn a Mahanalobis distance metric in the kNN classification setting by semi-definite programming (SDP), that
  - Enforce the k-nearest neighbors within the same class more closely
  - Examples from different classes are separated by a large margin

(Weinberger et al., 2006)



After training, we hope:

(1) k=3 target neighbors lie within a smaller radius

(2) differently labeled inputs lie outside this smaller radius with a margin of at least one unit distance

学习模型

属于同一类的邻近点之间的距离要小

同一邻域内，不属于同一类的点对

$$\min \sum_{ij} \boxed{\eta_{ij}(\mathbf{x}_i-\mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i-\mathbf{x}_j)} + \sum_{ijl} \boxed{\underline{\eta_{ij}}(1-\underline{y_{il}})}\xi_{ijl}$$

$$s.t.\ \underline{(\mathbf{x}_i-\mathbf{x}_l)^T\mathbf{M}(\mathbf{x}_i-\mathbf{x}_l)-(\mathbf{x}_i-\mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i-\mathbf{x}_j)} \geq 1-\boxed{\xi_{ijl}}$$

$$\xi_{ijl} \geq 0$$

同一邻域内，不属于同一类的点之间的距离要大于“1”

$$\mathbf{M} \succ = 0$$

relax variables

$$\eta_{ij} = \begin{cases} 1, & \text{if } (\mathbf{x}_i,\mathbf{x}_j) \in \text{the same class, and } \mathbf{x}_j \in N_k(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases}$$

$$y_{il} = \begin{cases} 1, & \text{if } (\mathbf{x}_i,\mathbf{x}_l) \in \text{the same class} \\ 0, & \text{otherwise} \end{cases}$$

Kilian Q. Weinberger, John Blitzer and Lawrence K. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification, NIPS, 2005.

15.14 Kernel Learning (Extension)

15.14.1 核主成分分析：KPCA

- **PCA**
  - 给定 $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n]\in R^{n\times m}$，**并假定均值为零**
  - 计算协方差矩阵：$\mathbf{C}=\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T$
  - 对 C 施行矩阵特征值分解：$\mathbf{C}=\mathbf{U}\Sigma\mathbf{U}^T$
  - 取**指定个数的**最大特征值对应的特征向量子集，组成投影向量 $\mathbf{W}=\mathbf{U}_S$，（$\mathbf{U}_S$为U的子矩阵）
  - 对新样本 x, 将其投影至低维子空间：$\mathbf{y}=\mathbf{W}^T\mathbf{x}$

- **核主成分分析：KPCA**
  - 引入非线性映射： $\phi: \mathrm{R}^m \rightarrow F$
  - 将数据进行映射： $\{\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \ldots, \phi(\mathbf{x}_n)\} \subset F$
  - 计算协方差矩阵： $\bar{\mathbf{C}} = \frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T$
  - 作特征值分解： $\bar{\mathbf{C}} = \bar{\mathbf{U}}\,\bar{\mathbf{\Sigma}}\,\bar{\mathbf{U}}^T$
  - **What are the difficulties?（如何实现）**
    - B. Scholkopf, A. J. Smola, K. R. Muller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation,10(5):1299–1319, 1998.

- **主要结论：与PCA作对比**
  - 在PCA中，**W**为样本协方差矩阵的前d个特征值对应的特征向量所构成：$\mathbf{W}=[\mathbf{w}_1,\mathbf{w}_2,\cdots,\mathbf{w}_d]\in R^{m\times d}$ ，
    - 样本点**x**的投影： $\mathbf{y}=\mathbf{W}^T\mathbf{x}=[\mathbf{w}_1\cdot\mathbf{x},\mathbf{w}_2\cdot\mathbf{x},\cdots,\mathbf{w}_d\cdot\mathbf{x}]^T\in R^d$
  - 在KPCA中，**W**为样本在高维特征空间中的协方差矩阵的前d个特征值对应的特征向量所构成：

$$\mathbf{W}=\left[\mathbf{v}_1=\sum_{i=1}^{n}\alpha_i^1\phi(\mathbf{x}_i),\ \mathbf{v}_2=\sum_{i=1}^{n}\alpha_i^2\phi(\mathbf{x}_i),\quad \dots,\quad \mathbf{v}_d=\sum_{i=1}^{n}\alpha_i^d\phi(\mathbf{x}_i)\right]$$

    - 样本点**x**的投影：

$$\mathbf{y}=\left[\mathbf{v}_1^T\phi(\mathbf{x}),\quad \mathbf{v}_2^T\phi(\mathbf{x}),\quad \cdots,\quad \mathbf{v}_d^T\phi(\mathbf{x})\right]^T$$

$$=\left[\sum_{i=1}^{n}\alpha_i^1K(\mathbf{x}_i,\mathbf{x}),\sum_{i=1}^{n}\alpha_i^2K(\mathbf{x}_i,\mathbf{x}),\cdots,\sum_{i=1}^{n}\alpha_i^dK(\mathbf{x}_i,\mathbf{x})\right]^T$$

由数据点构成的核矩阵的第一个（上标）特征向量第i个（下标）分量

15.14.2 关于核化的一般性理论

- **主要参考文献**
  - Changshui Zhang, Feiping Nie, Shiming Xiang: ***A general kernelization framework for learning algorithms based on kernel PCA***. Neurocomputing 73(4-6): 959-967, 2010

15.14.2 关于核化的一般性理论

- **满秩PCA**
  - 对于训练数据**X**，设其中心化的内积矩阵（即协方差矩阵）**C**的秩为 r，如果提取PCA的前 r 个主成分，则称此过程为满秩PCA。
- **满秩KPCA**
  - 对于训练数据**X**，设其中心化的核矩阵**K**的秩为r，如果提取KPCA的前 r 个主成分，则称此过程为满秩KPCA。

15.14.2 关于核化的一般性理论

- **定理：**

如果一个线性算法同时满足如下两个条件：

(1) 算法的**输出仅与内积运算** $\mathbf{x} \cdot \mathbf{x}_i$, $(i = 1, 2, \ldots, n)$ 有关；

(2) 对训练数据的**平移不会改变算法的输出结果**；

则该算法的核化可以通过对数据先做满秩KPCA变换，然后在变换后的数据上直接再做该线算法来实现。

15.14.2 关于核化的一般性理论

- **举例**
  - **KSVM = KPCA + SVM**
  - **KLDA = KPCA + LDA**
  - **KCCA = KPCA + CCA**
  - **KPLS = KPCA + PLS**
  - **核岭回归 = KPCA + 岭回归**

在实际应用中，对有噪声的数据，采用低秩PCA来做会更好！

15.15 Conclusion

- Structural Risk Minimization → VC Dimension
- Large Margin → Low VC Dimension
- Hard Margin SVM → Soft Margin SVM
- Dual Problem → Kernel Method
- Model Selection → Tradeoff C and Kernel
- Optimization → SMO
- Multi-Class SVM → Binary or Single-Machine
- Least Squares Extensions → Another Perspective
- PCA → KPCA
- Large Margin for Regression, Ranking, Nearest Neighbors
- 关于核学习的一般性结论
- Ideas of: large margin, hinge loss, kernel method, are widely used in different PR & ML tasks

Thank All of You!

(Questions?)

向世明

smxiang@nlpr.ia.ac.cn

people.ucas.ac.cn/~xiangshiming

时空数据分析与学习课题组(STDAL)

中科院自动化研究所· 模式识别国家重点实验室