Supervised Learning: Support Vector Machines

吉建民

USTC
jianmin@ustc.edu.cn

2024 年 5 月 7 日

Used Materials

Disclaimer: 本课件采用了 S. Russell and P. Norvig's Artificial Intelligence –A modern approach slides, 徐林莉老师课件和其他网络课程课件，也采用了 GitHub 中开源代码，以及部分网络博客内容

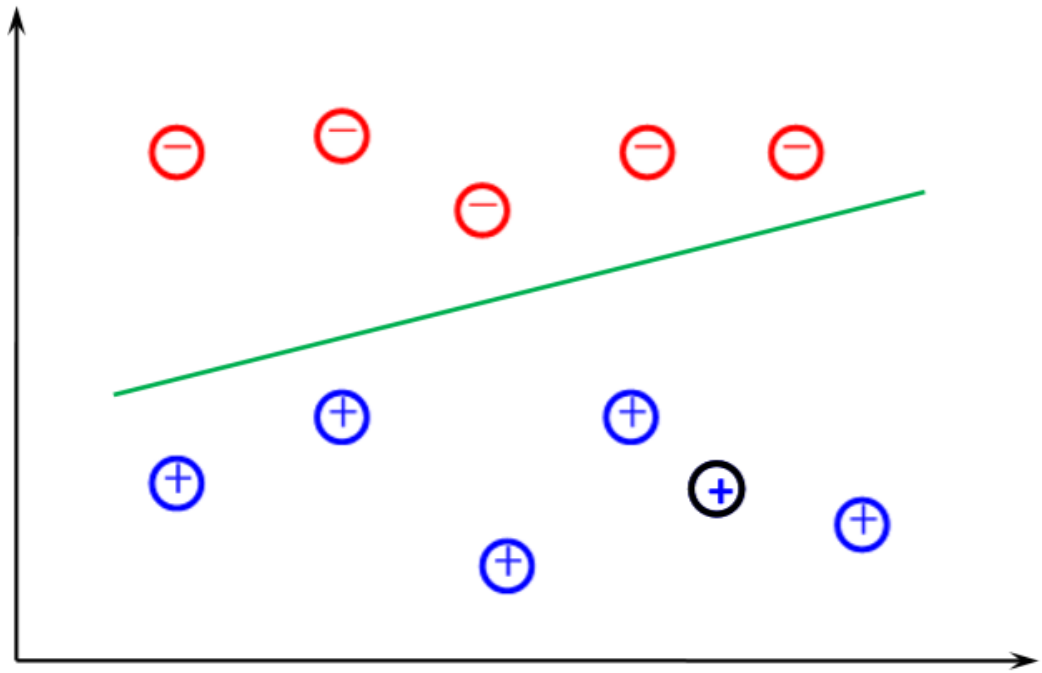
Table of Contents

Supervised learning

- Supervised learning
  - An agent or machine is given N sensory inputs $D = \{x_1, x_2, \ldots, x_N\}$, as well as the desired outputs $y_1, y_2, \ldots, y_N$, its goal is to learn to produce the correct output given a new input.
  - Given D what can we say about x_{N+1}?
- Classification: $y_1, y_2, \ldots, y_N$ are discrete class labels, learn a labeling function $f(\mathbf{x}) = y$
  - Naïve bayes
  - Decision tree
  - K nearest neighbor
  - Least squares classification
  - Logistic regression

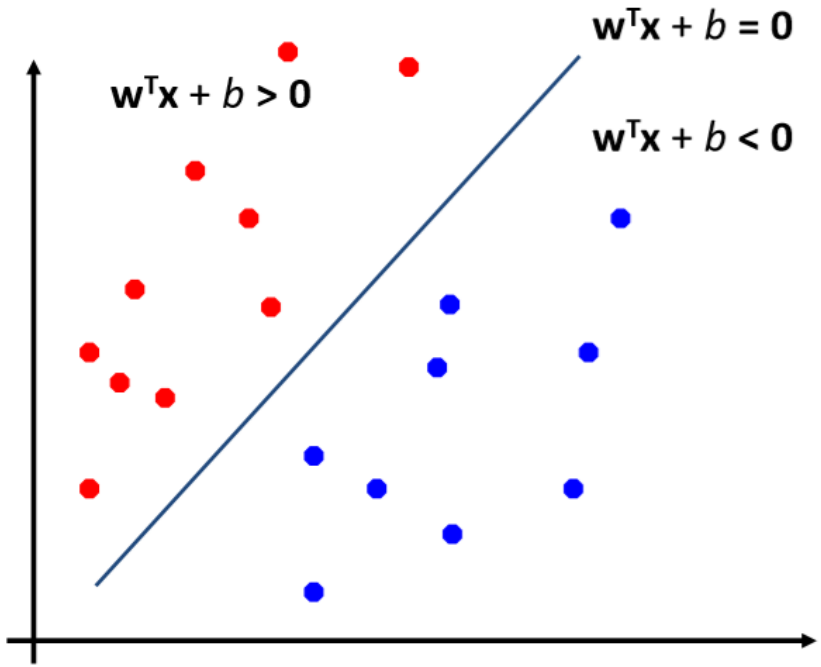
Classification

Classification = learning from labeled data. Dominant problem in Machine Learning



Linear Classification

Binary classification can be viewed as the task of separating classes in feature space (特征空间):

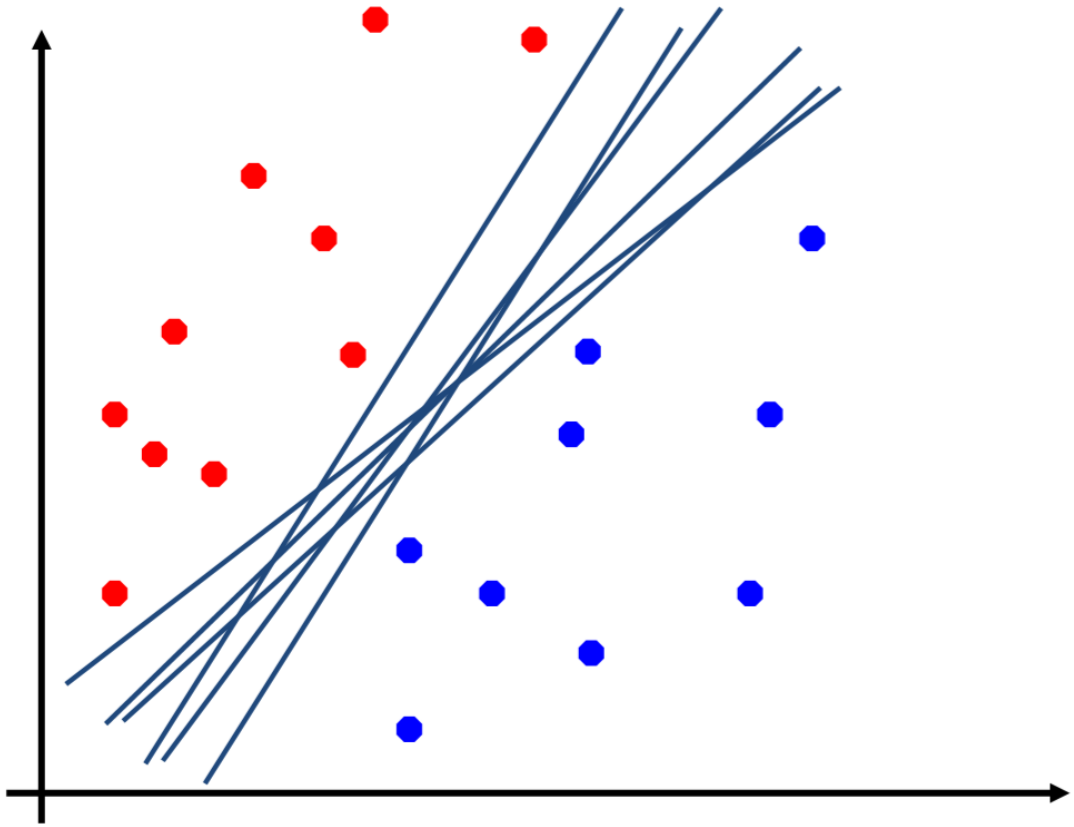


Decide $\hat{y} = 1$ if $\mathbf{w}^T\mathbf{x} + b > 0$,
otherwise $\hat{y} = -1$

$$\hat{y} = h(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + b)$$

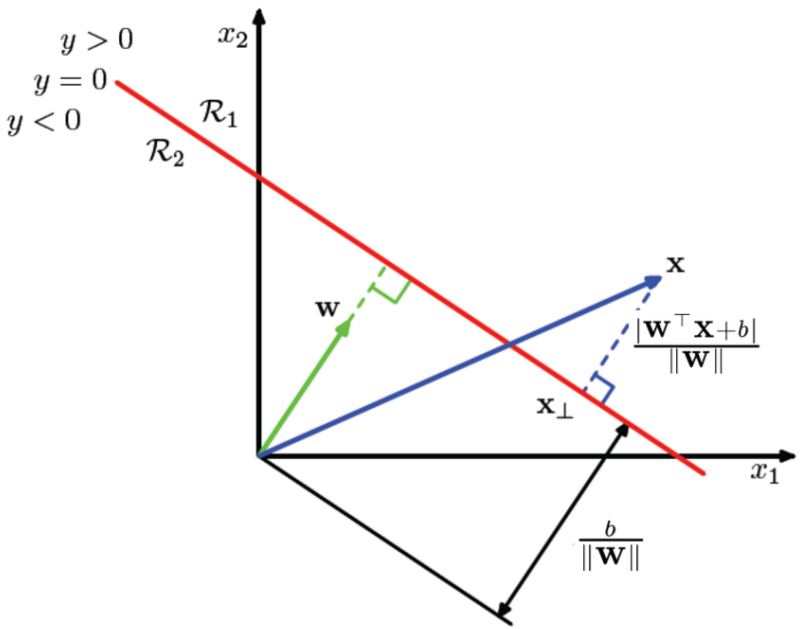
Linear Classification

Which of the linear separators is optimal?



Classification Margin (间距)

- Geometry of linear classification
- Discriminant function $\hat{y}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$
- **范数** (norm): $\|\vec{x}\|$ **表示向量在向量空间中的长度**
  - $\|\vec{x}\|_1 := \sum_{i=1}^n |x_i|$
  - $\|\vec{x}\|_2 := \sqrt{\sum_{i=1}^n x_i^2}$
  - $\|\vec{x}\|_p := \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$
  - $\|\vec{x}\|_\infty := \max_i |x_i|$
- Important: the distance does not change if we scale $\mathbf{w} \to a \cdot \mathbf{w},\ b \to a \cdot b$



点到平面距离公示推导

空间中任意一点 $\mathbf{x}_0$ 到超平面 S：$\mathbf{w}^\top \mathbf{x} + b = 0$ 的距离 d 为：

$$d = \frac{|\mathbf{w}^\top \mathbf{x}_0 + b|}{\|\mathbf{w}\|}$$

注：$\mathbf{x}_0$, $\mathbf{w}$, $\mathbf{x}$ 为 N 维向量

- 设点 $\mathbf{x}_0$ 在平面 S 上的投影为 $\mathbf{x}_1$，则 $\mathbf{w}^\top \mathbf{x}_1 + b = 0$
- 由于向量 $\overrightarrow{\mathbf{x}_0\mathbf{x}_1}$ 与平面 S 的法向量 $\mathbf{w}$ 平行，所以

$$|\mathbf{w} \cdot \overrightarrow{\mathbf{x}_0\mathbf{x}_1}| = \|\mathbf{w}\| \, \|\overrightarrow{\mathbf{x}_0\mathbf{x}_1}\| = \|\mathbf{w}\| \, d$$

向量点积公式：$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos\theta$，$\theta$ 为 $\mathbf{a}$ 和 $\mathbf{b}$ 的夹角

- 同时

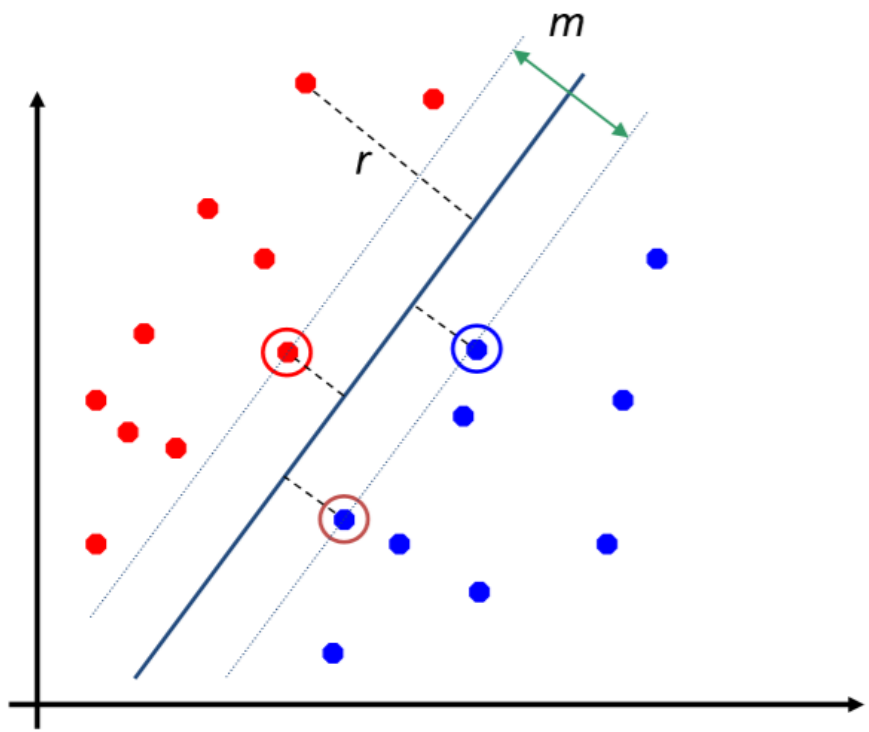
$$\begin{aligned}
\mathbf{w} \cdot \overrightarrow{\mathbf{x}_0\mathbf{x}_1} &= w^1(x_0^1 - x_1^1) + w^2(x_0^2 - x_1^2) + \cdots + w^N(x_0^N - x_1^N) \\
&= w^1 x_0^1 + w^2 x_0^2 + \cdots + w^N x_0^N - (w^1 x_1^1 + w^2 x_1^2 + \cdots + w^N x_1^N) \\
&= w^1 x_0^1 + w^2 x_0^2 + \cdots + w^N x_0^N - (-b)
\end{aligned}$$

- 所以

$$\|\mathbf{w}\| d = |\mathbf{w}^\top \mathbf{x}_0 + b|$$

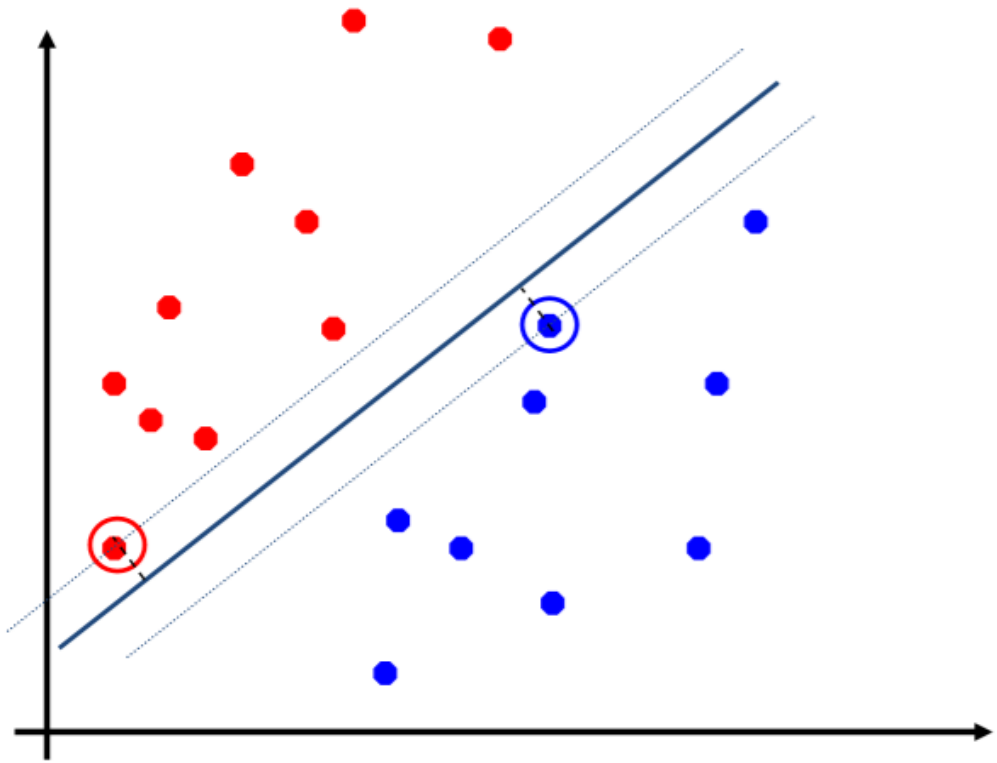
Classification Margin (间距)

- Distance from example $\mathbf{x}_i$ to the separator is $r = \frac{|\mathbf{w}^\top \mathbf{x}_i + b|}{\|\mathbf{w}\|}$
- Examples closest to the hyperplane (超平面) are support vectors (支持向量).
- Margin m of the separator is the distance between support vectors



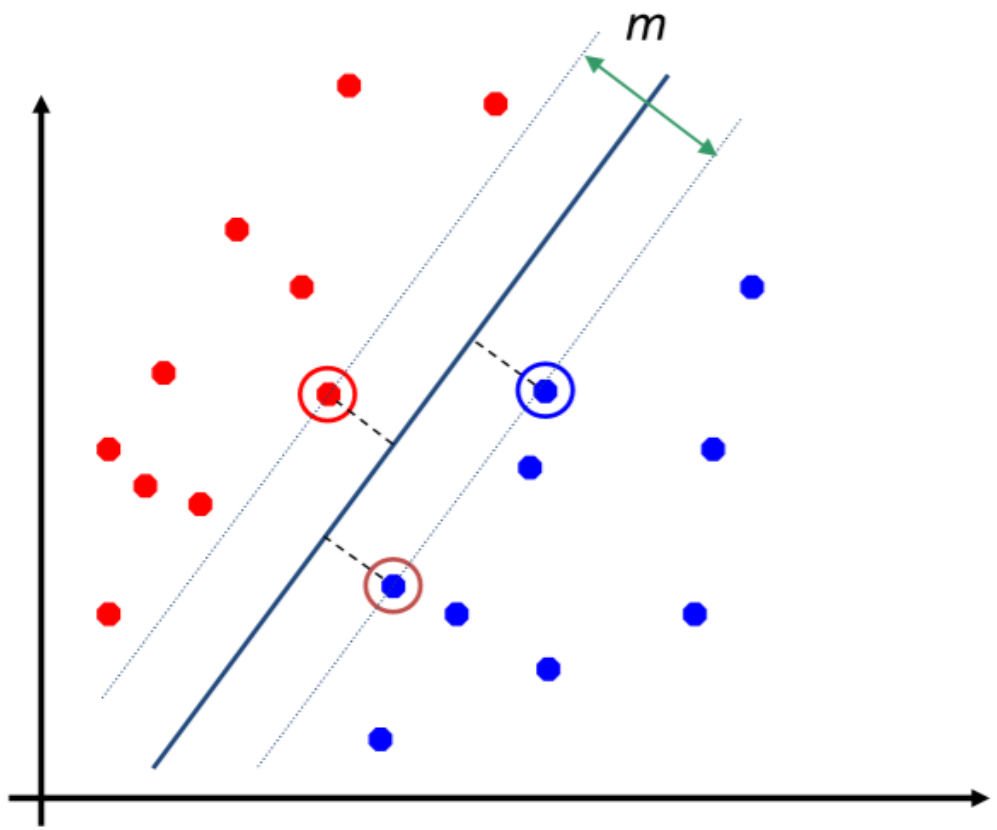
Maximum Margin Classification (最大间距分类)

- Maximizing the margin is good according to intuition and PAC theory.
- Implies that only support vectors matter; other training examples are ignorable.



Maximum Margin Classification (最大间距分类)

- Maximizing the margin is good according to intuition and PAC theory.
- Implies that only support vectors matter; other training examples are ignorable.



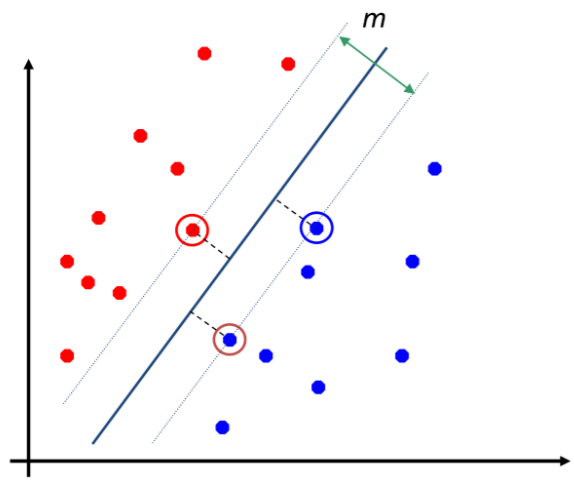
Maximum Margin Classification Mathematically

- Let training set $\{(\mathbf{x}_i, y_i)\}_{i=1..n'}, \mathbf{x}_i \in \mathbf{R}^d, y_i \in \{-1, 1\}$ be separated by a hyperplane with margin m. Then for each training example $(\mathbf{x}_i, y_i)$:

$$\begin{array}{ll} \mathbf{w}^\top \mathbf{x}_i + b \leq -c & \text{if } y_i = -1 \\ \mathbf{w}^\top \mathbf{x}_i + b \geq c & \text{if } y_i = 1 \end{array} \Leftrightarrow y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq c$$

- For every support vector $\mathbf{x}_s$ the above inequality is an equality. $y_s \left(\mathbf{w}^\top \mathbf{x}_s + b\right) = c$
- In the equality, we obtain that distance between each $\mathbf{x}_s$ and the hyperplane is

$$r = \frac{\left|\mathbf{w}^T \mathbf{x}_s + b\right|}{\|\mathbf{w}\|} = \frac{y_s\left(\mathbf{w}^T \mathbf{x}_s + b\right)}{\|\mathbf{w}\|} = \frac{c}{\|\mathbf{w}\|}$$



Maximum Margin Classification Mathematically

- Then the margin can be expressed through $\mathbf{w}$ and b : $m = 2r = \frac{2c}{\|\mathbf{w}\|}$
- Here is our Maximum Margin Classification problem:

$$\max_{\mathbf{w},b} \frac{2c}{\|\mathbf{w}\|} \text{ subject to } y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq c, \forall i$$

$$\max_{\mathbf{w},b} \frac{c}{\|\mathbf{w}\|} \text{ subject to } \quad y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq c, \forall i$$

- Note that the magnitude (大小) of c merely scales $\mathbf{w}$ and b, and does not change the classification boundary at all!
- So we have a cleaner problem:

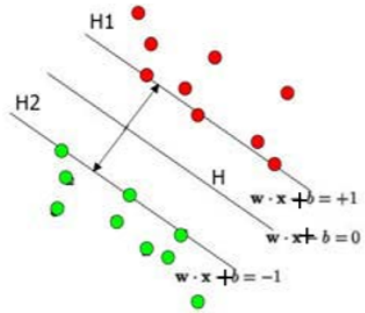
$$\max_{\mathbf{w},b} \frac{1}{\|\mathbf{w}\|} \quad \text{subject to} \quad y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1, \forall i$$

- This leads to the famous Support Vector Machines 支持向量机—believed by many to be the best "off-the-shelf" supervised learning algorithm

Support Vector Machine

- A convex quadratic programming (凸二次规划) problem with linear constraints:

$$\max_{\mathbf{w},b} \frac{1}{\|\mathbf{w}\|} \quad \text{subject to} \quad y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1$$



The attained margin is now given by $\frac{1}{\|\mathbf{w}\|}$

Only a few of the classification constraints are relevant→ support vectors

- Constrained optimization (约束优化)
  - We can directly solve this using commercial quadratic programming (QP) code
  - But we want to take a more careful investigation of Lagrange duality (拉格朗日对偶), and the solution of the above in its dual form.
  - deeper insight: support vectors, kernels (核)...

Quadratic Programming（二次规划）

- Minimize (with respect to $\mathbf{x}$)

$$g(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top Q\,\mathbf{x} + \mathbf{c}^\top \mathbf{x}$$

- Subject to one or more constraints of the form:

$$A\,\mathbf{x} \leq \mathbf{b} \quad \text{(inequality constraint)}$$
$$E\,\mathbf{x} = \mathbf{d} \quad \text{(equality constraint)}$$

- If $Q \geq 0$, then g(x) is a convex function（凸函数）: In this case the quadratic program has a global minimizer
- Quadratic program of support vector machine:

$$\min_{\mathbf{w},b} \mathbf{w}^\top \mathbf{w} \quad \text{subject to} \quad y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1, \forall i$$

拉格朗日乘子法

- 拉格朗日乘子法：对于一个等式约束的优化问题

$$\min f(x) \qquad \text{s.t. } h(x) = 0$$

等价于 $\min f(x) + \lambda h(x)$，其中 λ 是一个自由变量，可以取任何值。

- 对于不等式的约束

$$\min f(x) \qquad \text{s.t. } g(x) \leq 0$$

等价于

$$\min_x \max_\lambda f(x) + \lambda g(x) \qquad \text{s.t. } \lambda \geq 0$$

KKT 条件

- 对于不等式条件约束的优化，我们没有办法消除它的约束。但是我们能够推导出它的最优解一定满足的几个性质。

$$\min_{x} \max_{\lambda} f(x) + \lambda g(x) \qquad \text{s.t. } \lambda \geq 0$$

最优解 x^* 满足以下条件（KKT 条件）：

  - $g(x^*) \leq 0$
  - $\lambda \geq 0$
  - $\lambda g(x^*) = 0$
  - $\nabla f(x^*) + \lambda \nabla g(x^*) = 0$

- 满足 KKT 条件的不一定是最优解，但最优解一定满足 KKT 条件

对偶问题

- 对于一个不等式约束的原问题，定义对偶问题为：

$$\max_{\lambda} \min_{x} f(x) + \lambda g(x) \qquad \text{s.t. } \lambda \geq 0$$

  将 min 和 max 对调
- 对偶的好处：通常先确定 x 的函数最小值，比原问题先求 λ 的最大值更容易。但原问题跟对偶问题并不是等价的
- 所有对偶问题都满足弱对偶性（原始问题始终大于等于对偶问题）

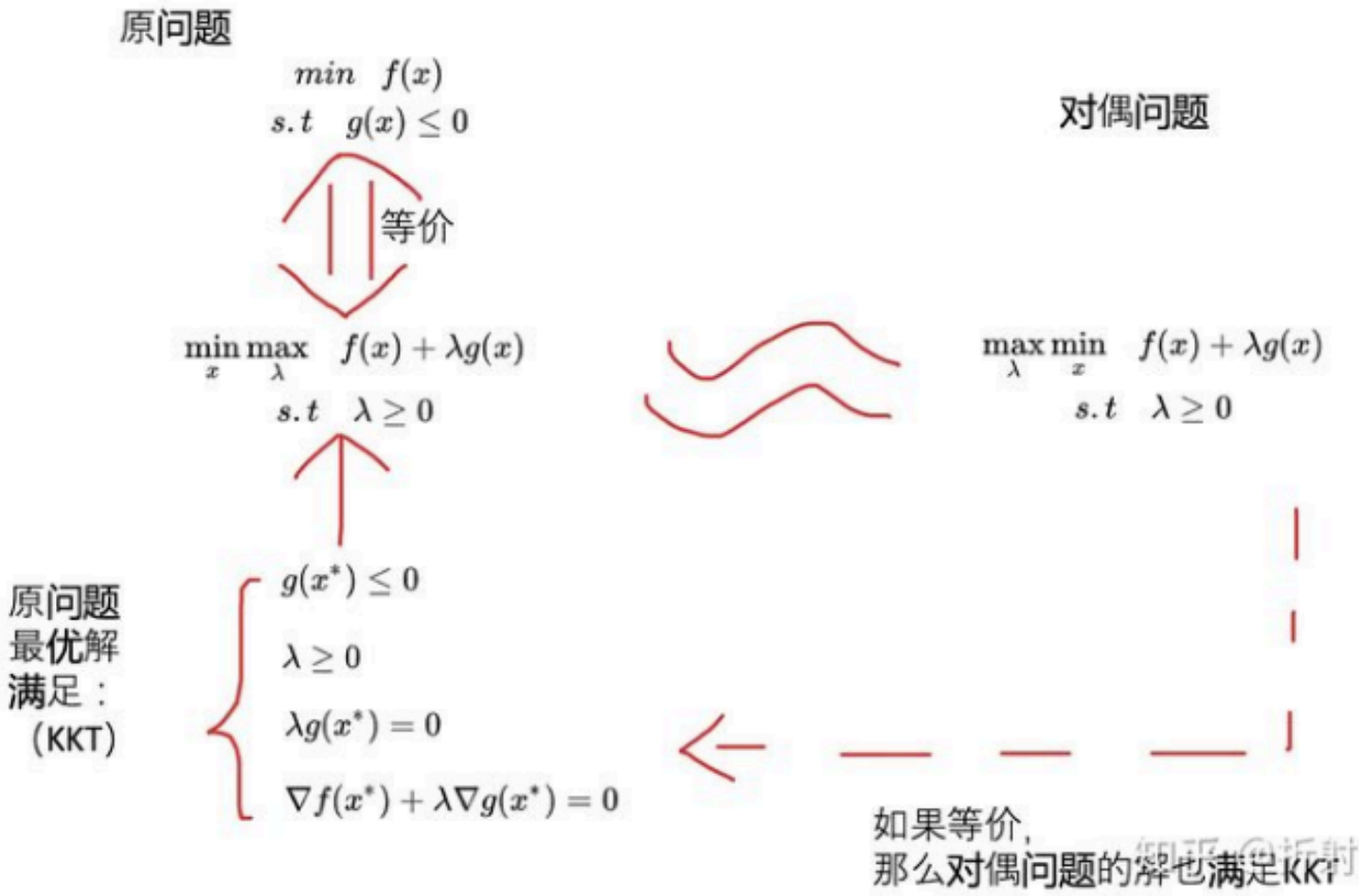
$$\max_{\lambda} \min_{x} f(x, \lambda) = \min_{x} f(x, \lambda^*) \leq \max_{\lambda} f(x^*, \lambda) = \min_{x} \max_{\lambda} f(x, \lambda)$$

- 如果具有强对偶性，则彼此等价：

$$\max_{\lambda} \min_{x} f(x, \lambda) = \min_{x} \max_{\lambda} f(x, \lambda)$$

- 几乎所有的凸优化问题（$f(x)$ 是一个凸函数）都满足强对偶性，在 SVM 中，它的损失函数是凸函数，它是一个强对偶问题

拉格朗日乘子法、KKT 条件、对偶问题



Solving Maximum Margin Classifier

- Our optimization problem:

$$\min_{\mathbf{w},b} \mathbf{w}^\top \mathbf{w} \quad \text{subject to} \quad 1 - y_i \left(\mathbf{w}^\top \mathbf{x}_i + b \right) \leq 0, \forall i$$

- The Lagrangian:

$$\begin{aligned} L(\mathbf{w}, b, \alpha) &= \frac{1}{2} \mathbf{w}^\top \mathbf{w} - \sum_{i=1}^{n} \alpha_i \left[y_i \left(\mathbf{w}^\top \mathbf{x}_i + b \right) - 1 \right] \\ &= \frac{1}{2} \mathbf{w}^\top \mathbf{w} + \sum_{i=1}^{n} \alpha_i \left[1 - y_i \left(\mathbf{w}^\top \mathbf{x}_i + b \right) \right] \end{aligned}$$

- Consider each constraint:

$$\begin{aligned} \max_{\alpha_i \geq 0} \alpha_i \left[1 - y_i \left(\mathbf{w}^\top \mathbf{x}_i + b \right) \right] &= 0 \quad \text{if } \mathbf{w}, b \text{ satisfies primal constraints} \\ &= \infty \quad \text{otherwise} \end{aligned}$$

Solving Maximum Margin Classifier

- Our optimization problem:

$$\min_{\mathbf{w},b} \mathbf{w}^\top \mathbf{w} \quad \text{subject to} \quad 1 - y_i \left(\mathbf{w}^\top \mathbf{x}_i + b \right) \leq 0, \forall i \qquad (1)$$

- The Lagrangian:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \mathbf{w}^\top \mathbf{w} - \sum_{i=1}^{n} \alpha_i \left[y_i \left(\mathbf{w}^\top \mathbf{x}_i + b \right) - 1 \right]$$

- Lemma:

$$\begin{aligned} \max_{\alpha \geq 0} L(\mathbf{w}, b, \alpha) &= \frac{1}{2} \mathbf{w}^\top \mathbf{w} && \text{if } \mathbf{w},\ b \text{ satisfies primal constraints} \\ &= \infty && \text{otherwise} \end{aligned}$$

- (1) can be reformulated as $\min_{\mathbf{w},b} \max_{\alpha \geq 0} L(\mathbf{w}, b, \alpha)$
- The dual problem (**对偶问题**): $\max_{\alpha \geq 0} \min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha)$

The Dual Problem (对偶问题)

$$\max_{\alpha \geq 0} \min_{\mathbf{w}, b} L(\mathbf{w}, b, \alpha)$$

We minimize L with respect to $\mathbf{w}$ and b first:

$$\frac{\partial L}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) = \mathbf{w}^\top - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^\top = 0 \tag{2}$$

$$\frac{\partial L}{\partial \mathbf{b}} L(\mathbf{w}, b, \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3}$$

Note: $d(\mathbf{A}\mathbf{x} + \mathbf{b})^T(\mathbf{A}\mathbf{x} + \mathbf{b}) = \left(2(\mathbf{A}\mathbf{x} + \mathbf{b})^T\mathbf{A}\right) d\mathbf{x}$
$d\left(\mathbf{x}^T\mathbf{a}\right) = d\left(\mathbf{a}^T\mathbf{x}\right) = \mathbf{a}^T d\mathbf{x}$
Note that the bias term b dropped out but had produced a "global" constraint on α

The Dual Problem (对偶问题)

$$\max_{\alpha \geq 0} \min_{\mathbf{w}, b} L(\mathbf{w}, b, \alpha)$$

We minimize L with respect to $\mathbf{w}$ and b first:

$$\frac{\partial L}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) = \mathbf{w}^\top - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^\top = 0 \tag{2}$$

$$\frac{\partial L}{\partial \mathbf{b}} L(\mathbf{w}, b, \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3}$$

Note that (2) implies

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{4}$$

Plug (4) back to L, and using (3), we have

$$L(\mathbf{w}, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i^\top \mathbf{x}_j \right)$$

The Dual Problem (对偶问题)

Now we have the following dual optimization problem:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i^\top \mathbf{x}_j \right) \quad \text{subject to} \quad \alpha_i \geq 0, \forall i, \sum_{i=1}^{n} \alpha_i y_i = 0$$

This is a quadratic programming problem again

- A global maximum can always be found

But what's the big deal?

1. $\mathbf{w}$ can be recovered by $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$
2. b can be recovered by $b = y_i - \mathbf{w}^\top \mathbf{x}_i$ for any i that $\alpha_i \neq 0$
3. The "kernel"— 核 $\mathbf{x}_i^\top \mathbf{x}_j$

Support Vectors

If a point $\mathbf{x}_i$ satisfies $y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) > 1$

Due to the fact that

$$\max_{\alpha_i \geq 0} \alpha_i \left[1 - y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right)\right] = 0 \qquad y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1$$
$$= \infty \qquad \text{otherwise}$$

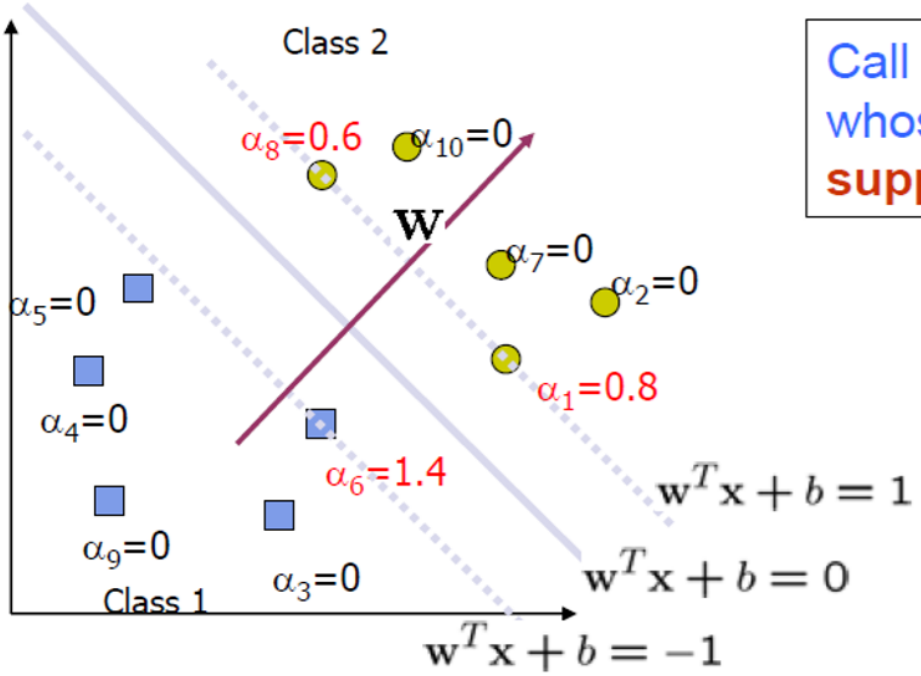
We have $\alpha^* = 0$; $\mathbf{x}_i$ not a support vector

$\mathbf{w}$ *is decided by the points with non-zero* α*'s*

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

Support Vectors

only a few α_i's can be nonzero!!



Call the training data points whose α_i's are nonzero the **support vectors** (SV)

Support Vector Machines

Once we have the Lagrange multipliers α_i, we can reconstruct the parameter vector $\mathbf{w}$ as a weighted combination of the training examples:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = \sum_{i \in SV} \alpha_i y_i \mathbf{x}_i$$

For testing with a new data $\mathbf{x}'$

- Compute $\mathbf{w}^\top \mathbf{x}' + b = \sum_{i \in SV} \alpha_i y_i \left(\mathbf{x}_i^\top \mathbf{x}'\right) + b$
- classify $\mathbf{x}'$ as class 1 if the sum is positive, and class 2 otherwise

Note: $\mathbf{w}$ need not be formed explicitly

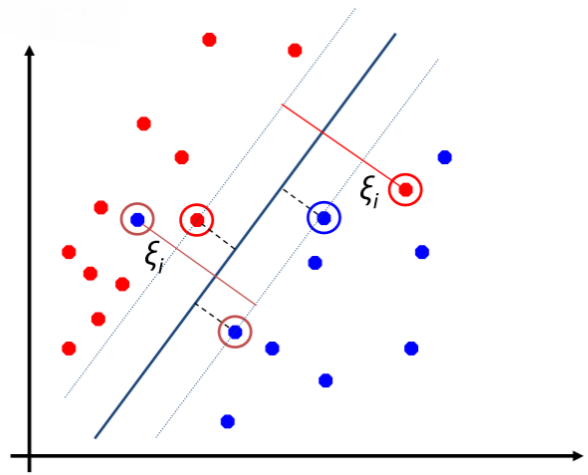
Interpretation of support vector machines

- The optimal $\mathbf{w}$ is a linear combination of a small number of data points. This "sparse 稀疏" representation can be viewed as data compression (数据压缩) as in the construction of kNN classifier
- To compute the weights α_i, and to use support vector machines we need to specify only the inner products 内积 (or kernel) between the examples $\mathbf{x}_i^\top \mathbf{x}_j$
- We make decisions by comparing each new example $\mathbf{x}'$ with only the support vectors:

$$y^* = \text{sign}\left(\sum_{i \in SV} \alpha_i y_i \left(\mathbf{x}_i^\top \mathbf{x}'\right) + b\right)$$

Soft Margin Classification

- What if the training set is not linearly separable?
- Slack variables (松弛变量) ξ_i can be added to allow misclassification of difficult or noisy examples, resulting margin called soft.
  - 分类正确的点 $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$，左侧的值越大，则点 $\mathbf{x}_i$ 离超平面 $\mathbf{w}^\top \mathbf{x} + b = 0$ 越远；分类错误的点 $y_i(\mathbf{w}^\top \mathbf{x}_i + b) < 1$，越小偏离越远
  - 用 ξ_i 来衡量这种偏离度，使得 $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i$，同时要求 ξ_i 尽可能小
  - 对于正确的点，本来就大于 1，ξ_i 自然为 0；而错误的点则需要 $1 - \xi_i$ 来纠正偏离，ξ_i 就衡量了点犯错误的大小程度



Soft Margin Classification Mathematically

- "Hard" margin QP:

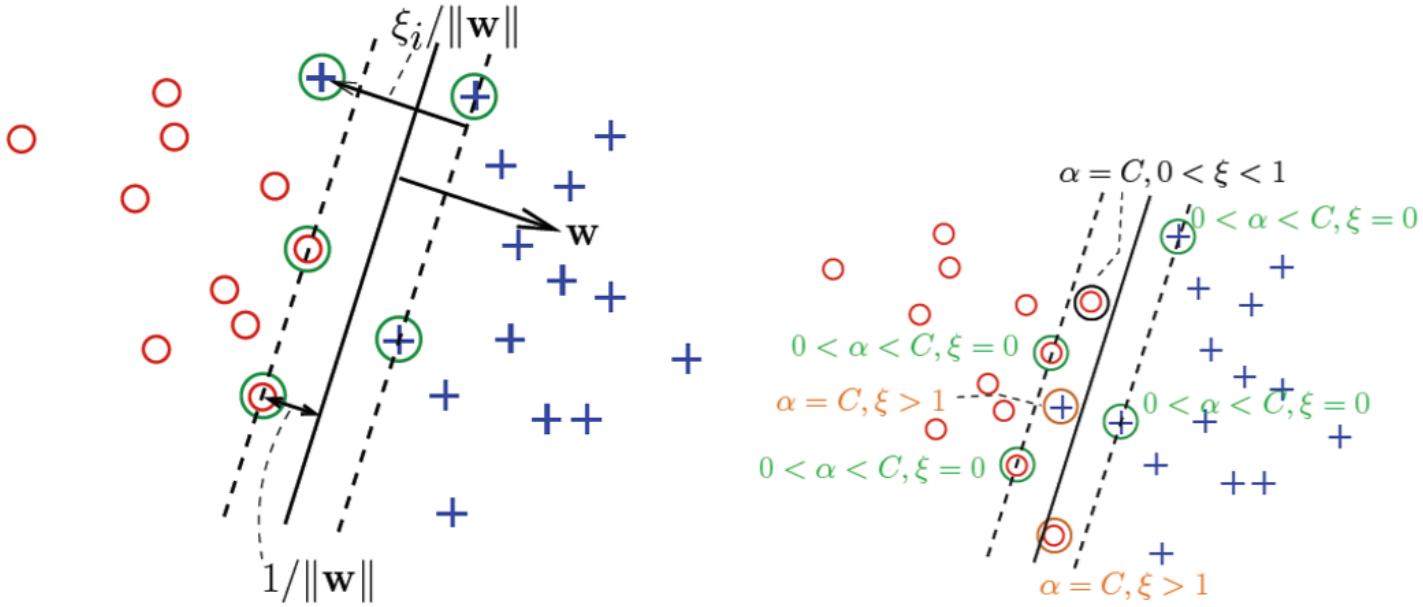
$$\min_{W,b} \mathbf{w}^\top \mathbf{w} \quad \text{subject to} \quad \forall i,\ y_i \left(\mathbf{w}^\top \mathbf{x}_i + b \right) \geq 1$$

- "Soft" margin QP:

$$\min_{\mathbf{w},b} \frac{1}{2} \mathbf{w}^\top \mathbf{w} + C \sum_i \xi_i \text{ subject to } \forall i,\ y_i \left(\mathbf{w}^\top \mathbf{x}_i + b \right) \geq 1 - \xi_i,\ \xi_i \geq 0$$

- Note that $\xi_i = 0$ if there is no error for $\mathbf{x}_i$
- ξ_i is an upper bound of the number of errors
- Parameter C can be viewed as a way to control "softness": it "trades off" the relative importance of maximizing the margin and fitting the training data (minimizing the error).
  - Larger $C \rightarrow$ more reluctant to make mistakes
  - Larger C 使得结果越接近 Hard Margin SVM

SVMs with slack variables



- Support vectors: points with $\alpha > 0$
- If $0 < \alpha < C$: SVs on the margin, $\xi = 0$
- If $0 < \alpha = C$: SVs over the margin, either misclassified $(\xi > 1)$ or not $(0 < \xi \leq 1)$

The Optimization Problem

- The dual of this new constrained optimization problem is

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i^\top \mathbf{x}_j \right)$$

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

- This is very similar to the optimization problem in the linear separable case, except that there is an upper bound C on α_i now
- Once again, a QP solver can be used to find α_i

Loss in SVM

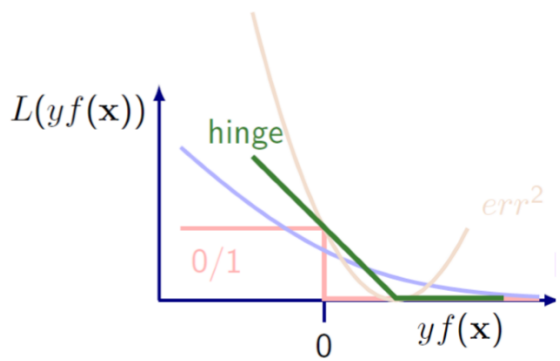
$$\min_{\mathbf{w},b} \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i \text{ subject to } \forall i,\ y_i\left(\mathbf{w}^\top\mathbf{x}_i + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0$$

Loss is measured as

$$\xi_i = \max\left(0, 1 - y_i\left(\mathbf{w}^\top\mathbf{x}_i + b\right)\right) = \left[1 - y_i\left(\mathbf{w}^\top\mathbf{x}_i + b\right)\right]_+$$

This loss is known as hinge loss

$$\min_{\mathbf{w},b} \frac{1}{2C}\mathbf{w}^\top\mathbf{w} + \sum_i hingeloss_i$$



Linear SVMs: Overview

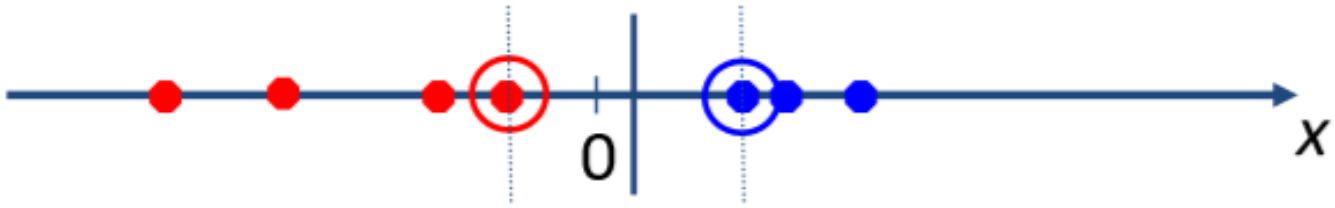
- The classifier is a separating hyperplane.
- Most "important" training points are support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points $\mathbf{x}_i$ are support vectors with non-zero Lagrangian multipliers α_i.
- Both in the dual formulation of the problem and in the solution training points appear only inside inner products:

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

$$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^T\mathbf{x} + b$$

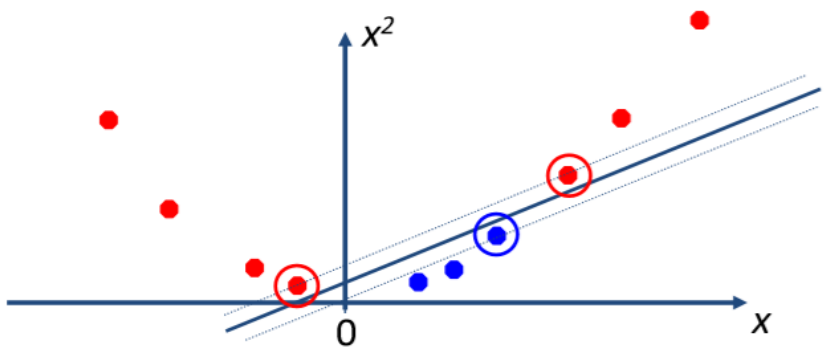
Non-linear SVMs

- Datasets that are linearly separable with some noise work out great:



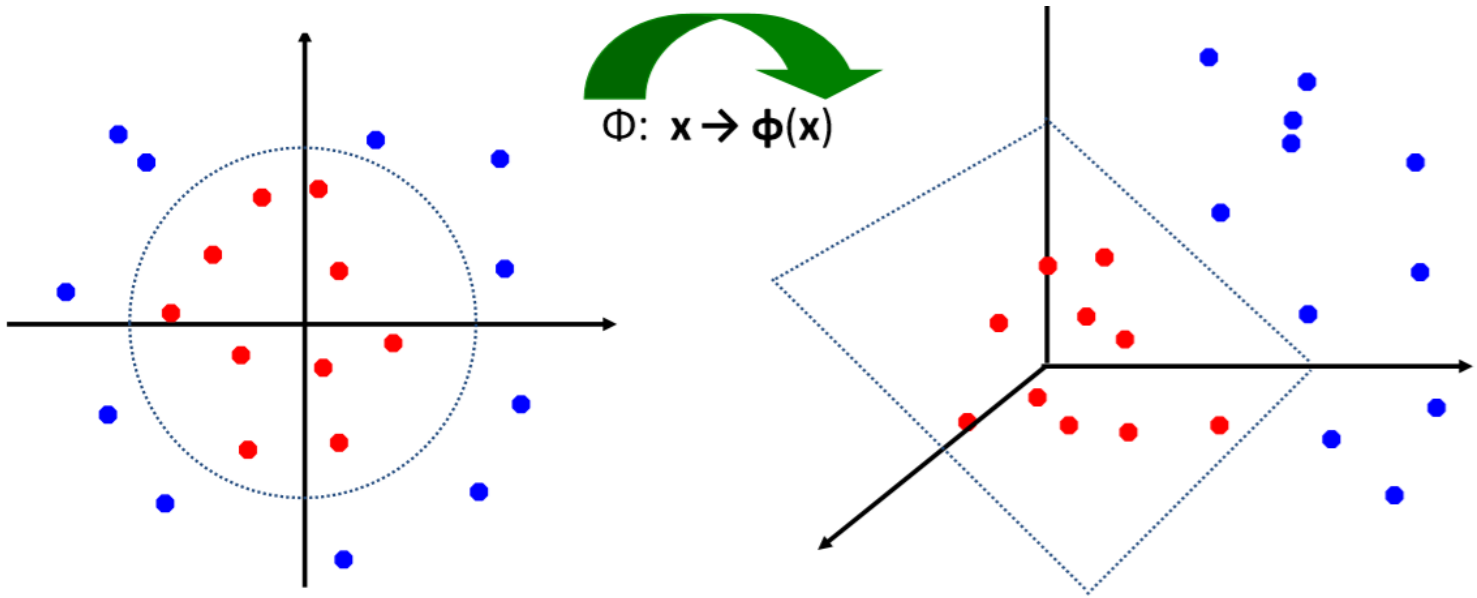
- But what are we going to do if the dataset is just too hard?

- How about $\cdots$ mapping data to a higher-dimensional space:



Non-linear SVMs: Feature spaces

General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:



The "Kernel Trick"

- Recall the SVM optimization problem

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i^\top \mathbf{x}_j \right)$$

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

- The data points only appear as inner product
  - As long as we can calculate the inner product in the feature space, we do not need the mapping explicitly
  - Many common geometric operations (angles, distances) can be expressed by inner products
- Define the kernel function K by $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^\top \varphi(\mathbf{x}_j)$

An Example for feature mapping and kernels

- Consider an input $\mathrm{x} = [x_1, x_2]^\top$
- Suppose $\phi(\cdot)$ is given as follows

$$\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \left[1,\ \sqrt{2}x_1,\ \sqrt{2}x_2,\ x_1^2,\ x_2^2,\ \sqrt{2}x_1x_2\right]^\top$$

- An inner product in the feature space is

$$\left\langle \phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right), \phi\left(\begin{bmatrix} x_1' \\ x_2' \end{bmatrix}\right) \right\rangle = \phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right)^\top \phi\left(\begin{bmatrix} x_1' \\ x_2' \end{bmatrix}\right)$$

- So, if we define the kernel function as follows, there is no need to carry out $\phi(\cdot)$ explicitly

$$K\left(\mathbf{x}, \mathbf{x}'\right) = \left(1 + \mathbf{x}^\top \mathbf{x}'\right)^2$$

More Examples of Kernel Functions

- Linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$
  - Mapping $\Phi : \mathbf{x} \to \varphi(\mathbf{x})$, where $\varphi(\mathbf{x})$ is $\mathbf{x}$ itself
- Polynomial (多项式) of power p: $K(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \mathbf{x}_i^\top \mathbf{x}_j\right)^p$
- Gaussian (radial-basis function 径向基函数):
  $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}}$
  - Mapping $\Phi : \mathbf{x} \to \varphi(\mathbf{x})$, where $\varphi(\mathbf{x})$ is infinite-dimensional
- Higher-dimensional space still has intrinsic dimensionality d (the mapping is not onto), but linear separators in it correspond to non-linear separators in original space

Kernel matrix

- Suppose for now that K is indeed a valid kernel corresponding to some feature mapping ϕ, then for $x_1, \ldots, x_n$, we can compute an $n \times n$ matrix $\{K_{i,j}\}$ where $K_{i,j} = \varphi(\mathbf{x}_i)^\top \varphi(\mathbf{x}_j)$
- This is called a kernel matrix!
- Now, if a kernel function is indeed a valid kernel, and its elements are dot-product in the transformed feature space, it must satisfy:
  - Symmetry $K = K^\top$
  - Positive-semidefinite (**半正定**) $\mathbf{z}^\top K\mathbf{z} \geq 0, \quad \forall \mathbf{z} \in R^n$

Matrix formulation

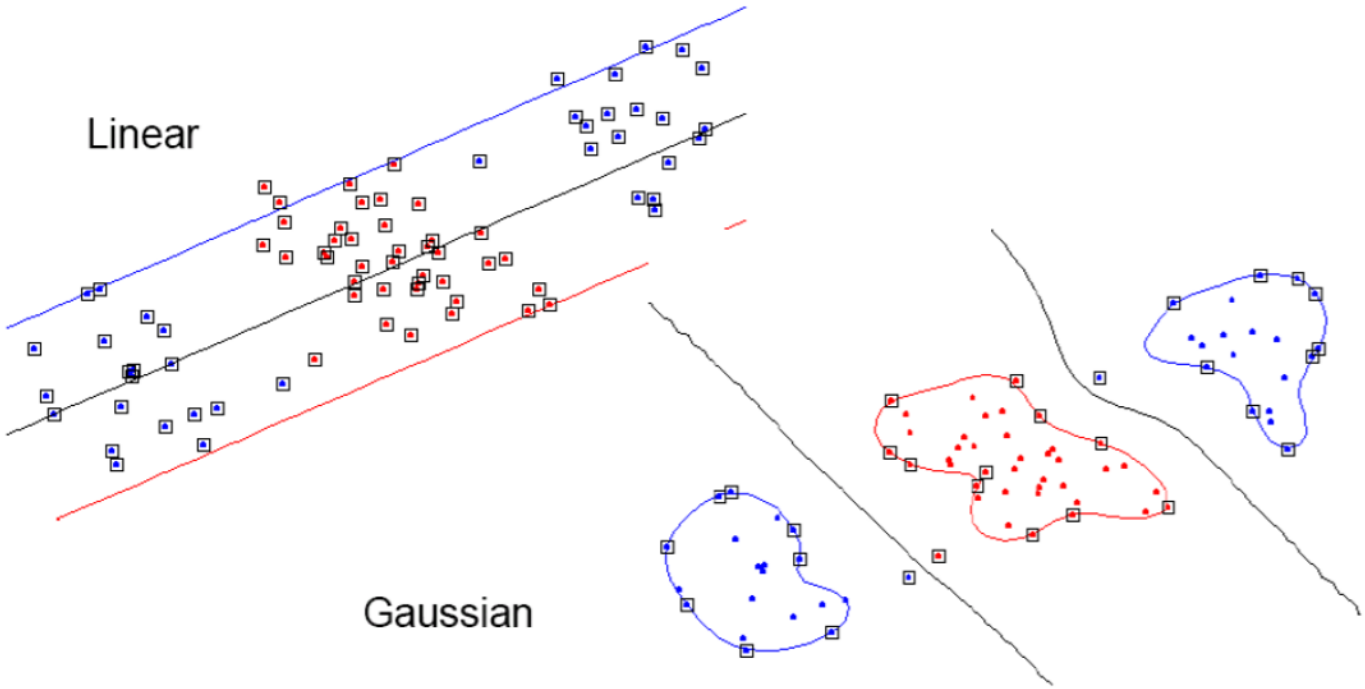
$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$= \max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K_{i,j}$$

$$= \max_{\alpha} \alpha^\top \mathbf{e} - \frac{1}{2} \alpha^\top \left(\mathbf{y}\mathbf{y}^\top \circ K \right) \alpha$$

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

Nonlinear SVMs – RBF Kernel



Summary: Support Vector Machines

- Linearly separable case $\rightarrow$ Hard margin SVM
  - Primal quadratic programming
  - Dual quadratic programming
- Not linearly separable? $\rightarrow$ Soft margin SVM
- Non-linear SVMs – Kernel trick

Summary: Support Vector Machines

- SVM training: build a kernel matrix K using training data
  - Linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^{\mathbf{T}} \mathbf{x}_j$
  - Gaussian (radial-basis function 径向基函数):

    $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}}$
- Solve the following quadratic program :

$$\max_{\alpha} \alpha^\top \mathbf{e} - \frac{1}{2} \alpha^\top \left(\mathbf{y}\mathbf{y}^\top \circ K \right) \alpha$$

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

- SVM testing: now with α_i, recover b

$$b = y_i - \sum_{j=1}^{n} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \qquad \text{for any } i \text{ that } \alpha_i \neq 0$$

- we can predict new data points by:

$$y^* = \text{sign} \left(\sum_{i \in SV} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}') + b \right)$$

Machine Learning with Scikit-Learn

Scikit-Learn

- Machine learning library written in Python
- Simple and efficient, for both experts and non-experts
- Classical, well-established machine learning algorithms
- Shipped with documentation and examples
  - documentation: https://scikit-learn.org/dev/index.html
  - examples: https://scikit-learn.org/dev/auto_examples/index.html

作业 1

- 试证明对于不含冲突数据集（即特征向量完全相同但标记不同）的训练集，必存在与训练集一致（即训练误差为 0）的决策树。
- 最小二乘学习方法在求解 $\min_{\mathbf{w}}(\mathbf{X}\mathbf{w}-\mathbf{y})^2$ 问题后得到闭式解 $\mathbf{w}^*=(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$（为简化问题，我们忽略偏差项 $\mathbf{b}$）。如果我们知道数据中部分特征有较大的误差，在不修改损失函数的情况下，引入规范化项 $\lambda\mathbf{w}^T\mathbf{D}\mathbf{w}$，其中 $\mathbf{D}$ 为对角矩阵，由我们取值。相应的最小二乘分类学习问题转换为以下形式的优化问题：

$$\min_{\mathbf{w}}(\mathbf{X}\mathbf{w}-\mathbf{y})^2+\lambda\mathbf{w}^T\mathbf{D}\mathbf{w}$$

(1) 请说明选择规范化项 $\mathbf{w}^T\mathbf{D}\mathbf{w}$ 而非 L2 规范化项 $\mathbf{w}^T\mathbf{w}$ 的理由是什么。$\mathbf{D}$ 的对角线元素 D_{ii} 有何意义，它的取值越大意味着什么？

(2) 请对以上问题进行求解。

作业 2

- 假设有 n 个数据点 $\mathbf{x}_1, \ldots, \mathbf{x}_n$ 以及一个映射 $\varphi\colon\ \mathbf{x} \to \varphi(\mathbf{x})$，以此定义核函数 $K(\mathbf{x}, \mathbf{x}') = \varphi(\mathbf{x}) \cdot \varphi(\mathbf{x}')$。试证明由该核函数所决定的核矩阵 K：$K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$ 有以下性质:
  (1) K 是一个对称矩阵；
  (2) K 是一个半正定矩阵，即 $\forall \mathbf{z} \in \mathbf{R}^n,\ \mathbf{z}^T K \mathbf{z} \geq 0$。

作业 3

- 已知正例点 $x_1 = (1, 2)^T$, $x_2 = (2, 3)^T$, $x_3 = (3, 3)^T$，负例点 $x_4 = (2, 1)^T$, $x_5 = (3, 2)^T$，试求 Hard Margin SVM 的最大间隔分离超平面和分类决策函数，并在图上画出分离超平面、间隔边界及支持向量。

作业 4

- 计算 $\frac{\partial}{\partial w_j} L_{CE}(\mathbf{w}, b)$，其中

$$L_{CE}(\mathbf{w}, b) = -[y \log \sigma(\mathbf{w} \cdot \mathbf{x} + b) + (1 - y) \log(1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b))]$$

为 Logistic Regression 的 Loss Function。

- 已知

$$\begin{aligned} \frac{\partial}{\partial z}\sigma(z) = \frac{\partial}{\partial z}\frac{1}{1+e^{-z}} &= -\left(\frac{1}{1+e^{-z}}\right)^2 \times (-e^{-z}) \\ &= \sigma^2(z)\left(\frac{1-\sigma(z)}{\sigma(z)}\right) = \sigma(z)(1-\sigma(z)) \end{aligned}$$