机器学习笔记

Notes on Machine Learning

J.R.Tsien
jade.ray.tsien@gmail.com

目　　录

第1章　绪论

§ 1.1 凸函数

定义 1.1 (凸组合) 设$x_1, x_2, \cdots, x_m \in \mathbb{R}^n$，称$x$是$x_1, x_2, \cdots, x_m$的一个凸组合，如果存在满足$\sum_{i=1}^m \lambda_i = 1$的非负的$\lambda_1, \lambda_2, \cdots, \lambda_m$使得

$$x = \sum_{i=1}^m \lambda_i x_i$$

定义 1.2 (凸集) 集合$S \subset \mathbb{R}^n$是凸集的充要条件是S中任意若干点的任一凸组合仍属于S。

定义 1.3 (凸函数) 设$S \subset \mathbb{R}^n$是非空凸集，f是定义在S上的函数。称f是定义在S上的凸函数，如果对$\forall x_1, x_2 \in S, \lambda \in (0,1)$，都有

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

称f为S上的严格凸函数，如果当$x_1 \neq x_2$时，上式中的不等号严格成立。

在二维平面中，凸函数是那种在函数上方所形成的点集是凸集的函数。事实上，凸函数更广泛的定义不一定是可微的。同时，所有的线性函数（linear）和仿射函数（affine）都是凸函数。仿射函数是定义为$f(x) = A^T x + B$的函数，跟线性函数一样，只不过多了个截距项B。

定理 1.1 (凸函数的充要条件) 设S为非空开凸集，$f(x)$是S上可微函数，则$f(x)$是凸函数的充要条件是，对$\forall x^* \in S$，都有

$$f(x) \geq f(x^*) + \nabla f(x^*)^T (x - x^*), \forall x \in S$$

类似的，$f(x)$是严格凸函数的充要条件是

$$f(x) > f(x^*) + \nabla f(x^*)^T (x - x^*), \forall x^* \neq x \in S$$

§ 1.2 最优化问题

1.2.1 原始问题

假设$f(x), c_i(x), h_j(x)$是定义在$\mathbb{R}^n$上的连续可微函数。则最优化问题

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{C1}$$

$$\text{s.t.} \quad c_i(x) \le 0, \quad i = 1, 2, \cdots, k \tag{C2}$$

$$h_j(x) = 0, \quad j = 1, 2, \cdots, l \tag{C3}$$

称为原始问题。

定义广义拉格朗日函数（generalized Lagrange function）

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

$\alpha_i \ge 0, \beta_j$称为拉格朗日乘子（Lagrange multipliers）。考虑等式

$$\theta_P(x) = \max_{\alpha,\beta;\alpha_i\ge 0} L(x, \alpha, \beta)$$

下标P表示原始（primal）问题。假设某个x不满足原始的约束条件C2或者C3，也即存在某个i，使得$c_i(x) > 0$或者$h_i(x) \ne 0$，那么总能找到一个α_i或者β_i，使得

$$\theta_P(x) = \max_{\alpha,\beta;\alpha_i\ge 0} L(x, \alpha, \beta) = \infty$$

相反的，如果约束条件满足，那么$\theta_P(x) = f(x)$，也即

$$\theta_P(x) = \begin{cases} f(x) & x\text{满足约束条件} \\ \infty & \text{否则} \end{cases}$$

由此可以看出，$\theta_P(x)$在原始问题约束条件满足时与优化目标$f(x)$有相同的值，而当约束条件不满足时，$\theta_P(x)$的值是∞，所以有下列等式

$$\min_x \theta_P(x) = \min_x \max_{\alpha,\beta;\alpha_i\ge 0} L(x, \alpha, \beta)$$

定义原始问题的最优解

$$p^\star = \min_x \theta_P(x)$$

1.2.2 拉格朗日对偶问题

首先定义

$$\theta_D(\alpha,\beta)=\min_x L(x,\alpha,\beta)$$

D表示对偶（dual）问题。θ_D其实是$L(x,\alpha,\beta)$关于x的最小值，而这个最小值是以α,β为参数的。对这个最小值进行极大化，得到对偶问题

$$\max_{\alpha,\beta;\alpha_i\geq 0}\theta_D(\alpha,\beta)=\max_{\alpha,\beta;\alpha_i\geq 0}\min_x L(x,\alpha,\beta)$$

对偶问题跟原始问题在形式上一样，除了max和min的顺序不同。

定义对偶问题的解

$$d^\star=\max_{\alpha,\beta;\alpha_i\geq 0}\theta_D(\alpha,\beta)$$

1.2.3 原始问题与对偶问题的关系及KKT条件

容易证明，有

$$d^\star=\max_{\alpha,\beta;\alpha_i\geq 0}\min_x L(x,\alpha,\beta)\leq\min_x\max_{\alpha,\beta;\alpha_i\geq 0}L(x,\alpha,\beta)=p^\star$$

事实上，函数的max min永远不大于函数的min max。

下面描述原始问题与对偶问题解相同的充要条件，也称为Karush-Kuhn-Tucker（KKT）条件。设x^*是原始问题的解，α^*,β^*是对偶问题的解，假设$f(x),c_i(x)$是凸函数，$h_j(x)$是仿射函数，并且不等式约束$c_i(x)$严格可行，即$\exists x(\forall i(c_i(x)<0))$，则原始问题与对偶问题解相同（即$p^\star=d^\star=L(x^*,\alpha^*,\beta^*)$）的充要条件是

$$\nabla_x L(x^*,\alpha^*,\beta^*)=0 \tag{KKT1}$$

$$\nabla_\alpha L(x^*,\alpha^*,\beta^*)=0 \tag{KKT2}$$

$$\nabla_\beta L(x^*,\alpha^*,\beta^*)=0 \tag{KKT3}$$

$$\alpha_i^* c_i(x^*)=0,i=1,2,\cdots,k \tag{KKT4}$$

$$c_i(x^*)\leq 0,i=1,2,\cdots,k \tag{KKT5}$$

$$\alpha_i^*\geq 0,i=1,2,\cdots,k \tag{KKT6}$$

$$h_j(x^*)=0,j=1,2,\cdots,l \tag{KKT7}$$

条件KKT5称为对偶互补条件（dual complementarity condition）。由此条件可知，当$c_i(x^*)<$

0时，恒有$\alpha_i^* = 0$。也即，对于与分离超平面的函数间隔不为1的向量点而言，其拉格朗日乘子为0。当$\alpha_i^* > 0$时，$c_i(x^*) = 0$。在SVM 模型中，此时向量x^* 与分离超平面的函数间隔为1，这种向量称之为支持向量（support vector），因为只有这种向量对模型的训练是有价值的。

§ 1.3 梯度下降法（gradient descent）

梯度下降法或者最速下降法（steepest descent）是求解无约束最优化问题的方法。特点是实现起来比较简单。其原理是如果函数$f(x)$在点a处可微且有定义，那么函数$f(x)$在a点沿着梯度的反方向，即$-\nabla f(a)$，下降最快。

所以，可以从一个初始值x_0出发，沿梯度反方向迭代的更新解。如下

$$x_{n+1} = x_n - \alpha \nabla f(x_n)$$

直到x_n的值不再发生变化，或者变化很小，此时，x_n等于或者接近$f(x)$的极小值。α称为学习率（learning rate）。α值过大，可能会在最小值附近振荡。α值过小，可能学习的时间比较长。同时，α值的选取可以是预先设定的固定值，也可以是根据解更新的情况变化的值。

梯度下降法的一个问题在于，能否得到最优解取决于初始值的选取。

§ 1.4 牛顿法和拟牛顿法

1.4.1 牛顿法（Newton Method）

牛顿法，或牛顿-拉夫逊法（Newton-Raphson Method）也是求解无约束优化问题的常用方法。牛顿法是二阶收敛的算法（不仅考虑梯度方向，同时考虑梯度的梯度），而梯度下降法是一阶收敛的，因此牛顿法的收敛速度比梯度下降快。换句话说，牛顿法用二次曲面来拟合当前所在位置的局部曲面，然后按照曲率最大的方向下降。而梯度法是用一个平面去拟合局部曲面，然后按照平面的法向量的方向下降。但是一次迭代的代价比较高，因为需要计算矩阵的逆。

考虑无约束的最优化问题

$$\min_{x \in \mathbb{R}^n} f(x)$$

假设$f(x)$有二阶连续偏导数，且设第k次迭代的解为x_k，将$f(x)$在点x_k处进行二阶泰勒展开

$$f(x) = f(x_k) + (x - x_k) f'(x_k) + \frac{1}{2}(x - x_k)^2 f''(x_k)$$

函数$f(x)$在下次迭代点x_{k+1}处取得极值的必要条件是$f'(x_{k+1})=0$，即

$$f'(x)|_{x_{k+1}} = f'(x_k) + (x_{k+1} - x_k)f''(x_k) = 0$$

解上式得到

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

迭代停止的条件可以设定为$f'(x_k) < \epsilon$。

当x是向量的时候，其一阶导数要修改成梯度的形式，二阶导数修改成其Hessian矩阵，即

$$f(x) = f(x_k) + (x - x_k)\nabla_x f(x_k) + \frac{1}{2}(x - x_k)^T H(x_k)(x - x_k)$$

当$H(x_k)$是正定矩阵时，$f(x)$的极值为极小值。其更新公式是

$$x_{k+1} = x_k - H^{-1}(x_k)\nabla_x f(x_k)$$

迭代终止的条件$\nabla_x f(x_k) < \epsilon$.

当初始点离极值点较远时，牛顿法可能不收敛，因为此时由Hessian矩阵的逆矩阵和梯度规定的牛顿方向不一定是下降方向。

1.4.2 拟牛顿法（Quasi Newton Method）

牛顿法中Hessian矩阵求逆比较复杂，所以实际中会考虑用一个近似的正定对称矩阵来代替Hessian矩阵，这就是拟牛顿法。不同的替代方法形成了不同的拟牛顿法。

1.4.2.1 拟牛顿条件

首先将函数$f(x)$在第k次迭代点处泰勒展开

$$f(x) = f(x_k) + (x - x_k)\nabla_x f(x_k) + \frac{1}{2}(x - x_k)^T H(x_k)(x - x_k)$$

求其此时的梯度

$$\nabla f(x) = \nabla f(x_k) + (x - x_k)H(x_k)$$

令$x = x_{k+1}$，

$$\nabla f(x_{k+1}) - \nabla f(x_k) = (x_{k+1} - x_k)H(x_k)$$

左边是梯度的变化，右边$(x_{k+1} - xk)$是自变量的变化，分别记为

$$g_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

$$\delta_k = x_{k+1} - x_k$$

得到拟牛顿条件

$$g_k = H_k \delta_k \tag{1.1}$$

或

$$H_k^{-1} g_k = \delta_k \tag{1.2}$$

拟牛顿法选择$G_k \approx H_k^{-1}$，或者$B_k \approx H_k$。而且，尽量使得近似矩阵的更新方式为迭代更新

$$G_{k+1} = G_k + \Delta G_k$$

且G_k满足拟牛顿条件

$$G_{k+1} g_k = \delta_k \tag{1.3}$$

一般令$G_0 = I$为单位阵，所以只需要找到ΔG_k即可。

1.4.2.2 DFP算法

DFP算法最早由William C. Davidon于1959年提出，后由Roger Fletcher和Michael J.D. Powell发展和完善。DFP算法令校正矩阵为

$$\Delta G_k = \frac{\delta_k \delta_k^T}{\delta_k^T g_k} - \frac{G_k g_k g_k^T G_k}{g_k^T G_k g_k}$$

构造过程主要的规则在于保证如果初始矩阵G_0正定，那么每个G_k都是正定。

DFP算法

输入：目标函数$f(x)$，梯度$\nabla f(x)$，精度要求ϵ

输出：$f(x)$的极小值点$x^\star$

（1）选定初始点x_0，取G_0为正定对称矩阵（单位阵），置$k = 0$

（2）计算$\nabla f(x_k)$，若$\|\nabla f(x_k)\| < \epsilon$，令$x^\star = x_k$，停止计算

（3）令$p_k = -G_k \nabla f(x_k)$

（4）一维搜索：求λ_k使得

$$f(x_k + \lambda_k p_k) = \min_{\lambda \geq 0} f(x_k + \lambda p_k)$$

（5）令$x_{k+1} = x_k + \lambda_k p_k$

（6）计算$\nabla f(x_{k+1})$，若$\|\nabla f(x_{k+1})\| < \epsilon$，令$x^\star = x_{k+1}$，停止计算；否则，按$G_{k+1} = G_k + \Delta G_k$计算$G_{k+1}$

（7）置$k = k + 1$，转（3）

1.4.2.3 BFGS算法

BFGS算法（Broyden-Fletcher-Goldfarb-Shanno）是最流行的拟牛顿法。DFP用一个正定矩阵来近似H^{-1}，BFGS用一个正定矩阵来近似H，其表示如下

$$B_{k+1} = B_k + \Delta B_k \tag{1.4}$$

$$\Delta B_k = \frac{g_k g_k^T}{g_k^T \delta_k} - \frac{B_k \delta_k \delta_k^T B_k}{\delta_k^T B_k \delta_k} \tag{1.5}$$

构造的核心也是满足如果初始矩阵B_0是正定的，那么迭代过程中的每一个B_k都是正定的。

BFGS算法

输入：目标函数$f(x)$，梯度$\nabla f(x)$，精度ϵ

输出：$f(x)$的极小值点$x^\star$

（1）选定初始点x_0，取B_0为正定对称矩阵（单位阵），置$k=0$

（2）计算$\nabla f(x_k)$，若$\|\nabla f(x_k)\| < \epsilon$，令$x^\star = x_k$，停止计算

（3）由$B_k p_k = -\nabla f(x_k)$求出p_k

（4）一维搜索：求λ_k使得

$$f(x_k + \lambda_k p_k) = \min_{\lambda \geq 0} f(x_k + \lambda p_k)$$

（5）令$x_{k+1} = x_k + \lambda_k p_k$

（6）计算$\nabla f(x_{k+1})$，若$\|\nabla f(x_{k+1})\| < \epsilon$，令$x^\star = x_{k+1}$，停止计算；否则，按$B_{k+1} = B_k + \Delta B_k$计算$B_{k+1}$

（7）置$k = k+1$，转（3）

令$G_k = B_k^{-1}$，对$B_{k+1} = B_k + \Delta B_k$运用两次Sherman-Morrison公式有

$$G_{k+1} = (I - \frac{\delta_k g_k^T}{\delta_k^T g_k}) G_k (I - \frac{\delta_k g_k^T}{\delta_k^T g_k})^T + \frac{\delta_k g_k^T}{\delta_k^T g_k}$$

称为BFGS算法关于G_k的迭代公式。

Sherman-Morrison公式：假设A是n阶可逆矩阵，u，v是n维向量，且$A + uv^T$也是可逆矩阵，则

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1} u v^T A^{-1}}{1 + v^T A^{-1} u}$$

记DFP关于G_k的迭代公式得到的G_{k+1}为G^{DFP}，由BFGS得到的记为G^{BFGS}，它们都满足拟牛顿公式，且其线性组合

$$G_{k+1} = \alpha G^{DFP} + (1-\alpha) G^{BFGS}$$

也满足拟牛顿条件，而且是正定是。其中$0 \leq \alpha \leq 1$。这样得到的一类拟牛顿法称为Broyden类算法。

实际在使用的时候使用的是LBFGS算法（Limited-memory-BFGS）。有很成熟的开源实现，直接调用即可。

§ 1.5 机器学习概论

1.5.1 生成模型与判别模型

判别模型（discriminative model）是给定输入X，要求预测的输出Y的类型。其方法是通过训练数据来学习决策函数$f(x)$或者条件概率分布$p(y|x)$来构造预测的模型。

生成模型（generative model）是通过学习联合分布$p(x,y)$或者$p(x|y)$以及$p(y)$来得到给定x的y的后验概率$p(y|x) = \frac{p(x,y)}{p(y)} = \frac{p(x|y)p(y)}{p(y)}$。

第2章　回归

给定数据集

$$T = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\},$$

其中，$x^{(i)} = (x_0, x_1, \cdots, x_n) \in \mathcal{X} = \mathbb{R}^{n+1}$，$y^{(i)} \in \mathcal{Y}$, $i = 0, 2, \cdots, m$，且$x_0 = 1$（表示截距，intercept term）。回归分析的任务是找出输入x与输出y之间的关系。

§ 2.1 线性回归

假设输入与输出之间满足的关系是线性的，$theta$称为参数（parameters）或者权重（weights）

$$y = h_\theta(x) = \theta^T x = \sum_{i=0}^{n} \theta_i x_i, \qquad \theta \in \mathbb{R}^{n+1}$$

对于这个模型，需要有一个损失函数（cost function）来表示其对训练数据的拟合程度

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2.$$

这个被称为最小二乘方效用函数（least-square cost function）。

则这个问题的求解可以表述为如下的无约束最优化问题

$$\min_\theta J(\theta)$$

2.1.1 直接求解

直接求$J(\theta)$对θ的极值。首先定义设计矩阵（design matrix）X

$$X = \begin{bmatrix} (x^{(1)})^T & (x^{(2)})^T & \cdots & (x^{(m)})^T \end{bmatrix}^T$$

$$Y = \begin{bmatrix} y^{(1)} & y^{(2)} & \cdots & y^{(m)} \end{bmatrix}^T$$

则有

$$X\theta - Y = \begin{bmatrix} h_\theta(x^{(1)}) - y^{(1)} & \cdots & h_\theta(x^{(m)}) - y^{(m)} \end{bmatrix}^T$$

考虑到$z^T z = \sum_i z_i^2$，有

$$J(\theta) = \frac{1}{2}(X\theta - Y)^T(X\theta - Y) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

使用公式$\nabla_A trABA^TC = CAB + C^TAB^T$，且由于$J(\theta)$只是个实数，所以

$$\begin{aligned} \nabla_\theta J(\theta) &= \nabla_\theta trJ(\theta) \\ &= X^TX\theta - X^TY \end{aligned}$$

令$\nabla_\theta J(\theta) = 0$得到

$$\theta = (X^TX)^{-1}X^TY$$

不过，这个公式用来直接计算θ不现实，因为矩阵求逆比较麻烦，同时可能会是数值不稳定的矩阵。

2.1.2 牛顿法（Newton's Method）

已知最优化问题

$$\min_\theta J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

函数$J(\theta)$的Hessian矩阵是（注意$x^{(i)}$和θ都是列向量）

$$H(\theta) = \begin{bmatrix} \frac{\partial^2 J(\theta)}{\partial\theta_1\partial\theta_1} & \frac{\partial^2 J(\theta)}{\partial\theta_1\partial\theta_2} & \cdots & \frac{\partial^2 J(\theta)}{\partial\theta_1\partial\theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 J(\theta)}{\partial\theta_n\partial\theta_1} & \frac{\partial^2 J(\theta)}{\partial\theta_n\partial\theta_2} & \cdots & \frac{\partial^2 J(\theta)}{\partial\theta_n\partial\theta_n} \end{bmatrix} = \sum_{i=1}^{m} x^{(i)}(x^{(i)})^T = X^TX$$

又，$J(\theta) = \frac{1}{2}(X\theta - Y)^T(X\theta - Y)$，所以其梯度

$$\nabla_\theta J(\theta) = X^TX\theta - X^TY$$

代入牛顿法的迭代公式

$$\theta_{n+1} = \theta_n - (X^TX)^{-1}(X^TX\theta_n - X^TY) = (X^TX)^{-1}X^TY$$

2.1.3 批处理梯度下降法（batch gradient descent）

$J(\theta)$对θ的梯度

$$\frac{\partial J(\theta)}{\partial \theta} = \sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x^{(i)}$$

所以，更新公式为

$$\theta_{n+1} = \theta_n - \alpha\frac{\partial J(\theta)}{\partial \theta}$$

$$= \theta_n - \alpha\sum_{i=1}^{m}(h_{\theta_n}(x^{(i)}) - y^{(i)})x^{(i)}$$

这个公式更新时每次都需要全部的训练数据集，所以称之为批处理梯度下降法。当数据集比较大时，进行一次更新就比较耗费时间。

2.1.4 随机梯度下降法（stochastic gradient descent）

随机梯度下降法，也称增量梯度下降（incremental gradient descent），一次使用一条训练数据来更新参数。算法如下

$$\theta := \theta - \alpha(h_\theta(x^{(i)}) - y^{(i)})x^{(i)}, i = 1, 2, \cdots, m$$

随机梯度下降法更新的速度比梯度下降法快，但可能收敛不到最优值，不过通过调节学习率可以使得算法得到较好的解。而且随机梯度下降法可以用作在线学习的算法。

§ 2.2 局部加权线性回归（LWR）

线性回归的方法是参数学习算法（parametric learning algorithm），其参数的个数，即特征的个数是固定的，一旦算法学习完成，训练数据集就不再对参数产生影响。但是，当选取的参数过多时，可能存在过拟合问题，而当选取的参数过少时，存在欠拟合问题。局部加权线性回归（local weighted linear regression，LWR）是一种非参学习算法（non-parametric learning algorithm），其参数是随着预测点的不同而发生变化的，每有一个新的预测点，就需要整个训练数据集重新参与学习。所谓局部，是因为目标函数的逼近仅仅根据查询点附近的数据。所谓加权，是因为每个训练样例的贡献都是由它与查询点间的距离加权的。而回归是指数值逼近的方法。

线性回归的优化目标是

$$\min_{\theta} \sum_{i=1}^{m} (\theta^T x^{(i)} - y^{(i)})^2$$

而LWR的优化目标是在上述公式上增加一个距离乘法项

$$\min_{\theta} \sum_{i=1}^{m} w^{(i)} (\theta^T x^{(i)} - y^{(i)})^2$$

一个相对标准的权重选择是

$$w^{(i)} = exp(-\frac{(x^{(i)} - x)^2}{2\tau^2})$$

其中，x是要预测的输入数据，τ称为带宽（bandwidth）。权重项使得离输入数据x越近的点影响越大。

非参数学习算法在局部预测能力上有时要比参数学习算法好，但是缺点是每次做预测都要重新学习，耗费时间空间。

§ 2.3 Logistic回归

2.3.1 二项逻辑斯谛回归

逻辑斯谛回归（logistic regression）是分类模型，跟线性回归模型有相同的输入，但是其输出是离散值。但是，之所以叫“回归”是因为它依然采用线性回归的算法来预测x与y之间的关系。也因此，需要限制模型中y值的输出，令

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

其中

$$g(z) = \frac{1}{e^{-z}}$$

称为逻辑斯谛函数（logistic function），或者S形曲线（sigmoid curve）。

为了推导方便，首先给出$g(z)$的导数

$$g'(z) = g(z)(1 - g(z))$$

设

$$P(y = 1|x; \theta) = h_\theta(x)$$

$$P(y = 0|x; \theta) = 1 - h_\theta(x)$$

或者简写成

$$p(y|x;\theta) = (h_\theta(x))^y(1-h_\theta(x))^{1-y}$$

其对数似然函数

$$l(\theta) = \log\prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta) = \sum_{i=1}^{m}(y^{(i)}\log h_\theta(x^{(i)}) + (1-y^{(i)})\log(1-h_\theta(x^{(i)})))$$

$$= \sum_{i=1}^{m}(y^{(i)}h_\theta(x^{(i)}) + \log(1-h_\theta(x^{(i)})))$$

其中，m是训练样本的个数。对θ求导，得

$$\frac{\partial l(\theta)}{\partial\theta} = \sum_{i=1}^{m}(y^{(i)} - h_\theta(x^{(i)}))x^{(i)}$$

则对于单条训练样本的更新公式是

$$\theta := \theta + \alpha(y^{(i)} - h_\theta(x^{(i)}))x^{(i)}$$

注意，之所以是+号，是因为所求的最优化问题是似然函数的最大化，也即按照梯度的方向增加。

这个更新公式跟LMS（least mean squares）更新准则一样。也即，更新的幅度由学习率以及特征向量和误差的乘积决定。这是由梯度下降法所决定的。

2.3.2 多项逻辑斯谛回归

逻辑斯谛回归其实是建立了逻辑斯谛概率模型之后依据似然函数最大化准备建立的回归模型，其学习算法一般是梯度下降或者牛顿法。多项逻辑斯谛回归的概率模型可以表述为：如果离散随机变量Y的取值集合是$\{1,2,\cdots,K\}$，那么多项逻辑斯谛回归模型是

$$P(Y=k|x;\theta) = \frac{e^{\theta^T x}}{1+\sum_{i=1}^{K-1}e^{\theta^T x}}, k=1,2,\cdots,K-1 \tag{2.1}$$

$$P(Y=K|x;\theta) = \frac{1}{1+\sum_{i=1}^{K-1}e^{\theta^T x}}, k=K \tag{2.2}$$

其中，$x,\theta\in\mathbb{R}^{n+1}$.

§ 2.4 广义线性模型

广义线性模型（Generalized Linear Models, GLMs）是线性回归（linear regression）和逻辑斯谛回归（logistic regression）的统一表述。

2.4.1 指数分布族（the exponential family）

首先定义指数分布族

$$p(y;\eta) = b(y)e^{\eta^T T(y)-a(\eta)} \tag{2.3}$$

其中，η称为分布的自然参数（natural parameter）或者标准参数（canonical parameter），η^T是取η的转置的意思，当η是标量（scalar）的时候，$\eta^T = \eta$，当η是向量（vector）的时候，上式就是向量的乘积，$T(y)$ 是充分统计量（sufficient statistic），通常令$T(y) = y$，$a(\eta)$称为对数分割函数（log partition function）。注意到$e^{-a(\eta)}$ 实际上起到的是归一化的作用，即保证分布$p(y;\eta)$对y的和或积分是1。固定T, a, b就得到了以η为参数的分布族。下面将伯努利分布和高斯分布表示成指数族的形式。

伯努利分布（Bernoulli distribution）的形式为

$$p(y;\phi) = \phi^y(1-\phi)^{1-y} \tag{2.4}$$

表示的是一次Bernoulli实验（二值）的结果。y表示实验成功与否：成功，则$y = 1$；否则，$y = 0$。

将上式表示成

$$\begin{aligned} p(y;\phi) &= \phi^y(1-\phi)^{1-y} \\ &= exp(y\log\phi + (1-y)\log(1-\phi)) \\ &= exp((\log(\frac{\phi}{1-\phi}))y + \log(1-\phi)) \end{aligned}$$

此时，令$\eta = \log\frac{\phi}{1-\phi}$就得到了Bernoulli分布的指数族形式。其中

$$T(y) = y$$

$$b(\eta) = 1$$

$$a(\eta) = -\log(1-\phi) = \log(1+e^{\eta})$$

注意到$\phi = \frac{1}{1+e^{-\eta}}$刚好是sigmoid函数。

高斯分布（Gaussian distribution）的一般形式

$$p(y;\mu,\sigma)=\frac{1}{\sqrt{2\pi}\sigma}exp(-\frac{(y-\mu)^2}{2\sigma^2}) \tag{2.5}$$

考虑到σ对假设函数没有影响，所以可以令$\sigma=1$

$$p(y;\mu)=\frac{1}{\sqrt{2\pi}}exp(-\frac{1}{2}(y-\mu)^2) \tag{2.6}$$

表示为GMLs形式，有

$$\eta=\mu$$

$$T(y)=y$$

$$a(\eta)=\frac{\mu^2}{2}=\frac{\eta^2}{2}$$

$$b(y)=\frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}$$

如果要考虑σ的话，需要用到GLMs的更为一般的模型

$$p(y;\eta,\tau)=b(a,\tau)exp(\frac{\eta^T T(y)-a(\eta)}{c(\tau)}) \tag{2.7}$$

其中，$\eta=(\mu,\sigma)\in\mathbb{R}^2$，$\tau$称为离差参数（dispersion parameter），对于高斯分布来说，$c(\tau)=\sigma^2$。

2.4.2 构造GLMs

构造GLMs有三个假设：

1.$y|x;\theta\sim ExponentialFamily(\eta)$，即给定特征向量$x$和参数$\theta$，输出$y$符合以$\eta$为参数的指数分布族。

2.给定x，目的是要求$T(y)$的期望值，即$h(x)=E[T(y)|x]$。$T(y)$是y的充分统计量，一般是$T(y)=y$。

3.自然参数η和输入x之间满足线性关系：$\eta=\theta^T x$。

2.4.2.1 最小二乘法（Ordinary Least Square）

考虑到目标变量（target variable）或者响应变量（response variable）y是连续的，假定其服从高斯分布，即$y|x;\theta\sim\mathcal{N}(\mu,\sigma^2)$。注意高斯分布的期望是$\mu$，标准差是$\sigma$。此时，$T(y)=y$，所以

$$h_\theta(x)=E[T(y)|x;\theta]=E[y|x;\theta]$$

$$= \mu$$

$$= \eta$$

$$= \theta^T x$$

第一行基于假设2，第二行基于高斯分布的特性，第三行基于假设1，即高斯分布作为指数分布族的一个特例，第四行基于假设3.

2.4.2.2 Logistic Regression

LogisticRegression的输出只有$y \in \{0, 1\}$两个值，所以考虑输出满足伯努利分布，即$y|x;\theta \sim Bernoulli(\phi)$，而伯努利分布的期望$E[y|x;\theta] = \phi$，所以有

$$h_\theta(x) = E[T(y)|x;\theta] = E[y|x;\theta]$$

$$= \phi$$

$$= \frac{1}{1+e^{-\eta}}$$

$$= \frac{1}{1+e^{-\theta^T x}}$$

这样就得到了logistic函数$\frac{1}{1+e^{-z}}$。

2.4.2.3 Softmax Regression

考虑一个多类分类问题，假设$y \in \{1, 2, \cdots, k\}$，此时可以用多项式分布对$y$进行建模。最终可以得到多元逻辑斯谛回归模型。

2.4.3 总结

从上面可以看出，广义线性模型（GLMs）的作用其实是提供一个统一的方法来实现训练模型的选择。其流程如下所示

$$y|x;\theta \sim GLMs \rightarrow \text{模型（logistic回归、softmax回归及最小二乘法等）}$$

$$\rightarrow \text{由likelihood函数建立最优化模型}$$

$$\rightarrow \text{参数求解（梯度法、牛顿法等）}$$

第3章　生成算法

§ 3.1 高斯判别分析（Gaussian Discriminant Analysis）

高斯判别法是生成模型，虽然名字中有“判别”两个字。

3.1.1 多元正态分布（Multivariate Normal Distribution）

假设$x \sim \mathcal{N}(\mu, \Sigma)$，即$x$服从以$\mu \in \mathbb{R}^n$为均值向量（mean vector），$\Sigma \in \mathbb{R}^{n \times n}$ 为协方差矩阵（covariance matrix）的多元高斯分布，则x的概率密度函数（PDF）为

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))$$

3.1.2 高斯判别分析（GDA）

生成模型对每一个分类分别进行建模，即首先做关于分类和不同分类的输入特征的分布的假设

$$y \sim Bernoulli(\phi)$$

$$x|y=0 \sim \mathcal{N}(\mu_0, \Sigma)$$

$$x|y=1 \sim \mathcal{N}(\mu_1, \Sigma)$$

由此，参数为$\phi, \Sigma, \mu_0, \mu_1$。注意两个高斯分布虽然有相同的均值向量，但是使用不同的协方差矩阵。

其对数似然函数为

$$\updownarrow(\phi, \mu_0, \mu_1, \Sigma) = \log \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma)$$

$$= \log \prod_{i=1}^{m} p(x^{(i)}|y^{(i)};\mu_0,\mu_1,\Sigma)p(y^{(i)};\phi)$$

解得参数的最大似然估计

$$\begin{aligned}
\phi &= \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)}=1\} \\
\mu_0 &= \frac{\sum_{i=1}^{m} 1\{y^{(i)}=0\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)}=0\}} \\
\mu_1 &= \frac{\sum_{i=1}^{m} 1\{y^{(i)}=1\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)}=1\}} \\
\Sigma &= \frac{1}{m}\sum_{i=1}^{m}(x^{(i)}-\mu_{y^{(i)}})(x^{(i)}-\mu_{y^{(i)}})^T.
\end{aligned}$$

3.1.3 高斯判别分析与logistic回归的对比

在高斯判别分析中，如果记$p(y=1|x;\phi,\mu_0,\mu_1,\Sigma)$为$x$的函数，则有如下形式

$$p(y=1|x;\phi,\mu_0,\mu_1,\Sigma) = \frac{1}{1+exp(-\theta^T x)}$$

θ是ϕ,Σ,μ_0,μ_1的函数。

事实上，

$$\left.\begin{array}{l} p(x|y=0)\sim\mathcal{N}(\mu_0,\Sigma) \\ p(x|y=1)\sim\mathcal{N}(\mu_1,\Sigma) \end{array}\right\} \Rightarrow p(y=1|x)\sim LogitsticRegression$$

反之，不成立；此外，还有

$$\left.\begin{array}{l} p(x|y=0)\sim \text{Poisson}(\lambda_0) \\ p(x|y=1)\sim \text{Poisson}(\lambda_1) \end{array}\right\} \Rightarrow p(y=1|x)\sim LogitsticRegression$$

反之，不成立；更一般的，有

$$\left.\begin{array}{l} p(x|y=0)\sim \text{ExponentialFamily}(\eta_0) \\ p(x|y=1)\sim \text{ExponentialFamily}(\eta_1) \end{array}\right\} \Rightarrow p(y=1|x)\sim LogitsticRegression$$

反之，不成立。

证明一下最后一个公式。显然有

$$p(y) = \phi^y(1-\phi)^{1-y}$$

$$p(x|y=0) = b(x)e^{\eta_0^T T(x) - a(\eta_0)}$$

$$p(x|y=1) = b(x)e^{\eta_1^T T(x) - a(\eta_1)}$$

则

$$\begin{aligned} p(y=1|x) &= \frac{p(x|y=1)p(y=1)}{p(x|y=1)p(y=1) + p(x|y=0)p(y=0)} \\ &= \frac{1}{1 + \frac{p(x|y=0)p(y=0)}{p(x|y=1)p(y=1)}} \\ &= \frac{1}{1 + \frac{1-\phi}{\phi} exp(\eta_0^T T(x) - a(\eta_0) - \eta_1^T T(x) + a(\eta_1))} \\ &= \frac{1}{1 + exp[(\eta_0^T T(x) - a(\eta_0) - \eta_1^T T(x) + a(\eta_1)) \ln \frac{1-\phi}{\phi}]} \end{aligned}$$

高斯判别分析事实上包含有比logistic回归更强的假设。高斯判别分析对输入数据分布做了假设，而logistic回归对输入数据符合什么分布，并没有做一般性的假设。所以logistic回归的泛化性能好一些，比高斯判别分析的鲁棒性好。但是高斯判别分析需要的训练数据比logistic回归需要的训练数据要少，也即即使只有少量的训练数据，即使其不完全符合高斯分布，高斯判别分析也能得出很好的分类模型。

§ 3.2 朴素贝叶斯法

3.2.1 贝叶斯公式

划分 设S为实验E的样本空间，$B_1, B_2, \cdots, B_n$为E的一组事件，若
(1)$B_i B_j = \emptyset, i \neq j, i, j = 1, 2, \cdots, n$
(2)$B_1 \cup B_2 \cup \cdots \cup B_n = S$
则称$B_1, B_2, \cdots, B_n$为样本空间S的一个**划分**。

全概率公式 设实验E的样本空间为S，A为E的事件，$B_1, B_2, \cdots, B_n$为S的一组划分，且$P(B_i) > 0, i = 1, 2, \cdots, n$，则

$$P(A) = \sum_{i=1}^{n} P(A|B_i)$$

称为**全概率公式**。

贝叶斯公式 设实验E的样本空间为S，A为E的事件，$B_1, B_2, \cdots, B_n$为S的一组划分，

且$P(A)>0, P(B_i)>0, i=1,2,\cdots,n$，则

$$P(B_i|A)=\frac{P(A|B_i)P(B_i)}{P(A)}=\frac{P(A|B_i)P(B_i)}{\sum_{i=1}^{n}P(A|B_i)}$$

称为**贝叶斯公式**。

其中，$P(A), P(B_i)$分别称为先验概率（prior probability）。$P(A|B_i), P(B_i|A)$称为后验概率（posterior probability）。

3.2.2 朴素贝叶斯法

3.2.2.1 模型

朴素贝叶斯法（naive Bayes）之所以被称为朴素的，是因为其包含有一些朴素的假设：类条件独立假设。

假设D是训练元组及其分类标号的集合。特征向量表示为$X=\{x_1,x_2,\cdots,x_n\}$，其中$x_1,x_2,\cdots,x_n$为n个属性的取值。假设有m个类，$C_1,C_2,\cdots,C_m$。给定元组X，朴素贝叶斯分类法将X分给后验概率最大的类。即，X属于C_i类，当且仅当

$$P(C_i|X)>P(C_j|X), j=1,2,\cdots,m, i\neq j$$

则这个分类过程等价于最大化后验概率$P(C_i|X)$。由贝叶斯公式得

$$\max_i P(C_i|X)=\frac{P(X|C_i)P(C_i)}{P(X)}$$

其中，$P(X)$对所有分类都是一样的，$P(C_i)$如果做等概率假设，即$P(C_1)=P(C_2)=\cdots=P(C_m)$，则上式等价于

$$\max_i P(X|C_i)$$

否则，用训练样本里面每个分类出现的频次作为其先验概率估计值，即$P(C_i)=\frac{freq(C_i)}{|D|}$，则最优化问题等价于

$$\max_i P(X|C_i)P(C_i)$$

下面考虑$P(X|C_i)$的计算方法。为了简化计算量，可以做**类条件独立**的朴素假设，即属性之间不存在依赖关系，则

$$P(X|C_i)=\prod_{k=1}^{n}P(x_k|C_i)=P(x_1|C_i)P(x_2|C_i)\cdots P(x_n|C_i)$$

这样，求$P(X|C_i)$的问题就转化为求属性x_k在分类C_i中出现频次的问题了。此时，需要区分属性取值是离散的，还是连续的。

(1)如果属性k的取值是离散的，则有$P(x_k|C_i)=\frac{freq(<x_k,C_i>)}{freq(C_i)}$。

(2)如果属性k的取值是连续的，假定连续值属性的取值$x_k \backsim \mathcal{N}(\mu, \sigma)$，则定义

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

同时，有

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

μ_{C_i}和σ_{C_i}分别表示C_i类中属性k的均值和标准差。

3.2.2.2 实现

计算概率及其乘积的过程涉及到数据的一些处理。首先是数据平滑问题，有些属性值可能在某类的统计数据中并不出现，这样会导致某一属性的概率值为0，影响结果。实际使用时可以通过对分子分母同时加上一个常数来避免概率为0。

另一问题是数据下溢出的问题，因为涉及到很多小数的乘积。一个解决方法是对概率乘积取自然对数。这样$p_1 p_2$ 就可以表示为$\ln p_1 + \ln p_2$，取自然对数保证结果的单调性是相同的。

第4章　决策树

第5章　最大熵

第6章　支持向量机

支持向量机首先是一个二类分类模型。这点与感知机类似。其次，它是定义在特征空间上的间隔最大的线性分类器，间隔最大使它有别于感知机。因为根据感知机训练出来的分离超平面可能不是唯一的，例如一些与分离超平面平行的平面可能也满足分类要求。但是，当考虑到间隔最大化时，这样的平面就可以唯一确定下来。这里需要留意的几个概念包括：特征空间和间隔。

支持向量机的分类。按照训练数据是否线性可分，当训练数据线性可分时，通过硬间隔最大化（hard margin maximization），定义线性可分支持向量机（linear support vector machine in linearly separable case）或者硬间隔支持向量机。当训练数据近似线性可分（也即可能有一些噪声点是会导致原数据不是线性可分的）时，通过软间隔最大化（soft margin maximization），定义线性支持向量机（linear support vector machine），或者软间隔支持向量机。当训练数据线性不可分（也即分离超平面可能是超曲面）时，通过核技巧（kernel trick）和软间隔最大化，定义非线性支持向量机（non-linear support vector machine）。

在支持向量机学习中，拉格朗日对偶法之所以会发生特殊的作用还在于，对偶表示之后数据仅作为Gram矩阵的项出现，而不需要通过单个属性出现。类似地，在决策函数的对偶表示里仅需要与测试点输入的内积。

支持向量机也被有些人认为是现存最好的监督学习算法。

§ 6.1　模型

首先定义支持向量机的输出$y \in \{-1, 1\}$。

其次定义支持向量机的参数，将之前一直使用的θ拆分成法向量w和截距b的形式，支持向量机的训练目的即得到分离超平面（separating hyperplane）$w^T x + b$。

最后，SVM的判别函数

$$h_{w,b}(x) = g(w^T x + b)$$

其中

$$g(z)=\begin{cases}1 & , \quad z\geq 0\\ -1 & , \quad z<0\end{cases}$$

§ 6.2 函数间隔和几何间隔（functional and geometric margins）

训练样本$(x^{(i)},y^{(i)})$函数间隔定义为

$$\hat{\gamma}^{(i)}=y^{(i)}(w^Tx^{(i)}+b)$$

不难看出，恒有$\hat{\gamma}^{(i)}\geq 0$。

所有训练样本与分离超平面之间的函数间隔的最小值定义为

$$\hat{\gamma}=\min_{i=1,2\cdots,m}\hat{\gamma}^{(i)}$$

函数间隔可以通过等比例增大w,b来增大。比如令$w=2w,b=2b$，则有$\hat{\gamma}^{(i)}=2\hat{\gamma}^{(i)}$。所以需要引入几何间隔的概念。

函数间隔实际上并不是点到平面的距离。点到平面的距离称为几何间隔，定义为

$$\gamma^{(i)}=y^{(i)}((\frac{w}{\|w\|})^Tx^{(i)}+\frac{b}{\|w\|})$$

同时，定义几何间隔的最小值

$$\gamma=\min_{i=1,2\cdots,m}\gamma^{(i)}$$

几何间隔不会随着w,b的变化而变化，因为只要平面本身没有变，点到平面的距离是不会随着平面的表达方式的变化而变化的。

函数间距与几何间距的关系。

$$\gamma^{(i)}=\frac{\hat{\gamma}^{(i)}}{\|w\|}$$

$$\gamma=\frac{\hat{\gamma}}{\|w\|}$$

这个关系对于简化最优化的目标公式有用。

§ 6.3 SVM模型

按照几何间隔最大话，定义SVM的最优化模型

$$\begin{aligned} \max_{w,b} \quad & \gamma \\ \text{s.t.} \quad & y^{(i)}((\frac{w}{\|w\|})^T x^{(i)} + \frac{b}{\|w\|}) \geq \gamma, \quad i = 1, 2, \cdots, m \end{aligned}$$

引入函数间隔与几何间隔的关系，将约束条件变成线性的

$$\begin{aligned} \max_{w,b} \quad & \frac{\hat{\gamma}}{\|w\|} \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq \hat{\gamma}, \quad i = 1, 2, \cdots, m \end{aligned}$$

目标函数$\frac{\hat{\gamma}}{\|w\|}$不是凸函数（non-convex），目前还没有方法来解决这类优化问题。

从前面关于函数间隔的讨论可以看出，函数间隔本身是可以根据w, b的变化（缩放）而变化的。也即，函数间隔对最优化问题没有影响，可以直接令$\hat{\gamma} = 1$。这一点也可以通过缩放w, b来实现。另一方面，注意到最大化$\frac{1}{\|w\|}$等价于最小化$\|w\|$（为了便于计算，可以使用$\frac{1}{2}\|w\|^2$），所以可以将上面的优化问题改为

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, 2, \cdots, m \end{aligned} \tag{SVM1}$$

这样转化为一个凸二次规划问题（convex quadratic programming）。

§ 6.4 线性可分SVM

线性可分支持向量机的模型如SVM1所示。求解的方法是首先构造SVM1的拉格朗日函数。

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i [y^{(i)}(w^T x^{(i)} + b) - 1] \tag{6.1}$$

注意，先将SVM1中的约束条件改成≤ 0的形式，然后应用拉格朗日乘子法。w, b是目标函数的参数，α是拉格朗日乘子，m是训练样本的个数。

（1）先求$\theta_D(\alpha)=\min\limits_{w,b}L(w,b,\alpha)$。令

$$\nabla_w L(w,b,\alpha)=w-\sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}=0$$

$$\nabla_b L(w,b,\alpha)=\sum_{i=1}^{m}\alpha_i y^{(i)}=0$$

解得

$$w=\sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}$$

$$\sum_{i=1}^{m}\alpha_i y^{(i)}=0$$

将上式代入6.1中，得到

$$L(w,b,\alpha)=\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y^{(i)}y^{(j)}x^{(i)}x^{(j)}-\sum_{i=1}^{m}\alpha_i[y^{(i)}((\sum_{i=j}^{m}\alpha_j y^{(j)}x^{(j)})x^{(i)}+b)]+\sum_{i=1}^{m}\alpha_i$$

$$=-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y^{(i)}y^{(j)}x^{(i)}x^{(j)}+\sum_{i=1}^{m}\alpha_i$$

（2）求对偶问题关于α的最优解

$$\max_{\alpha}\quad W(\alpha)=-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y^{(i)}y^{(j)}\langle x^{(i)},x^{(j)}\rangle+\sum_{i=1}^{m}\alpha_i$$

$$\text{s.t.}\quad \alpha_i\geq 0, i=1,2,\cdots,m$$

$$\sum_{i=1}^{m}\alpha_i y^{(i)}=0$$

设α^*是上述对偶问题的解，则原始问题的解为

$$w^*=\sum_{i=1}^{m}\alpha_i^* y^{(i)}x^{(i)}$$

$$b^*=y^{(j)}-\sum_{i=1}^{m}\alpha_i^* y^{(i)}\langle x^{(i)},x^{(j)}\rangle$$

其中$(x^{(j)},y^{(j)})$是支持向量。b^*可以通过$y^{(j)}((w^*)^Tx^{(j)}+b^*)-1=0$求得。或者如Andrew Ng的讲义中所说，用支持向量的正例与负例所对应的与分离超平面平行的平面的截距的平均值来定

义b^*

$$b^* = -\frac{\max\limits_{i:y^{(i)}=-1}(w^*)^T x^{(i)} + \min\limits_{i:y^{(i)}=1}(w^*)^T x^{(i)}}{2}$$

根据上述解，可以求得分离超平面

$$\begin{aligned} w^T x + b &= (\sum_{i=1}^{m} \alpha_i^* y^{(i)} x^{(i)})^T x + b \\ &= \sum_{i=1}^{m} \alpha_i^* y^{(i)} \langle x^{(i)}, x \rangle + b \\ &= 0 \end{aligned}$$

分类决策函数

$$f(x) = sign(\sum_{i=1}^{m} \alpha_i^* y^{(i)} \langle x^{(i)}, x \rangle + b)$$

注意，只有支持向量的$\alpha > 0$，其他向量的$\alpha = 0$。也就是说，对于一个新的待分类的点x，分类的结果只取决于x与支持向量的内积。而支持向量的个数往往比训练集中全部向量的个数小得多，这一点对于支持向量机很重要。

§ 6.5 线性SVM

当训练样本中存在一些特异点（outlier）时，训练样本可能不是线性可分的。假设去掉特异点之后的训练样本是线性可分的，则可以构造线性SVM。线性不可分意味着某些样本点$(x^{(j)}, y^{(j)})$不满足函数间隔大于等于1的约束条件。为了解决这个问题，可以通过对每个训练样本引入松弛变量$\xi_j \geq 0$，使得函数间隔加上松弛变量大于等于1.这样，约束条件变成

$$y^{(j)}(w^T x^{(j)} + b) + \xi_j \geq 1$$

同时，对每个松弛变量ξ_j，支付一个代价ξ_j，则目标函数变成

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i$$

其中$C > 0$称为惩罚参数。C值大时，对误分类的惩罚增大，C值小时，对误分类的惩罚减小。线性SVM的优化形式可以表述为

$$\min_{w,b,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i \tag{SVM2}$$

$$\text{s.t.} \quad y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, 2, \cdots, m$$

$$\xi_i \geq 0, \qquad i = 1, 2, \cdots, m$$

使用拉格朗日乘子法得到其对偶形式

$$\max_{\alpha} \quad W(\alpha) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y^{(i)}y^{(j)}\langle x^{(i)}, x^{(j)}\rangle + \sum_{i=1}^{m}\alpha_i$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C, i = 1, 2, \cdots, m$$

$$\sum_{i=1}^{m}\alpha_i y^{(i)} = 0$$

有意思的是，SVM2的对偶形式跟SVM1的对偶形式的差别仅仅是$0 \leq \alpha_i$变成了$0 \leq \alpha_i \leq C$。同时，KKT对偶互补性条件变成

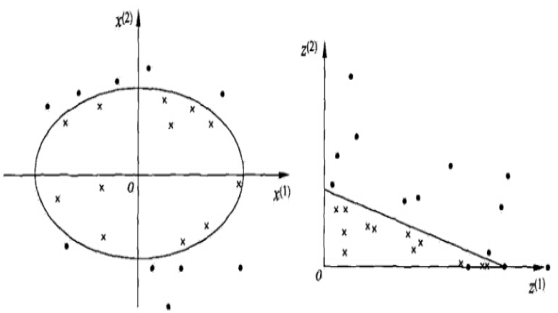
$$\alpha_i = 0 \Rightarrow y^{(i)}(w^T x^{(i)} + b) \geq 1 \tag{6.2}$$

$$\alpha_i = C \Rightarrow y^{(i)}(w^T x^{(i)} + b) \leq 1 \tag{6.3}$$

$$0 < \alpha_i < C \Rightarrow y^{(i)}(w^T x^{(i)} + b) = 1 \tag{6.4}$$

§ 6.6 非线性SVM

6.6.1 非线性分类问题



上图左边显示了一个非线性分类问题，使用一个分离超平面无法将正负实例点完全分开，

但是使用一个分离超曲面（椭圆）$w_1x_1^2 + w_2x_2^2 + b = 0$ 可以分开。我们做一些变化，令

$$\begin{aligned} z &= (z_1, z_2)^T \\ &= \phi(x) \\ &= (x_1^2, x_2^2)^T \end{aligned}$$

则左图的分离超曲面可以表示为$w_1z_1 + w_2z_2 + b = 0$。如右图所示，这样通过将原输入空间$\mathcal{X} \subset \mathbb{R}^2$映射到特征空间$\mathcal{Z} \subset \mathbb{R}^2$，使得原来线性不可分的数据变成了线性可分的。由此可见，解决线性不可分问题的关键在于选择映射函数ϕ.

假设原始训练数据集为

$$T = \{(x^{(i)}, y^{(i)}) | i = 1, 2, \cdots, m\} \in (\mathbb{R}^n \times \mathcal{Y})^m$$

其中，$x^{(i)} \in \mathbb{R}^n$，$y^{(i)} \in \mathcal{Y} = \{-1, 1\}$。引入映射$\phi : \mathbb{R}^n \to \mathcal{H}$，使得$z = \phi(x) \in \mathcal{H}$。则原始训练集变成

$$T_\phi = \{(z^{(i)}, y^{(i)}) | i = 1, 2, \cdots, m\} \in (\mathcal{H} \times \mathcal{Y})^m$$

其中，$z^{(i)} = \phi(x^{(i)}) \in \mathcal{H}$，$y^{(i)} \in \mathcal{Y} = \{-1, 1\}$。

原问题变为在希尔伯特空间$\mathcal{H}$中求得线性划分超平面$(w^*)^T z + b^* = 0$，从而导出$\mathbb{R}^n$空间中的分离超曲面$(w^*)^T \phi(x) + b^* = 0$。

在希尔伯特空间$\mathcal{H}$中构造对应分类问题的最优化形式，并应用拉格朗日乘子法（与欧式空间中的拉格朗日乘子法相同，同样需要满足KKT条件），不难得出希尔伯特空间中分类问题的如下对偶形式

$$\max_{\alpha} \quad W(\alpha) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y^{(i)}y^{(j)}\langle\phi(x^{(i)}), \phi(x^{(j)})\rangle + \sum_{i=1}^{m}\alpha_i$$

$$\text{s.t.} \quad 0 \le \alpha_i \le C, i = 1, 2, \cdots, m$$

$$\sum_{i=1}^{m}\alpha_i y^{(i)} = 0$$

并且有

$$w^* = \sum_{i=1}^{m}\alpha_i^* y^{(i)}\phi(x^{(i)})$$

$$b^* = y^{(j)} - \sum_{i=1}^{m}\alpha_i^* y^{(i)}\langle\phi(x^{(i)}), \phi(x^{(j)})\rangle$$

分类决策函数

$$f(x) = sign(\sum_{i=1}^{m} \alpha_i^* y^{(i)} \langle \phi(x^{(i)}), \phi(x) \rangle + b)$$

6.6.2 核（Kernel）

在希尔伯特空间中，变化ϕ完全是通过内积的形式$\langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle$和$\langle \phi(x^{(i)}), \phi(x) \rangle$出现，因此可以考虑使用一个函数代替$\phi$的内积形式$\langle \phi(\bullet), \phi(\bullet) \rangle$，这个函数称为核函数，定义如下

$$K(x, x') = \langle \phi(x), \phi(x') \rangle$$

内积实际上描述的是向量之间的相似度（因为内积可以同时表征向量之间的长度差、向量之间的距离和向量之间的夹角）。在将原空间映射为希尔伯特空间时，向量之间的相似度同时映射到希尔伯特空间。但是映射后的希尔伯特空间可能维数非常高（可能是无穷维，比如采用高斯核函数时，映射函数映射后的希尔伯特空间是无穷维的），而采用核技巧，又将希尔伯特空间中的内积运算转化为原空间中的代数运算。这样有效的降低了运算的复杂度。注意，核函数$K(x, x')$要保持内积所表征的向量之间的相似度，即当x与x^*相似时（比如平行），$K(x, x')$会很大。但是当x和x^*不相似时（比如正交），$K(x, x')$会很小。

为了描述核函数的必要特性，首先定义核矩阵（或者Gram矩阵）。

定义 6.1 (核矩阵) 给定函数$K(x, x') : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$，对任意的$x^{(i)} \in \mathbb{R}^n$，$i = 1, 2, \cdots, m$，函数$K(x, x')$对应的核矩阵为

$$K = [K(x^{(i)}, x^{(j)})]_{m \times m}$$

核矩阵的对称性显而易见。

定理 6.1 (核函数的特征) 定义在$\mathbb{R}^n \times \mathbb{R}^n$上的对称函数$K(x, x')$是核函数的充要条件是对任意的$x^{(i)} \in \mathbb{R}^n$，其核矩阵$K$半正定。

应该注意，核技巧是一种比SVM更为广泛的技巧。

6.6.3 常用核函数

1.多项式核函数（polynomial kernel function）

d阶其次多项式函数

$$K(x, z) = (x \cdot z)^p$$

和d阶非齐次多项式函数

$$K(x, z) = (x \cdot z + 1)^p$$

都是核函数。

2.高斯核函数（Gaussian kernel function）

$$K(x,z) = exp(-\frac{\|x-z\|^2}{2\sigma^2})$$

对应的SVM是高斯径向基函数（radial basis function）分类器。

第7章　提升方法

第8章　EM方法

§ 8.1 Jensen不等式

第9章　隐马尔可夫模型

第10章　附录

§ 10.1 矩阵分析（matrix analysis）

10.1.1 迹(Trace)和导数(matrix derivatives)

令$f:\mathbb{R}^{m\times n}\mapsto\mathbb{R}$表示将$m\times n$（$m$-by-$n$）矩阵映射为实数的函数。定义$f$对矩阵$\boldsymbol{A}$的导数

$$\nabla_{\boldsymbol{A}}f(\boldsymbol{A})=\begin{pmatrix}\frac{\partial f(\boldsymbol{A})}{\partial a_{11}} & \cdots & \frac{\partial f(\boldsymbol{A})}{\partial a_{1n}}\\ \vdots & \ddots & \vdots\\ \frac{\partial f(\boldsymbol{A})}{\partial a_{n1}} & \cdots & \frac{\partial f(\boldsymbol{A})}{\partial a_{nn}}\end{pmatrix}$$

矩阵的迹（trace）表示的是矩阵的对角元素的和，

$$tr\boldsymbol{A}=\sum_{i=1}^{n}a_{ii}$$

假设A，B，C，D均是方阵

$$trABCD=trDABC=trCDAB=trBCDA \tag{10.1}$$

循环将最右边矩阵放到最左边。假设a是实数

$$trA=trA^T \tag{10.2}$$

$$tr(A+B)=trA+trB \tag{10.3}$$

$$traA=atrA \tag{10.4}$$

下面的一些公式出自Andrew Ng的机器学习讲义，这里证明一下。

$$\nabla_A trAB=B^T \tag{10.5}$$

$$\nabla_{A^T} f(A) = (\nabla_A f(A))^T \tag{10.6}$$

$$\nabla_A trABA^TC = CAB + C^TAB^T \tag{10.7}$$

$$\nabla_A |A| = |A|(A^{-1})^T \tag{10.8}$$

证明 (1) $(\nabla_A trAB)_{ij} = \frac{\partial trAB}{\partial a_{ij}} = \frac{\partial \sum_m \sum_k a_{mk} b_{km}}{\partial a_{ij}}$，只有当$m = i, k = j$时才有$a_{ij}$的系数，所以$(\nabla_A trAB)_{ij} = b_{ji}$，即证。

(2) $(\nabla_{A^T} f(A))_{ij} = \frac{\partial f(A)}{\partial a_{ji}}$，即证。

(3) $trABA^TC = \sum_m \sum_k \sum_t \sum_s a_{mk} b_{kt} a_{st} c_{sm}$，所以

$$(\nabla_A trABA^TC)_{ij} = \frac{\partial \sum_m \sum_k \sum_t \sum_s a_{mk} b_{kt} a_{st} c_{sm}}{\partial a_{ij}}$$

$$= \sum_m \sum_k \sum_t \sum_s \frac{\partial a_{mk}}{\partial a_{ij}} b_{kt} a_{st} c_{sm} + \sum_m \sum_k \sum_t \sum_s a_{mk} b_{kt} \frac{\partial a_{st}}{\partial a_{ij}} c_{sm}$$

左边，令$m = i, k = j$，右边，令$s = i, t = j$，

$$\begin{aligned}(\nabla_A trABA^TC)_{ij} &= \sum_t \sum_s b_{jt} a_{st} c_{si} + \sum_m \sum_k a_{mk} b_{kj} c_{im} \\ &= \sum_t \sum_s b_{jt} a_{st} c_{si} + \sum_m \sum_k c_{im} a_{mk} b_{kj} \\ &= (BA^TC)_{ji} + (CAB)_{ij} \\ &= (C^TAB^T + CAB)_{ij}\end{aligned}$$

§ 10.2 常用不等式

10.2.1 柯西不等式（Cauchy Inequality）

柯西不等式，又称柯西-施瓦茨不等式（Cauchy-Schwarz inequality）。对于一个内积空间所有向量$\boldsymbol{x}$和$\boldsymbol{y}$，

$$|\langle \boldsymbol{x}, \boldsymbol{y} \rangle|^2 \leq \langle \boldsymbol{x}, \boldsymbol{x} \rangle \cdot \langle \boldsymbol{y}, \boldsymbol{y} \rangle$$

其中$\langle \cdot, \cdot \rangle$表示内积（点积），当且仅当$\boldsymbol{x}$与$\boldsymbol{y}$线性相关时等式成立。

对于欧几里得空间$\mathbb{R}^2$，

$$(\sum_{i=1}^{n} x_i y_i)^2 \leq (\sum_{i=1}^{n} x_i^2)(\sum_{i=1}^{n} y_i^2)$$

当且仅当$\frac{x_1}{y_1} = \frac{x_2}{y_2} = \cdots = \frac{x_n}{y_n}$时等式成立。

10.2.2 赫尔德不等式（Hölder Inequality）

赫尔德不等式揭示了L^p空间的相互关系。设S为测度空间，$1 \leq p, q \leq \inf$，且$\frac{1}{p} + \frac{1}{q} = 1$，若$f \in L^p(S)$，$g \in L^q(S)$，则$fg \in L^1(S)$，且

$$\| fg \|_1 \leq \| f \|_p \| g \|_q$$

写成序列或向量的形式

$$\sum_{i=1}^{n} |a_i b_i| \leq (\sum_{i=1}^{n} |a_i|^p)^{\frac{1}{p}} (\sum_{i=1}^{n} |b_i|^q)^{\frac{1}{q}}$$

10.2.3 闵可夫斯基不等式（Minkowski Inequality）

闵可夫斯基不等式表明L^p空间是一个赋范向量空间。设S是一个度量空间，$f, g \in L^p(S), 1 \leq p \leq \inf$，那么$f + g \in L^p(S)$，有

$$\| f + g \|_p \leq \| f \|_p + \| g \|_p$$

写成序列或向量的形式

$$(\sum_{k=1}^{n} |x_k + y_k|^p)^{\frac{1}{p}} \leq (\sum_{k=1}^{n} |x_k|^p)^{\frac{1}{p}} (\sum_{k=1}^{n} |y_k|^p)^{\frac{1}{p}}$$

参考文献

[1] 李航著. 《统计学习方法》. 北京:清华大学出版社，2012，3

[2] Jiawei Han, Micheline Kamber, Jian Pei 著.范明, 孟小峰译. 《数据挖掘：概念与技术》. 机械工业出版社, 2012, 8

[3] Tom M. Mitchell 著.曾华军等译. 《机器学习》.机械工业出版社, 2003, 1

[4] Andrew Ng, 《Machine Learning》公开课讲义