Support Vector Machine

Fish

2019 年 6 月 25 日

Content

第一章 perception

1.1 concept

1. 感知机是二类分类的线性分类模型, 其输入空间为实例的特征向量, 输出为实例的类别, 取 +1 和-1 二值.

2. 感知机对应于输入控件（(特征空间) 中将实例划分为正负两类的分离超平面, 是一种线性分类模型, 属于判别模型.

3. 感知机学习旨在求出将训练数据进行线性划分的分离超平面.

4. 导入基于误分类的损失函数, 利用梯度下降法对损失函数进行极小化, 求得感知机模型.

5. $f(x) = sign(w \cdot x + b)$

1.2 linearly separable dataset

给定数据集 $T = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, 其中，$x_i \in \chi = \mathrm{R}^n$, $y_i \in \gamma = \{+1, -1\}$, $i = 1, 2, ..., N$.

如果存在某个超平面 S: $w \cdot x + b = 0$ 将数据集的正负实例点完全正确划分到超平面的两侧：

对所有 $y_i = +1$ 的实例 i, 有 $w \cdot x_i + b > 0$

对所有 $y_i = -1$ 的实例 i, 有 $w \cdot x_i + b < 0$

则数据集 T 为线性数据可分数据集, 否则称数据集 T 线性不可分

1.3 loss function

假设训练数据集是线性可分的, 为了找出可将训练集正负实例点完全正确分开的分离超平面, 即确定参数 w, b, 需要确定一个学习策略, 即定义 (经验) 损失函数并将损失函数极小化

两种选择：

1. 选择误分类点的总数, 但这样的损失函数不是参数 w, b 的连续可导函数, 不易优化

2. 选择误分类点到超平面 S 的总距离

定义输入空间 R^n 中任一点 x_0 到超平面 S 的距离：

$$\frac{1}{\parallel w \parallel}|w \cdot x_0 + b| \tag{1.1}$$

这里 $\parallel w \parallel$ 是 w 的 L_2 范数.

对于误分类的数据 (x_i, y_i) 来说：

$$-y_i(w \cdot w_i + b) > 0 \tag{1.2}$$

成立.

因此, 误分类点 x_i 到超平面 S 的距离是

$$-\frac{1}{\parallel w \parallel} y_i(w \cdot x_i + b) \tag{1.3}$$

这样, 假设超平面 S 的误分类点集合为 M, 则所有误分类点到超平面 S 的总距离为

$$-\frac{1}{\parallel w \parallel} \sum_{x_i \in M} y_i(w \cdot x_i + b) \tag{1.4}$$

不考虑 $\frac{1}{\|w\|}$, 得到感知机 $sign(w \cdot x + b)$ 在训练集 T 的损失函数定义为

$$L(w, b) = -\sum_{x_i \in M} y_i(w \cdot x_i + b) \tag{1.5}$$

1.4 original form of perception algorithm

关键: 感知机算法是误分类驱动的, 当误分类点位于分离超平面的错误一侧时, 调整 w, b 的值, 使分离超平面向该误分类点的一侧移动, 以减少该误分类点与超平面间的距离, 直至超平面越过该误分类点使其被正确分类.

输入: 训练数据集 $T = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, 其中，$x_i \in \chi = \mathrm{R}^n$, $y_i \in \gamma = \{+1, -1\}$, $i = 1, 2, ..., N$; 学习率 η $(0 < \eta \leq 1)$.

输出: w, b; 感知机模型 $f(x) = sign(w \cdot x + b)$.

1. 选取初值 w_0, b_0

2. 在训练集中选取数据 (x_i, y_i)

3. 如果 $y_i(w_i + b) \leq 0$

   (a) 损失函极小化 $\min\limits_{w,b} L(w,b) = -\sum\limits_{x_i \in M} y_i(w \cdot x_i + b)$

   (b) 梯度

$$\nabla_w L(w,b) = -\sum_{x_i \in M} y_i x_i \tag{1.6}$$

$$\nabla_b L(w,b) = -\sum_{x_i \in M} y_i \tag{1.7}$$

   (c) 随机选取一个误分类点 (x_i, y_i), 更新参数 w, b

$$w \leftarrow w + \eta y_i x_i \tag{1.8}$$

$$b \leftarrow b + \eta y_i \tag{1.9}$$

4. 转至 (2), 直至训练集中没有误分类点。

1.5 dual form of perception algorithm

基本思路: 将 w 和 b 表示为实例 x_i 和标记 y_i 的线性组合的形式, 通过求解其系数而求得 w 和 b.

输入：训练数据集 $T = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, 其中，$x_i \in \chi = \mathrm{R}^n$, $y_i \in \gamma = \{+1, -1\}$, $i = 1, 2, ..., N$; 学习率 η $(0 < \eta \leq 1)$.

输出: α, b; 感知机模型 $f(x) = sign\left(\sum\limits_{j=1}^{N} \alpha_j y_j x_j \cdot x + b\right)$

其中, $\alpha = (\alpha_1, \alpha_2, ..., \alpha_N)^T$, 注意这里是一个向量, 所以后面更新时, α 直接定义步长更新

1. $\alpha \leftarrow 0, b \leftarrow 0$

2. 在训练集中选取数据 (x_i, y_i)

3. 如果 $y_i\left(\sum\limits_{j=1}^{N} \alpha_j y_j x_j \cdot x_i + b\right)$

   (a) 损失函极小化 $\min\limits_{w,b} L(w,b) = \min\limits_{\alpha,b} L(\alpha,b) = -\sum\limits_{x_i \in M} y_i(\alpha_j y_j x_j \cdot x_i + b)$

   (b) 梯度

$$\nabla_b L(w,b) = -\sum_{x_i \in M} y_i \tag{1.10}$$

(c) 随机选取一个误分类点 (x_i, y_i), 更新参数 w, b

$$\alpha \leftarrow \alpha + \eta \tag{1.11}$$

$$b \leftarrow b + \eta y_i \tag{1.12}$$

4. 转至 (2) 直到没有误分类数据.

注意: 对偶形式中训练实例仅以内积的形式出现.

为了方便, 预先将训练集中实例间的内积计算出来并以矩阵的形式存储，即 Gram 矩阵

$$G = [x_i \cdot x_j] \tag{1.13}$$

第二章 Convex function

2.1 Convex collection and Convex function

2.1.1 Convex function

$y = x^2$ 是凸函数, 函数图像上位于 $y = x^2$ 上方的区域构成凸集.

1. 凸函数图像的上方区域, 一定是凸集
2. 一个函数图像的上方区域为凸集, 则该函数是凸函数

2.1.2 Convex collection

1. Afiine set 仿射集
   (a) define: 通过集合 C 中任意两个不同点的直线仍然在集合 C 内, 则称集合 C 为仿射集.

$$\forall x_1, x_2 \in C, \forall \theta \in R, \text{ 则 } x = \theta \cdot x_1 + (1-\theta) \cdot x_2 \in C \tag{2.1}$$

   (b) 仿射集的例子: 直线、平面、超平面
   n 维空间的 $n-1$ 维仿射集为 $n-1$ 维超平面

2. Convex 凸集
   两种表述 (可以思考其内涵一样吗) :
   (a) 集合 C 内任意两点间的线段均在集合 C 内, 则称集合 C 为凸集.

$$\forall x_1, x_2 \in C, \forall \theta \in [0,1], \text{ 则 } x = \theta \cdot x_1 + (1-\theta) \cdot x_2 \in C \tag{2.2}$$

   (b) k 个点的版本:

$$\forall x_1, x_2, ..., x_k \in C, \theta_i \in [0,1] \text{且} \sum_{i=1}^{k} \theta_i x_i \in C \tag{2.3}$$

3. 因为仿射集的条件比凸集的条件强, 所以, 仿射集必然是凸集

4. judgement of convex collection

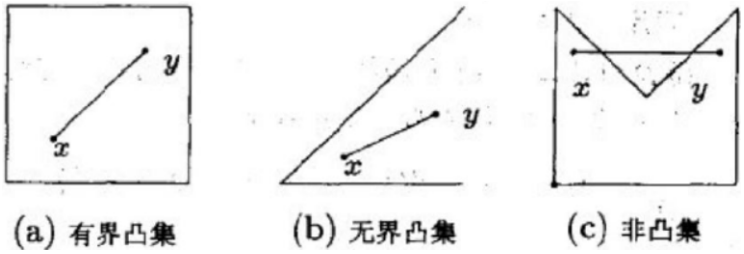


Fig 2.1: convex collection

2.2 affine transformation

2.2.1 concept

函数 $f(x) = Ax + b$ 的形式, 称函数是放射的: 即线性函数加常数的形式

2.2.2 theory

1. 仿射变换 $f(x) = Ax + b$, $A \in \mathrm{R}^{m \times n}$, $b \in \mathrm{R}^m$

   (a) 伸缩、平移、投影

2. 若 f 是仿射变换, $f : \mathrm{R}^n \to \mathrm{R}^m \quad f(S) = \{f(x) | x \in S\}$

   (a) 若 S 为凸集, 则 $f(x)$ 为凸集;

   (b) 若 $f(S)$ 为凸集, 则 S 为凸集.

3. 两个凸集的和为凸集

$$S_1 + S_2 = \{x + y | x \in S_1, y \in S_2\} \tag{2.4}$$

4. 两个凸集的笛卡尔积 (直积) 为凸集

$$S_1 \times S_2 = \{(x_1, x_2) | x \in S_1, x_2 \in S_2\} \tag{2.5}$$

5. 两个集合的部分和为凸集 (分配率)

$$S = \{(x, y_1 + y_2) | (x, y_1) \in S_1, (x, y_2) \in S_2\} \tag{2.6}$$

2.3 perspective collineation

2.3.1 concept

透视函数对向量进行伸缩 (规范化), 使得最后一维的分量为 1 并舍弃之.

$$P : \mathrm{R}^{n+1} \to \mathrm{R}^n, \ P(z,t) = z/t \tag{2.7}$$

2.3.2 direct significance

小孔成像

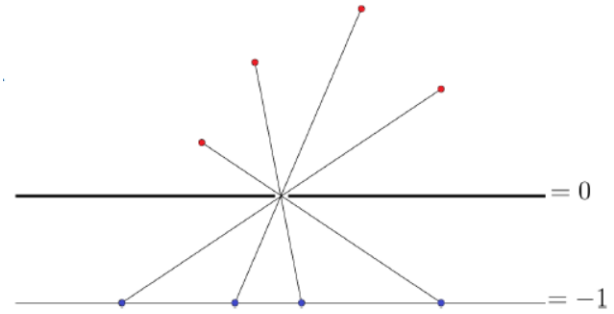


Fig 2.2: significance of perspective collineation

2.3.3 summary

凸集的透视变换仍然是凸集

2.4 segmentation plane

2.4.1 concept

设 C 和 D 为两不相交的凸集, 则存在超平面 P, P 可以将 C 和 D 分离.

$$\forall x \in C, a^T x \leq b \text{ 且 } \forall x \in D, a^T x \geq b \tag{2.8}$$

2.4.2 segmentation plane

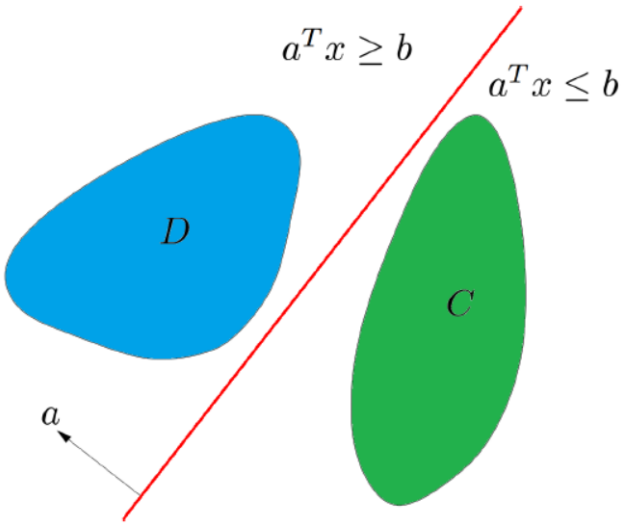


Fig 2.3: segmentation plane

2.4.3 the structure of sefmentation plane

1. 两个集合的距离, 定义为两个集合间元素的最短距离
2. 做集合 C 和集合 D 最短线段的垂直平分线
3. picture

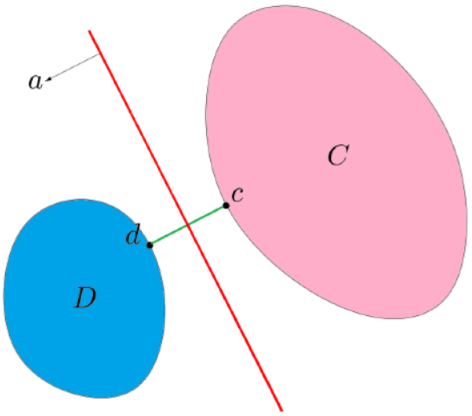


Fig 2.4: the distance of segmentation plane

2.4.4 support hyperplane

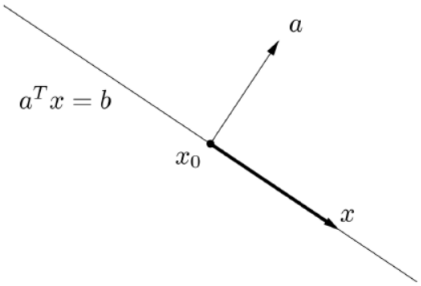


Fig 2.5: hyperplane

1. 设集合 C , x_0 为 C 边界上的点. 若存在 $a \neq 0$, 满足对任意 $x \in C$, 都有 $a^T x \leq a^T x_0$ 成立, 则称超平面 $\{x | a^T x = a^T x_0\}$ 为集合 C 在点 x_0 处的支撑超平面.

2. 凸集边界上任意一点, 均存在支撑超平面

3. 反之, 若在一个闭的非中空 (内部点不为空) 集合, 在边界上的任意一点存在支撑超平面, 则该集合为凸集

2.4.5 question

1. 如何定义两个集合的“最优”分割超平面?

   (a) 找到集合“边界”上的若干点, 以这些点为“基础”计算超平面的方向; 以两个集合边界上的这些点的平均作为超平面的“截距”

   (b) 支持向量: support vector

   (c) optimal segmentation plane

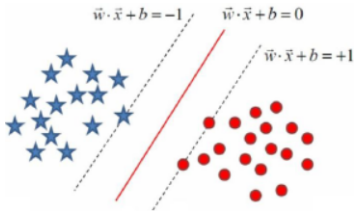


Fig 2.6: optimal segmentation plane

2. 若两个集合有部分相交, 如何定义超平面, 使得两个集合“尽量”分开?

   (a) 注: 上述“集合”不一定是凸集, 可能是由若干离散点组成. 若一组集合为 $(x, 1)$, 另一组集合为 $(x, 2)$, 则为机器学习中的分类问题.

2.5 property of convex function

2.5.1 concept

若函数 $fx(x)$ 的定义域 domf 为凸集, 且满足

$$\forall x, y \in \text{dom } f,\ 0 \leq \theta \leq 1,\ \text{有} \tag{2.9}$$

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y) \tag{2.10}$$

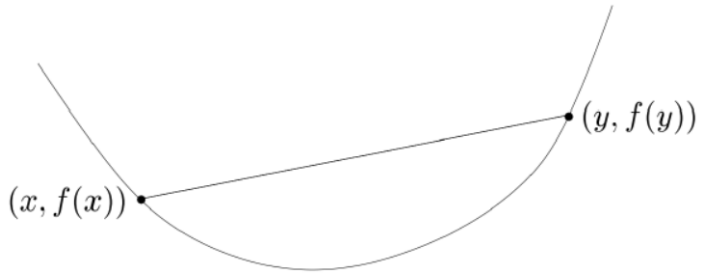


Fig 2.7: convex function

2.5.2 first-order derivative of convex function

若 $f(x)$ 一阶可微, 则函数 f 为凸函数当前仅当 f 的定义域 domf 为凸集, 且

$$\forall x, y \in \text{dom } f,\ f(y) \geq f(x) + \nabla f(x)^T (y - x) \tag{2.11}$$

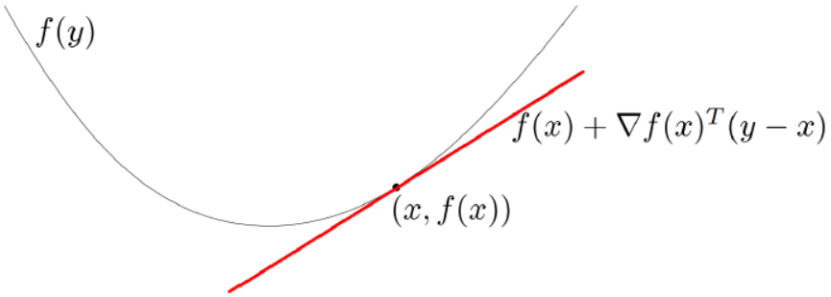


Fig 2.8: first-order derivative of convex function

对于凸函数, 其一阶 Taylor 近似本质上是该函数的全局下估计

2.5.3 second-order derivative of convex function

1. 若函数 f 二阶可微, 则函数 f 为凸函数当前仅当 dom 为凸集, 且

$$\nabla^2 f(x) \succ= 0 \tag{2.12}$$

2. 若 f 是一元函数, 上式表示二阶导大于等于 0

3. 若 f 是多元函数, 上式表示二阶导 Hessian 矩阵半正定

2.6 examples of convex function

- 指数函数 $f(x) = e^{ax}$
- 幂函数 $f(x) = x^a, x \in R^+, a \geq 1$ 或 $a \leq 0$
- 负对数函数 $f(x) = -\ln x$
- 负熵函数 $f(x) = x \ln x$
- 范数函数 $f(\vec{x}) = \| x \|$
- 最大值函数 $f(\vec{x}) = \max(x_1, x_2, ..., x_n)$
- 指数线性函数 $f(\vec{x}) = \log(e^{x_1} + e^{x_2} + ... + e^{x_n})$

第三章 Convex Optimization

3.1 optimization problem

1. common format

$$\begin{aligned} \text{minimize } & f_0(x), x \in \mathrm{R}^n \\ \text{subject to } & g_i(x) \leq 0,\ i = 1, ..., m \\ & h_j(x) = 0,\ j = 1, ..., p \\ \text{优化变量} \quad & x \in \boldsymbol{R}^n \\ \text{不等式约束} \quad & g_i(x) \leq 0 \\ \text{等式约束} \quad & h_j(x) = 0 \\ \text{无约束优化} \quad & m = p = 0 \end{aligned} \tag{3.1}$$

2. domain of optimal problem

$$D = \bigcap_{i=0}^{m} \mathrm{dom} g_i \cap \bigcup_{j=1}^{p} \mathrm{dom} h_j \tag{3.2}$$

3. feasible dot (solution)

   - $x \in D$, 满足式 3.1 中约束条件

4. feasible domain (feasible collection)

   - 所有可行点的集合

5. optimization value

$$p^* = \inf\{f_0(x) | g_i(x) \leq 0, i = 1, ..., m, h_j(x) = 0, j = 1, ..., p\} \tag{3.3}$$

6. optimization solution

$$p^* = f_0(x^*) \tag{3.4}$$

3.2 convex optimization

3.2.1 basic form

$$\begin{aligned} \text{minimize} \quad & f_0(x), x \in \mathrm{R}^n \\ \text{subject to} \quad & g_i(x) \leq 0, i = 1, ..., m \\ & h_j(x) = 0, j = 1, ..., p \end{aligned} \tag{3.5}$$

1. among, $g_i(x)$ 为凸函数, $h_j(x)$ 为仿射函数
2. important property of convex optimization problem
   (a) 凸优化问题的可行域为凸集
   (b) 凸优化问题的局部最优解即为 全局最优解

3.2.2 Lagrange 乘子法

在支持向量机模型 (SVM) 的推导中一步很关键的就是利用 Lagrange 对偶性将原问题转化为对偶问题.

1. theoty:
   一般的求极值问题, 求导等于 0. 但是如果不但要求极值, 还要求一个满足一定约束条件的极值, 那么此时就可以构造 Lagrange 函数, 其实就是 把约束项添加到原函数上, 然后对构造的新函数求导
2. example picture

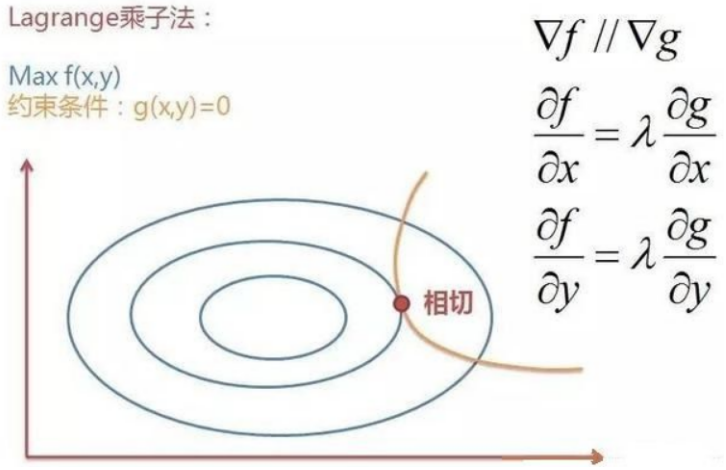


Fig 3.1: Lagrange multiplier method

3. analysis:
   对于一个要求极值的函数 $f(x,y)$, 图上的蓝圈就是这个函数的等高图, 就是说 $f(x,y) = c_1, c_2, ..., c_n$ 分别代表不同的数值 (每个值代表一圈，等高图), 我要找到一组 (x,y), 使它的 c_i 值越大越好, 但是这点必须满足约束条件 $g(x,y)$ (在黄线上)

4. conclusion:
   就是说 $f(x,y)$ 和 $g(x,y)$ 相切, 或者说它们的梯度 ∇f 和 ∇g 平行, 因此它们的梯度 (偏导) 成倍关系; 那我们就假设为 λ 倍, 然后把约束条件加到原函数后再对它求导, 其实就等于满足了图上的式子了.

3.2.3 Lagrange 函数的极小极大最优值 $\min_x \max_{\lambda,\nu:\lambda_i \geq 0} L(x,\lambda,\nu)$ 和原问题的最优值 p^*

1. 一般优化问题的 Lagrange 乘子法

$$\begin{aligned} \text{minimize} \quad & f_0(x), x \in \mathrm{R}^n \\ \text{subject to} \quad & g_i(x) \leq 0, i = 1, ..., m \\ & h_j(x) = 0, j = 1, ..., p \end{aligned} \tag{3.6}$$

2. Lagrange 函数

$$L(x,\lambda,\nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{p} \nu_j h_j(x) \tag{3.7}$$

   对固定的 x, Lagrange 函数 $L(x,\lambda,\nu)$ 为关于 λ 和 ν 的仿射函数, 这里, $x = (x^1, x^2, ..., x^n)^T \in \mathrm{R}^n, \lambda, \nu$ 是 Lagrange 乘子, $\lambda \geq 0$

3. Lagrange 函数的极小极大最优值 $\min_x \max_{\lambda,\nu:\lambda_i \geq 0} L(x,\lambda,\nu)$ 和原问题的最优值 p^* 之间的关系

   考虑 x 的函数:

$$f(x) = \max_{\lambda,\nu:\lambda \geq 0} L(x,\lambda,\nu) \tag{3.8}$$

   即原问题 $f(x)$ 与 Lagrange 函数在乘子变量最大化时的值是等价的

4. 证明上式: 假设给定某个 x，如果 x 违反原始问题的约束条件, 即存在某个 i 使得 $g_i(x) > 0$ 或者存在某个 j 使得 $h_j(x) \neq 0$, 那么就有

$$\phi_P(x) = \max_{\lambda,\nu:\lambda \geq 0} \left[f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right] = +\infty \tag{3.9}$$

   这里, 下标 P 表示原始问题

5. 因为若某个 i 使约束 $g_i(x) > 0$, 则可令 $\lambda_i \to +\infty$, 若某个 j 使 $h_j(x) \neq 0$, 则可令 ν_i 使 $\nu_j h_j(x) \to +\infty$, 而将其余各 λ_i, ν_j 均取为 0

   如果 x 满足约束条件, 则由式 3.7 和式 3.8 可知, $\phi_P(x) = f(x)$

6. summary:

$$\phi_P(x) = \begin{cases} f(x), & x \text{ 满足原始问题约束} \\ +\infty, & \text{其他} \end{cases} \tag{3.10}$$

   所以如果考虑极小化问题

$$\min_x \phi_P(x) = \min_x \max_{\lambda,\nu:\lambda\geq 0} L(x, \lambda, \nu) \tag{3.11}$$

   它是与原始最优化问题 3.1 等价的, 即它们有相同的解

   此时将原始最优化问题表示为广义 Lagrange 函数的极小极大问题.

   为了方便, 定义原始问题的最优值

$$p^* = \min_x \phi_P(x) = \min f(x) \tag{3.12}$$

   称为原始问题的值

3.2.4 Lagrange 对偶函数 (dual function)

1. Lagrange 对偶函数, 注意此处 下式中 x 需不需要改成 $\widetilde{x}$

$$\Gamma(\lambda, \nu) = \inf_{x\in D} L(x, \lambda, \nu) = \inf_{x\in D}(f_0(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)) \tag{3.13}$$

2. 若 $\widetilde{x} \in \mathbb{D}$ 为主问题 3.1 可行域中的点, 则对任意 ν 和 $\lambda \succeq 0$ 都有

$$\sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{i=1}^{n} \nu_j h_j(x) \leq 0 \tag{3.14}$$

   进而有

$$\Gamma(\lambda, \nu) = \inf_{x\in D} L(x, \lambda, \nu) \leq \inf L(\widetilde{x}, \lambda, \nu) \leq f(\widetilde{x}) \tag{3.15}$$

3. 若没有下确界, 定义 主问题下界:

$$\Gamma(\lambda, \nu) = -\infty \tag{3.16}$$

4. 根据定义, 显然有: 对 $\forall \lambda > 0$, $\forall \nu$, 若原优化问题有最优值 p^*, 则

$$\Gamma(\lambda, \nu) \leq p^* \tag{3.17}$$

即对偶函数给出了主问题的最优值的下界, 显然, 这个下界取决于 λ 和 ν 的值. 于是, 一个很自然的问题是: 基于对偶函数能获得的最好下界是什么?

5. 上述引出了 对偶优化问题

$$\max \Gamma(\lambda, \nu) \quad \text{s.t. } \lambda \succeq 0 \tag{3.18}$$

式 3.18 就是主问题 3.1 的对偶问题, 其中 λ 和 ν 称为“对偶变量”(dual variable)

6. 进一步: 无论主问题的 3.1 的凸性如何, 对偶问题 Lagrange 对偶函数 3.18 始终是凹函数

7. 假设式 3.18 的最优值为 $d^* = \max\limits_{\lambda,\nu} \Gamma(\lambda, \nu)$, 也即主问题 3.1 中最优值 p^* 的下界

8. 显然有 $d^* = \max\limits_{\lambda,\nu} \Gamma(\lambda, \nu) \leq p^*$, 这称为“弱对偶性”(weak duality) 成立;

若 $d^* = \max\limits_{\lambda,\nu} \Gamma(\lambda, \nu) = p^*$, 则称为“强对偶性”(strong duality) 成立, 此时由对偶问题能获得主问题的最优下界.

9. 对于一般优化问题, 强对偶性通常不成立. 但是, 若主问题为凸优化问题, 如式 3.1 中 $f(x)$ 和 $g_j(x)$ 均为凸函数, $h_i(x)$ 为仿射函数, 且其可行域中至少有一点使不等式约束严格成立, 此时强对偶性成立. 注意: 在强对偶性成立时, 将 Lagrange 函数对原变量求导, 并令导数等于 0, 即可得到原变量与对偶变量的数值关系.

10. 上述分析, 若满足强对偶性的 KKT 条件, 此时:

$$\begin{aligned} d^* &= \max_{\lambda,\nu:\lambda_i \geq 0} \min_x L(x, \lambda, \nu) = \min_x \max_{\lambda,\nu:\lambda_i \geq 0} L(x, \lambda, \nu) \\ &= p^* = \min f(x) \quad \{x \text{ subject to 约束条件}\} \end{aligned} \tag{3.19}$$

11. prove:
若原始问题和对偶问题都有最优值, 则 $d^* = \max\limits_{\lambda,\nu:\lambda_i \geq 0} \min\limits_x L(x, \lambda, \nu) \leq \min\limits_x \max\limits_{\lambda,\nu:\lambda_i \geq 0} L(x, \lambda, \nu) = p^* = \min f(x) \quad \{x \text{ subject to 约束条件}\}$

process:

对任意的 λ, ν 和 x, 有

$$L(x, \lambda, \nu) \leq \max_{\lambda,\nu:\lambda\geq 0} L(x, \lambda, \nu) \tag{3.20}$$

$$\therefore \quad \Gamma_D(\lambda, \nu) = \min_x L(x, \lambda, \nu) \leq L(x, \lambda, \nu) \leq \max_{\lambda,\nu:\lambda\geq 0} L(x, \lambda, \nu) = \phi_P(x) \tag{3.21}$$

‘ 即:

$$\Gamma_D(\lambda, \nu) \leq \phi_P(x) \tag{3.22}$$

由于原始问题和对偶问题均有最优值

$$\therefore \quad \max_{\lambda,\nu:\lambda\geq 0} \Gamma_D(\lambda, \nu) \leq \min_x \phi_P(x) \tag{3.23}$$

即:

$$\begin{aligned} d^* &= \max_{\lambda,\nu:\lambda_i\geq 0} \min_x L(x, \lambda, \nu) \leq \min_x \max_{\lambda,\nu:\lambda_i\geq 0} L(x, \lambda, \nu) \\ &= p^* = \min f(x) \quad \{x \text{ subject to 约束条件}\} \end{aligned} \tag{3.24}$$

12. KKT 条件:

$$g_i(x^*) \leq 0, \quad i = 1, 2, .., k \tag{3.25}$$

$$\lambda_i^* \geq 0, \quad i = 1, 2, ..., k \tag{3.26}$$

$$\lambda_i^* g_i(x^*) = 0, \quad i = 1, 2, ..., k \tag{3.27}$$

$$h_j(x^*) = 0, \quad j = 1, 2, ..., l \tag{3.28}$$

$$\nabla L(x^*, \lambda^*, \nu^*) = \nabla f(x^*) + \sum_{i=1}^{k} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{l} \nu_i^* \nabla h_j(x^*) = 0 \tag{3.29}$$

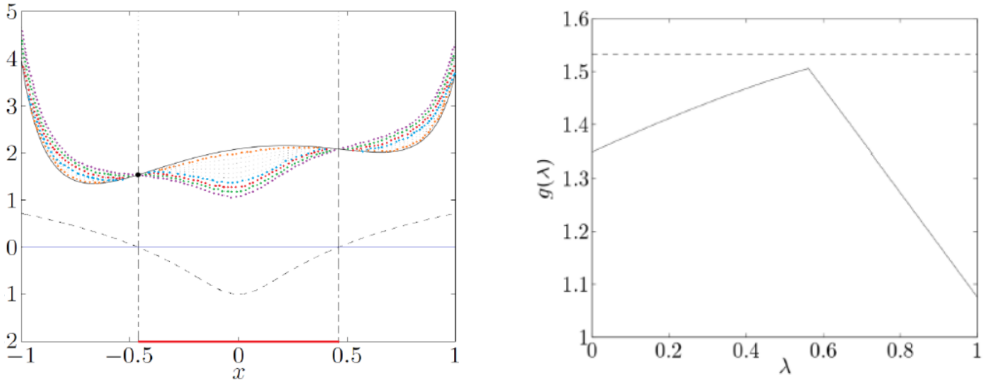
上式中 3.29 等价于

$$\begin{cases} \nabla_x L(x^*, \lambda^*, \nu^*) = 0 \\ \nabla_\lambda L(x^*, \lambda^*, \nu^*) = 0 \\ \nabla_\nu L(x^*, \lambda^*, \nu^*) = 0 \end{cases} \tag{3.30}$$

特别指出, 式 3.27 称为 KKT 的对偶互补条件. 由此条件可知: 若 $\lambda_i^* > 0$, 则 $g_i(x^*) = 0$

上式中, x^* 是原始变量的解, λ^*, ν^* 是对偶变量的解, 并且 $p^* = d^* = L(x^*, \lambda^*, \nu^*)$

13. 左侧是原函数, 右侧为对偶函数



(a) Original function (b) Dual function

Fig 3.2: dual problem

SVM 里的 w,b 看做是 Lagrange 函数里的变量 x

第四章　　SVM

4.1　jiange