相关算法理论

线性判别分析

协方差

给定样本集 $D=\{(x_1,\ y_1),\ (x_2,\ y_2),\ldots,\ (x_m,\ y_m)\}$，$y\in\{0,\ 1\}$，$x_i$ 为第 i 个样本，y_i 为 x_i 的类别标记，构造矩阵 $X=(x_1,\ x_2,\ \ldots,\ x_n)$，将样本的 n 个特征均看作随机变量，则第 i 个特征的观测值的集合构成 X 的行向量 β_i，$i=1,2,\ldots,n$，即 $X=(\beta_1;\ \beta_2;\ \ldots\ \beta_n;)$，第 k 个样本的第 i 个分量可以表示为 x_{ik} 或 β_{ik}, $k=1,2,\ldots,m$，则 β_i，β_j 之间的协方差为

$$cov(\beta_i,\ \beta_j)=\frac{1}{m-1}\sum_{k=1}^{m}(\beta_{ik}-\overline{\mu_i})(\beta_{jk}-\overline{\mu_j})=\frac{1}{m-1}(\beta_i-\mu_i)(\beta_j-\mu_j)^T.\quad(1)$$

其中，$\overline{\mu_i}$ 为 β_i 的期望，$\mu_i=(\overline{\mu_i})_{1\times m}=(\overline{\mu_i},\ \overline{\mu_i},\ \ldots,\ \overline{\mu_i})$. 样本任意两个特征之间的协方差构成了协方差矩阵 Σ，称 Σ 为 D 中样本的协方差矩阵

$$\Sigma=\begin{pmatrix}cov(\beta_1,\beta_1) & cov(\beta_1,\beta_2) & \ldots & cov(\beta_1,\beta_n)\\ cov(\beta_2,\beta_1) & cov(\beta_2,\beta_2) & \ldots & cov(\beta_2,\beta_n)\\ \cdots & & & \cdots\\ cov(\beta_n,\beta_1) & cov(\beta_n,\beta_2) & \ldots & cov(\beta_n,\beta_n)\end{pmatrix}\qquad(2)$$

将式 (1) 代入式 (2)，并令均值向量 $\mu=(\overline{\mu_1};\ \overline{\mu_2};\ \ldots;\ \overline{\mu_n};)^T$，则

$$\begin{aligned}\Sigma&=\frac{1}{m-1}\sum_{k=1}^{m}\begin{pmatrix}(\beta_{1k}-\overline{\mu_1})(\beta_{1k}-\overline{\mu_1}) & \ldots & \ldots & (\beta_{1k}-\overline{\mu_1})(\beta_{nk}-\overline{\mu_n})\\ \cdots & & & \cdots\\ \cdots & & & \cdots\\ (\beta_{nk}-\overline{\mu_n})(\beta_{1k}-\overline{\mu_1}) & \ldots & \ldots & (\beta_{nk}-\overline{\mu_n})(\beta_{nk}-\overline{\mu_n})\end{pmatrix}\\ &=\frac{1}{m-1}\sum_{k=1}^{m}(x_k-\mu)(x_k-\mu)^T\end{aligned}\qquad(3)$$

线性判别分析

线性判别分析 (Linear Discriminant Analysis，简称 LDA) 是对 Fisher 的线性鉴别方法的归纳，是一种经典的线性学习方法. 该方法可用于解决二分类问题，现在也常用作对高维样本的监督降维. LDA 用于二分类时，其主要思想是试图将所有样本投影到一条直线上，使得不同类样本点尽量远离，同类别的样本点尽量接近. 找到这条直线后，对所有待预测的样本，都将其投影至该直线上，再根据投影的位置判断样本的类别.

在获得数据集 $D=\{(x_1,y_1),\ (x_2,y_2),\ \ldots,\ (x_m,y_m)\}$, $y\in\{0,\ 1\}$ 后，令 X_i，μ_i，Σ_i，$i\in\{0,\ 1\}$ 表示第 i 类的集合、均值向量、协方差矩阵，ω 为目标直线. 为使同类样本投影点相近，异类投影点相离，应使同一类样本投影点在直线 ω 上的方差 Σ 尽量小，不同类样本投影中心的距离 d 尽量大. 定义 J

$$\begin{aligned}J=\frac{d}{\Sigma}&=\frac{||\omega^T\mu_0-\omega^T\mu_1||_2^2}{\omega^T\Sigma_0\omega+\omega^T\Sigma_1\omega}\\ &=\frac{\omega^T(\mu_0-\mu_1)(\mu_0-\mu_1)^T\omega}{\omega^T(\Sigma_0+\Sigma_1)\omega}\end{aligned}\qquad(4)$$

我们需要求出使得 J 最大的 ω，即

$$\omega = arg\max_{\omega} J = arg\max_{\omega} \frac{\omega^T(\mu_0-\mu_1)(\mu_0-\mu_1)^T\omega}{\omega^T(\Sigma_0+\Sigma_1)\omega} \tag{5}$$

由于分子分母都是关于 ω 的二次项，从而式 (4) 的解与 $||\omega||$ 无关，仅与 ω 方向有关. 不妨令 $\omega^T(\Sigma_0+\Sigma_1)\omega=1$，则问题转化为

$$\min_{\omega}\left[-\omega^T(\mu_0-\mu_1)(\mu_0-\mu_1)^T\omega\right] \tag{6}$$
$$s.t.\ \ \omega^T(\Sigma_0+\Sigma_1)\omega=1$$

对式 (6) 使用拉格朗日乘子法，进一步地，因为 $(\mu_0-\mu_1)^T\omega$ 为实数，不妨令 $(\mu_0-\mu_1)(\mu_0-\mu_1)^T\omega=\lambda(\mu_0-\mu_1)$，可得

$$\omega=(\Sigma_0+\Sigma_1)^{-1}(\mu_0-\mu_1) \tag{7}$$

这里假设了各类样本的协方差矩阵满秩. 在得到目标直线后，LDA 将测试集样本投影至该直线上，并把聚集在一起的投影点分为一类，最终得到预测结果.

LDA 的不足之处在于若直线上两类别的距离较近，该方法对于处在两类边界的投影点的归类有些模糊，容易分类出错. 但总体来说 LDA 能够较准确找到对数据集进行有效分类的方向，而它的另一个用途——样本降维，如今被广泛应用于数据预处理，为复杂样本的处理提供了有力工具. 将 LDA 推广到 N 分类后，该方法可以将样本维数从高维降至最多 N-1 维，大大简化了模型复杂度.

决策树

算法思想

决策树 (decision tree) 是一种应用广泛的分类与回归机器学习方法，最早于 1962 年由 E. B. Hunt 提出. 他的学生 Quinlan 推广了导师的思想，并先后提出了著名的 ID3、C4. 5 算法. 决策树采用“分而治之”的策略，树的生成是一个递归过程. 在给定数据集 $D=\{(x_1,\,y_1),\,(x_2,\,y_2),\ldots,\,(x_m,\,y_m)\},\,y\in\{0,\,1\}$ 与非空样本特征集合 $\beta=(\beta_1,\,\beta_2,\,\ldots,\,\beta_n)$ 后，算法主要思想为：

1. 将数据集 D 作为树 T 根结点；
2. 若 D 中样本在 β 上取值相同，则将根节点标记为叶节点，其类别标记为 D 中样本数最多的类，树生成完毕；否则，进入3；
3. 从 β 中选择最优划分特征 β_*，并对 β_* 的每一个可能值 b_i，依 $\beta_*=b_i$ 将 D 分割为若干非空子集 D_i，将 D_i 中样本数最多的类作为标记，构建子结点；
4. 根结点与所有子结点构成树 T；
5. 对第 i 个子结点，以 D_i 为数据集，以 $\beta-\{\beta_*\}$ 为特征集，重复 1 至 4，生成子树 T_i，将 T_i 与其父结点拼接.

上述过程的关键在于最优划分特征如何选出以及子树的生成何时停止. 对于第一点，我们引入信息增益、信息增益比和基尼指数三个指标来筛选特征；对于第二点，通常子树生成会进行至特征集只含一个元素或某一子结点中样本均为同一类别为止，但这常常会导致过拟合问题，因此我们引入决策树生成过程中的“预剪枝”和决策树生成后的“后剪枝”策略来对树的分支进行简化.

经典算法

经典的决策树算法有 ID3、C4. 5 和 CART 算法. ID3 与 C4. 5 中涉及了熵的概念. 给定数据集 D 和特征 β_i，设样本共有 N 个类 C_k，$k=1,2,\ldots,N$，设特征 β_i 有 n 个不同的可能值 $\{b_1,\,b_2,\,\ldots,\,b_n\}$，根据 β_i 的取值将 D 划分为 n 个子集 $D_1,\,D_2,\,\ldots,\,D_n$，子集 D_i 中属于类 C_k 的样本集合记为 D_{ik}. 数据集 D 的熵定义为

$$H(D)=-\sum_{k=1}^{N}\frac{|C_k|}{|D|}\log_2\frac{|C_k|}{|D|} \tag{8}$$

特征 β_i 对数据集 D 的条件熵为

$$H(D|\beta_i)=\sum_{i=1}^{n}\frac{|D_i|}{|D|}H(D_i)=-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\sum_{k=1}^{N}\frac{|D_{ik}|}{|D_i|}\log_2\frac{|D_{ik}|}{|D_i|} \tag{9}$$

特征 β_i 对数据集 D 的信息增益定义为

$$g(D,\ \beta_i)=H(D)-H(D|\beta_i) \tag{10}$$

特征 β_i 对数据集 D 的信息增益比定义为

$$g_R(D,\ \beta_i)=\frac{g(D,\ \beta_i)}{H_{\beta_i}(D)} \tag{11}$$

$$H_{\beta_i}(D)=-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\log_2\frac{|D_i|}{|D|}$$

其中，$H_{\beta_i}(D)$ 为数据集 D 关于特征 β_i 的值的熵. 每次结点的划分选择信息增益 (ID3) 或信息增益比 (C4. 5) 最大的特征作为最优划分特征. 而 CART 算法则使用基尼指数作为指标. 对于 N 分类问题，C_k 是数据集 D 中属于第 k 类的样本子集，D 的基尼指数为

$$Gini(D)=1-\sum_{k=1}^{N}(\frac{|C_k|}{|D|})^2 \tag{12}$$

在特征 β_i 的条件下，集合 D 的基尼指数定义为

$$Gini(D,\beta_i)=\frac{|D_1|}{|D|}Gini(D_1)+\frac{|D_2|}{|D|}Gini(D_2) \tag{13}$$

其中 D_1为特征 $\beta_i=b$ (b 为 β_i 某一可能值) 的所有样本的集合，D_2 是 D_1 的关于 D 的补集. CART 每次选择使得基尼指数最小的特征作为最优特征.

支持向量机

支持向量机 (support vector machines,简称SVM) 是一种有监督的学习模型，可用于分类和回归问题. Vapnik 在 1963 年首次提出了支持向量这一概念，之后不断完善形成了 SVM 的雏形. SVM 基于结构风险最小化原则，在特征空间中构建最优超平面，使得学习器得到全局最优化，换句话说，SVM 试图在数据集的特征空间中求解一个能够正确划分训练数据集的超平面并尽量使得两类样本到超平面的最小距离最大化.

假设超平面 Γ 方程为:

$$\omega^T x+b=0 \tag{14}$$

其中 $\omega=(\omega_1;\ \omega_2;\ \ldots;\ \omega_n;)$ 为超平面法向量，b 为截距项.

给定样本集 $D=\{(x_1,y_1),\ (x_2,y_2),\ \ldots,\ (x_m,y_m)\},\ y\in\{-1,\ 1\}$ ，如果超平面 Γ 能将不同类的样本完全划分开，即对于 $\forall\ (x_i,\ y_i)\in D$，若 $y_i=+1$ ，有 $\omega^T x+b>0$ ；若 $y_i=-1$ ，有 $\omega^T x+b<0$ ，此时我们称样本在特征空间中是线性可分的. 那么样本点 x_i 到超平面 Γ 的距离为

$$d_i=\frac{|\omega^T x_i+b|}{||\omega||}=\frac{y_i(\omega^T x_i+b)}{||\omega||} \tag{15}$$

为使不同类样本到超平面的最小距离最大化，问题转化为以下最优化问题:

$$\max_{\omega,\ b} d \tag{16}$$
$$s.t.\quad \frac{y_i(\omega^T x_i + b)}{||\omega||} \geq d,\ i = 1, 2, \ldots, m$$

令 $\gamma = \min y_i(\omega^T x_i + b)$，则式 (16) 等价于

$$\max_{\omega,\ b} \frac{\gamma}{||\omega||} \tag{17}$$
$$s.t.\quad y_i(\omega^T x_i + b) \geq \gamma,\ i = 1, 2, \ldots, m$$

通过等比例缩放 ω, b 可以使得 $\gamma = 1$，且并不影响目标函数的优化. 同时，注意到最大化 $||\omega||^{-1}$ 等价于最小化 $\frac{1}{2}||\omega||^2$，于是式 (17) 可重写为

$$\max_{\omega,\ b} \frac{1}{2}||\omega||^2 \tag{18}$$
$$s.t.\quad y_i(\omega^T x_i + b) \geq 1,\ i = 1, 2, \ldots, m$$

由拉格朗日乘子法及对偶性，得到式 (18) 的对偶问题

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{19}$$
$$s.t.\quad \sum_{i=1}^{m} \alpha_i y_i = 0,\ \alpha_i \geq 0,\ i = 1, 2, \ldots, m$$

求出 α 后，再根据

$$\omega = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{20}$$
$$b = y_i - \sum_{i=1}^{m} \alpha_i y_i x_i^T x_j$$

最终得到分类决策函数：

$$f(x) = sign(\omega^T x + b) \tag{21}$$

其中上述最优化问题应满足 KKT 条件：

$$\begin{cases} \alpha_i \geq 0; \\ y_i f(x_i) - 1 \geq 0; \\ \alpha_i (y_i f(x_i) - 1) = 0. \end{cases} \tag{22}$$

当样本线性不可分时，通过加上松弛变量改变目标函数与约束条件为

$$\frac{1}{2}||\omega||^2 + C \sum_{i=1}^{m} \xi_i \tag{23}$$
$$y_i(\omega^T x_i + b) \geq 1 - \xi_i,\ i = 1, 2, .., m$$

进行求解. 若效果仍不理想，说明实际样本并非线性分类问题，则选择使用核函数将原空间的数据映射到新空间，然后在新空间中用线性方法进行学习，常用核函数有线性核，多项式核，高斯核 (RBF核) 等，其中，使用高斯核，并结合泰勒展开公式可将原样本数据映射至无穷维空间.

评价指标

对于二分类问题，我们通常将精确率 (precision) 与召回率 (recall) 作为评价指标. 将违约类记为正类，未违约类记为负类，则分类器在测试集上的预测与实际类别之间有如下四种情况，即混淆矩阵：

表2. 1 混淆矩阵

实际\预测	Positive	Negative
正类	TP	FN
负类	FP	TN

TP 代表正类预测结果为正类数，FN 代表正类预测结果为负类数，FP 代表负类预测结果为正类数，TN 代表负类预测结果为负类数. 定义精确率 P 和召回率 R 分别为：

$$
\begin{aligned}
P &= \frac{TP}{TP+FP} \\
R &= \frac{TP}{TP+FN}
\end{aligned} \tag{24}
$$

为了比较不同分类器性能的优劣，人们引入 $F1$ 值来进行综合衡量精确率与召回率. $F1$ 是 P, R 的调和平均：

$$
\begin{aligned}
\frac{2}{F1} &= \frac{1}{P} + \frac{1}{R} \\
F1 &= \frac{2TP}{2TP+FP+FN}
\end{aligned} \tag{25}
$$

而分类准确率 (accuracy) 指测试集中样本正确分类数与样本总数之比，即

$$
ACC = \frac{TP+TN}{TP+FP+FN+TN} \tag{26}
$$