

Chapter 1 猜论

1.1 引言

机器学习：对计算机一部数据进行学习，然后对另外一些数据进行预测和判断。

1.2 基本术语

- 数据集：一组记录的集合。
- 例/样本：每条记录是关于一个事件或对象的描述。
- 属性/特征：反映事件或对象在某方面的表现或性质的事项。
- 属性值：属性上的取值。
- 属性空间/样本空间/输入空间：属性张成的空间。
- 特征向量：示例可以用坐标向量表示。每个示例由 d 个属性描述， d 称为样本的维数。
- 学习/训练：从数据中学会模型的过程。训练中使用的数据叫训练数据，每个样本叫训练样本，训练样本组成的集合叫训练集。
- 标记：关于示例结果的信息。
- 样例：拥有了标记信息的示例。

所有标记的集合称为标记空间或输出空间。

- 测试：学得模型后使用其进行预测的过程。测试样本
- 分类：预测的是离散值。
- 回归：预测连续值。
- 二分类：只涉及两个类别。多分类：涉及多个类别。

正类 反类

· 聚类：了解数据的内在规律，为更深入地分析数据建立基础。

· 监督学习：训练数据拥有标记信息 \rightarrow 分类和回归 } 半监督学习
无监督学习 \rightarrow 聚类

· 泛化：学得模型适用于新样本的能力。

1.3 假设空间

· 学习过程 \rightarrow 在所有假设组成的空间上进行搜索的过程。

假设空间大小：设有 m 个属性： d_1, d_2, \dots, d_m ，每个属性有 n_1, n_2, \dots, n_m 个可能的取值，则

$$\text{Size} = \left[\prod_{i=1}^m (n_i + 1) \right] + 1 \quad \text{加1加的是中}$$

· 版本空间：与训练集一致的假设集合。

1.4 / 1.5 约偏好

- 归纳偏好：ml-algorithm 在学习过程中对某种类型假设的偏好.
- 奥卡姆剃刀原则：若多个假设与观察一致，则选最简单的那个.
- 学习算法的归纳偏好是否与问题本身匹配，大多数时候直接决定了算法能否取得好的性能.
- NFL定理 (No Free Lunch)：一个算法如果在某些问题上比另一个算法好，必然存在另一些问题，反之亦然.

1.5 发展历程

1956年达特茅斯会议标志着人工智能的诞生

推理期 → 知识期 → 学习期

符号主义 | 连接主义 | 统计学习

Chapter 2 模型评估与选择

2.1 经验误差与过拟合

· 误差：预测输出与样本的真实输出之间的差异.

经验误差 / 训练误差
测试误差
泛化误差 (新样本上)

· 错误率：错分样本的占比.



希望泛化误差小的学习器.

经验误差不是越小越好，太小时会出现过拟合！泛化性能下降.

· 过拟合：学习能力过于强大，不可避免，只能缓解.

· 欠拟合：学习能力不足，加大学习.

2.2 评估方法

通过测试集来测试学习器对新样本的判别能力，然后以测试集上的“测试误差”作为“泛化误差”的近似.

· 测试样本：①从样本真实分布中独立同分布采样.

②与训练集尽可能互斥.

对一个包含 m 个样例的数据集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$.

2.2.1 留出法

直接将 D 划分为两个互斥集合, $D = S \cup T$, $S \cap T = \emptyset$. S 与 T 数据分布尽可能一致。(分层采样) 多次随机划分、重复实验取平均值. 测试集不能太大、不能太小.

(1/5 ~ 1/3)

2.2.2 交叉验证法

将数据集 D 抽样划分为 k 个大小相似的互斥子集, 每次用 $k-1$ 个子集的并集作为训练集, 余下的作为测试集, 最终返回 k 个测试结果的均值, k 常取 10.

将数据集划分为 k 个子集有多种方法, k 折交叉验证法通常随机使用不同的划分重复 k 次, 最终的评估结果是这 k 次 k 折交叉验证结果的均值.

假设数据集中包含 m 个样本, 若令 $k=m$, 则得到留一法:

① 不受随机划分方式影响.

② 结果准确.

③ 当数据集很大时, 计算开销难以忍受.

2.2.3 自助法

以自助采样法为基础, 对数据集 D 有放回采样 m 次得到训练集 D' , $D \setminus D'$ 作为测试集.

① 实际模型与预期模型都使用了 m 个训练样本

② 约有 $1/3$ 的样本没在训练集中出现. (包外估计)

$$\lim_{m \rightarrow \infty} \left(1 - \frac{1}{m}\right)^m \rightarrow \frac{1}{e} \approx 0.368$$

③ 能产生多个不同的训练集, 对集成学习有很大的好处.

④ 自助法在数据集较小、难以有效划分训练/测试集时很有用; 由于改变了

数据分布可能会引入估计偏差, 在数据量足够时, 留出法和交叉验证更常用.

2.2.4 调参与最终模型

算法的参数一般由人工设定, 称为超参数.

模型的参数一般由学习确定.

算法参数选定后, 要用“训练集 + 验证集”重新训练最终模型.

· 验证集: 评估测试的数据集

2.3 性能度量

用来衡量模型泛化能力的评价标准.

性能度量反映了任务需求，使用不同的性能度量往往会导致不同的评判结果。

2.3.1 错误率和精度

$$E(f; D) = \frac{1}{m} \sum_{i=1}^m I(f(x_i) \neq y_i)$$

f 是学习器, $f(x_i)$ 是预测结果, y_i 是真实标记
 I 是指示函数, D 是数据集。

$$\text{acc}(f; D) = \frac{1}{m} \sum_{i=1}^m I(f(x_i) = y_i) = 1 - E(f; D)$$

2.3.2 查准率、查全率与 F_1

P 准确度 R 召回率

混淆矩阵

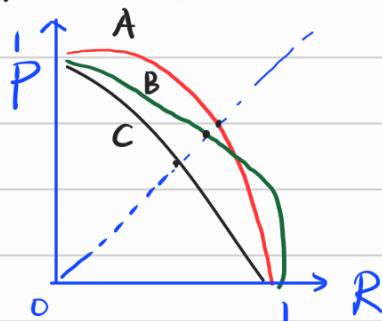
		预测结果	
真实情况		正例	反例
正例	TP (真正例)	FN (假反例)	
	FP (假正例)	TN (真反例)	

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

① P-R 相互矛盾

② P-R 曲线：



学习器 A, B, C 的优劣：

(i) 包住？

(ii) 围成的面积

(iii) 平衡点 BEP where $P=R$.



$$F_1 = \frac{2PR}{P+R} \quad (\text{调和平均})$$

$$= \frac{2TP}{\text{样例总数} + TP - TN}$$

④ 一般形式 (考虑对 P 和 R 的偏好):

$$F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P + R} \Leftrightarrow \frac{1}{F_\beta} = \frac{1}{1+\beta^2} \left(\frac{1}{P} + \frac{\beta^2}{R} \right).$$

$$\begin{cases} \beta = 1, \text{ 即 } F_1 \\ \beta > 1, \text{ R 更重要} \\ \beta < 1, \text{ P 更重要} \end{cases}$$

③ 很多时候有多个二分类混淆矩阵，在 n 个 confusion matrix 上：

宏：分别计算在confusion matrix上的
R和P，记 $(P_1, R_1), (P_2, R_2) \dots$
 (P_n, R_n) ，求 mean.

$$\text{macro-P} = \frac{1}{n} \sum_{i=1}^n P_i$$

$$\text{macro-R} = \frac{1}{n} \sum_{i=1}^n R_i$$

$$\text{macro-F}_1 = \frac{2 \text{macro-P} \times \text{macro-R}}{\text{macro-P} + \text{macro-R}}$$

微：将 confusion matrix 对应元素平均。
得到 TP, TN, FP, FN 的 mean.

$$\text{micro-P} = \frac{\bar{TP}}{\bar{TP} + \bar{FP}}$$

$$\text{micro-R} = \frac{\bar{TP}}{\bar{TP} + \bar{FN}}$$

$$\text{micro-F}_1 = \frac{2 \text{micro-P} \times \text{micro-R}}{\text{micro-P} + \text{micro-R}}$$

2.3.3 ROC 与 AUC

ROC: Receiver Operating Characteristic. 受试者工作特征

AUC: Area Under ROC Curve. ROC 曲线下面积

很多学习器是为测试样本产生一个实值或概率预测，将其与分类阈值 threshold 作比较，大于 threshold 为正类，反之为反类。

纵轴真正例率: $TPR = \frac{TP}{TP+FN} = R$

横轴假正例率: $FPR = \frac{FP}{TN+FP}$

绘图过程。给定 m^+ 个正例, m^- 个反例, 首先根据预测排序, 然后将分类阈值设为最大, 即把所有样例均预测为反类, 此时 $(0, 0)$. 然后将分类阈值依次设为每个样例预测值, 依次将每个样例划分为正类, 设前一个坐标为 (x, y) , 若前若为真正例, 坐标为 $(x, y + \frac{1}{m^+})$; 若为假正例, 坐标为 $(x + \frac{1}{m^-}, y)$.

$$AUC = \frac{1}{2} \sum_{i=1}^{m-1} (x_{i+1} - x_i) (y_i + y_{i+1})$$

提升: $\text{lrank} = \frac{1}{m+m^-} \sum_{x^+ \in D^+} \sum_{x^- \in D^-} (I(f(x^+) < f(x^-)) + \frac{1}{2} I(f(x^+) = f(x^-)))$

$$AUC = 1 - \text{lrank}$$

2.3.4 代价敏感错误率与代价曲线

现实中, 不同类型错误所造成的后果不同. 为权衡不同类型错误所造成的不同损失, 可将错误赋予“非均等代价”. 以二分类为例, 设这个代价矩阵.

Cost matrix:

真实	预测	
	第0类	第1类
第0类	0	cost_01
第1类	cost_10	0

cost-sensitive error:

$$E(f; D; cost) = \frac{1}{m} \left(\sum_{x_i \in D^+} I(f(x_i) \neq y_i) \times cost_01 + \sum_{x_i \in D^-} I(f(x_i) \neq y_i) \times cost_10 \right)$$

Cost curve:

横轴是 [0,1] 的正例概率代价:

$$P(+|cost) = \frac{p \times cost_01}{p \times cost_01 + (1-p) \times cost_10}$$

↑ 客户代价
↑ 最大代价

是样例正例概率

纵轴是 [0,1] 的代价.

$$cost_{norm} = \frac{FNR \times p \times cost_01 + FPR \times (1-p) \times cost_10}{p \times cost_01 + (1-p) \times cost_10}$$

↑ Ecost
↑ max(Ecost)

$$FNR = 1 - TPR$$

代价曲线图的绘制:

- ① ROC 上的每个点对应了代价曲线上的一条线段
- ② 设 ROC 曲线上的点的坐标为 (FPR, TPR) , 则可计算出 FNR .
- ③ 然后在代价平面画一条从 $(0, FPR)$ 到 $(0, FNR)$ 的线段, 线段下的面积即表示了该条件下的期望总体代价.
- ④ 如此将 ROC 上的每个点转化为代价平面上的一条线段, 然后取所有线段的下界, 围成的面积即为所有条件下学习器的期望总体代价.

2.4 比较检验

为什么机器学习中性能比较非常复杂?

两个学习器不能直接比吗?

- ① 测试性能并不等于泛化性能
- ② 测试性能随着测试集的变化而变化.
- ③ 很多机器学习算法本身有一定的随机性.

直接选择相应方法在相对度量下比大小的方法不可取!

假设检验为泛化性能比较提供了重要依据.

假设检验逻辑如下:

① 原假设与备选假设.

② 证据: 在原假设前提下计算的机率 P

③ 判断标准: α 显著水平

④ 如果 $P \leq \alpha$, 拒绝原假设, 备选假设成立; $P > \alpha$, 原假设成立.

假设检验中“假设”是对学习器泛化错误率分布的判断或猜想.

泛化错误率 $\varepsilon = \varepsilon_0$ 不知道, 要求 } \Rightarrow 用 $\hat{\varepsilon}$ 近似等于 ε_0 . 但
测试错误率 $\hat{\varepsilon}$ 计算得到 } 为结果.

泛化错误率为 ε 的学习器被测试错误率为 $\hat{\varepsilon}$ 的概率:

$$P(\hat{\varepsilon}, \varepsilon) = C_m^{m \times \hat{\varepsilon}} \cdot \varepsilon^{\hat{\varepsilon} \times m} \cdot (1 - \varepsilon)^{m - \hat{\varepsilon} \times m}$$

两学习器比较:

交叉验证法检验 (k 折交叉验证)

McNemar 检验 (基于列联表, 卡方检验).

多学习器比较:

Friedman 检验: 基于序值, F 检验; 判断是否都相同.

Nemenyi 后续检验: 基于序值, 进一步判断两两差异.

2.5 偏差与方差.

$$E[f; D] = b \cdot \text{as}^2(x) + \text{var}(x) + \varepsilon^2$$

$$= (\bar{f}(x_i) - y)^2 + E_D[(f(x_i; D) - \bar{f}(x_i))^2] + E_D[(y_D - y)^2]$$

泛化性能是由学习算法的能力、数据的充分性以及学习任务本身的难度共同决定

泛化误差可分解为偏差方差与噪声之和.

· 偏差度量了学习算法期望预测与真实结果的偏离程度, 即刻画了

学习算法本身拟合能力.

- 方差度量了同样大小训练集的变动所导致的学习性能的变化；即刻画了数据扰动造成的影响。
- 噪声表达了在当前任务上任何学习算法所能达到的期望泛化误差的下界。即刻画了学习问题本身的难度。

给定学习任务为了取得好的泛化性能，需要使偏差小（充分拟合数据）而且方差较小（减小数据扰动产生的影响）。

偏差-方差窘境：

在训练不足时，学习器拟合能力不强，训练数据的扰动不足以使学习器的拟合能力产生显著变化，此时偏差主导 泛化错误率。

随着训练程度加深，学习器拟合能力逐渐增强，方差逐渐主导泛化错误率。

训练充足后，学习器的拟合能力非常强，训练数据的轻微扰动都会导致学习器的显著变化，若训练数据自身非全局特性被学到则会过拟合，方差占主导。

Chapter 3 线性模型

3.1 基本形式

多个属性描述示例 $x = (x_1, x_2, \dots, x_d)$ 其中 x_i 是 x 在第 i 个属性取值。

linear model 通过属性的线性组合预测函数。

$$f(x) = w_1 x_1 + w_2 x_2 + \dots + w_d x_d + b$$

向量形式： $f(x) = w^T x + b$ $w = (w_1, w_2, \dots, w_d)$

w, b 定得后，模型确定。

优点：
① 形式简单，易于建模
② 可解译性
③ 非线性模型基础：引入层以结构或高维映射。

3.2 线性回归

数据集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ 其中 $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$, $y_i \in \mathbb{R}$

有序关系：连续化为离散值，如高、长、矮 {1, 0}

无序关系：k 个属性值，转化为 k 维向量 (0, 0, 1), (0, 1, 0), (1, 0, 0)

线性回归目的：获得一个线性模型以及可能准确地预测实值输出特征。

$$f(x_i) = w x_i + b \quad \text{s.t. } f(x_i) \approx y_i$$

(1) 单一属性线性回归。Q. 如何确定 w, b ?

最小二乘法：令均方误差最小化。

$$(w^*, b^*) = \underset{(w, b)}{\operatorname{arg\min}} \sum_{i=1}^m (f(x_i) - y_i)^2 = \underset{(w, b)}{\operatorname{arg\min}} \sum_{i=1}^m (y_i - w x_i - b)^2$$

对 $E(w, b) = \sum_{i=1}^m (y_i - w x_i - b)^2$ 进行最小二乘参数估计。

$$\begin{cases} \frac{\partial E(w, b)}{\partial w} = 2 \left(w \sum_{i=1}^m x_i^2 - \sum_{i=1}^m (y_i - b) x_i \right) \\ \frac{\partial E(w, b)}{\partial b} = 2 \left(m b - \sum_{i=1}^m (y_i - w x_i) \right) \end{cases}$$

↓ 全导数法得

$$w = \frac{\sum_{i=1}^m y_i (x_i - \bar{x})}{\sum_{i=1}^m x_i^2 - \frac{1}{m} \left(\sum_{i=1}^m x_i \right)^2} \quad b = \frac{1}{m} \sum_{i=1}^m (y_i - w x_i) \quad \bar{x} = \frac{1}{m} \sum_{i=1}^m x_i$$

(2) 多元线性回归。 $f(x_i) = w^T x_i + b = (w^T, b) \begin{pmatrix} x_i \\ 1 \end{pmatrix}$, s.t. $f(x_i) \approx y_i$.

求 w 和 b 吸收为一个 $\hat{w} = (w, b)$

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} & 1 \\ x_{21} & x_{22} & \dots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{md} & 1 \end{pmatrix} = \begin{pmatrix} x_1^T & 1 \\ x_2^T & 1 \\ \vdots & \vdots \\ x_m^T & 1 \end{pmatrix} \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$$

$$\hat{w}^* = \underset{\hat{w}}{\operatorname{arg\min}} (y - X \hat{w})^T (y - X \hat{w}), \text{ 令 } E_{\hat{w}} = (y - X \hat{w})^T (y - X \hat{w}), \text{ 对 } \hat{w}$$

求导得：

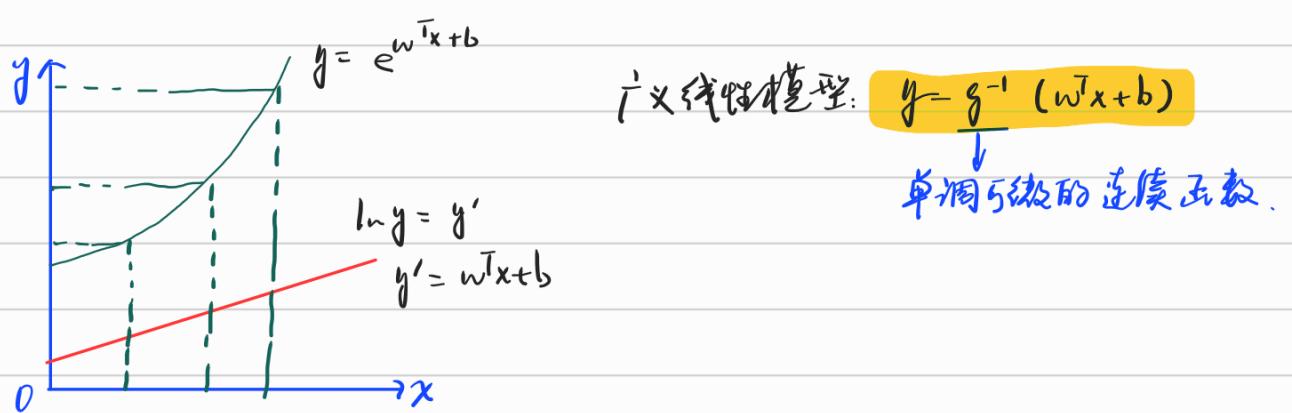
$$\frac{\partial E_{\hat{w}}}{\partial \hat{w}} = 2 X^T (X \hat{w} - y)$$

若 $X^T X$ 为满秩矩阵或正定矩阵，则 $\hat{w}^* = (X^T X)^{-1} X^T y$

线性回归模型为： $f(\hat{x}_i) = \hat{x}_i^T (X^T X)^{-1} X^T y$

若 $X^T X$ 不是满秩矩阵，则可解出多个 \hat{w} ，根据岭归偏好选择解，引入正则化。

(3) 对数线性回归： $\ln y = w^T x + b \Leftrightarrow e^{w^T x + b}$ 适应 y 。



3.3 对数几率回归.

若某二分类任务，输出标记 $y \in \{0, 1\}$.

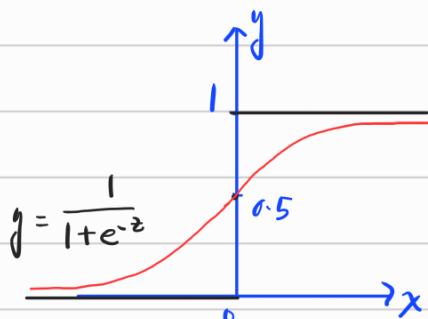
线性回归预测值 $z = w^T x + b$ 只需将 z 转换成 $0/1$.

最理想的是“单位阶跃函数”. 但是性质不好(不连续).

$$y = \begin{cases} 0, & z < 0 \\ 0.5, & z = 0 \\ 1, & z > 0 \end{cases}$$

替代函数：对数几率函数:

$$y = \frac{1}{1 + e^{-z}}$$



$$\textcircled{\text{1}} \quad y = \frac{1}{1 + e^{-(w^T x + b)}}$$

$$\textcircled{\text{2}} \quad \ln \frac{y}{1-y} = w^T x + b$$

y 视为样本正例的可能性，则 $1-y$ 是其反例的可能性.

几率反映了 x 视为正例的相对可能性. $\frac{y}{1-y}$. 对数几率: $\ln \frac{y}{1-y}$.

用线性回归模型预测结果去逼近真实标记的对数几率叫对数几率回归.

注意: 这是分离学习算法!

极大似然法: 将 y 视为类后验概率估计 $P(y=1|x)$

$$\ln \frac{y}{1-y} = \ln \frac{P(y=1|x)}{P(y=0|x)} = w^T x + b$$

$$\left\{ \begin{array}{l} P(y=1|x) = \frac{e^{w^T x + b}}{1 + e^{w^T x + b}} \\ P(y=0|x) = \frac{1}{1 + e^{w^T x + b}} \end{array} \right.$$

通过极大似然法估计 w 和 b , 给定数据集 $\{(x_i, y_i)\}_{i=1}^m$, 最大化对数似然:

$$l(w, b) = \sum_{i=1}^m \ln p(y_i | x_i; w, b) \quad (1)$$

转化为最大化负对数似然函数求解:

$$\text{令 } \beta = (w; b), \hat{x} = (x; 1) \Rightarrow w^T x + b \text{ 可写为 } \beta^T \hat{x}.$$

$$\text{再令 } p_1(\hat{x}_i; \beta) = p(y=1 | \hat{x}_i; \beta) = \frac{e^{w^T \hat{x}_i + b}}{1 + e^{w^T \hat{x}_i + b}}$$

$$p_0(\hat{x}_i; \beta) = p(y=0 | \hat{x}_i; \beta) = \frac{1}{1 + e^{w^T \hat{x}_i + b}} = 1 - p_1(\hat{x}_i; \beta)$$

则似然项可重写为 $p(y_i | x_i; w, b) = y_i p_1(\hat{x}_i; \beta) + (1-y_i) p_0(\hat{x}_i; \beta)$
于是最大化似然函数等价形式为最大化:

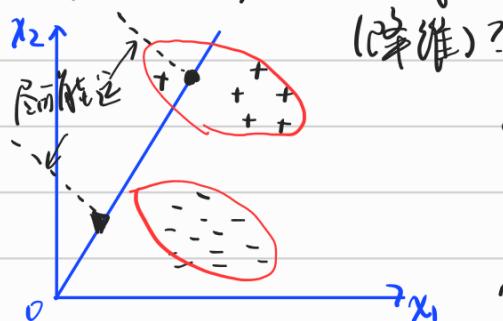
$$l(\beta) = \sum_{i=1}^m (-y_i \beta^T \hat{x}_i + \ln(1 + e^{\beta^T \hat{x}_i}))$$

求解得 $\beta^* = \underset{\beta}{\operatorname{argmin}} l(\beta)$.

牛顿法第 $t+1$ 轮迭代解的更新公式: $\beta^{t+1} = \beta^t - \left(\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial l(\beta)}{\partial \beta}$
(梯度下降法/牛顿法).

3.4 线性判别分析: Linear Discriminant Analysis (LDA)

思想: 投影到某直线上, 同类样例尽可能近, 异类样例尽可能远.



数据集 $D = \{(x_i, y_i)\}_{i=1}^m, y_i \in \{0, 1\}$

第 i 类样例的集合 X_i

第 i 类样例的均值向量: μ_i

第 i 类样例的协方差矩阵 Σ_i

两类样本的中心在直线上的投影: $w^T \mu_0$ 和 $w^T \mu_1$.

两类样本的协方差: $w^T \Sigma_0 w$ 和 $w^T \Sigma_1 w$

①同类样例的投影点尽可能近: $w^T \Sigma_0 w + w^T \Sigma_1 w$ 尽可能小

②异类样例的投影点尽可能远: $\|w^T \mu_0 - w^T \mu_1\|_2^2$ 尽可能大

同时考虑②得最大化目标:

$$J = \frac{\|w^T \mu_0 - w^T \mu_1\|_2^2}{w^T \Sigma_0 w + w^T \Sigma_1 w} = \frac{w^T (\mu_0 - \mu_1) (\mu_0 - \mu_1)^T w}{w^T (\Sigma_0 + \Sigma_1) w}$$

类内散度矩阵:

$$S_w = \Sigma_0 + \Sigma_1 = \sum_{x \in X_0} (x - \mu_0)(x - \mu_0)^T + \sum_{x \in X_1} (x - \mu_1)(x - \mu_1)^T$$

车间散度矩阵:

$$S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T$$

Then $\frac{1}{2} w^T S_w w = 1$, 最大化广义特征值.

如何求 w ?

$\sum w^T S_w w = 1$, 最大化 J 等价于 $\min_w -w^T S_b w$.

计算: 使用拉格朗日乘子法, 有 $S_b w = \lambda S_w w$

$$\text{由 } S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T \text{ 有 } S_b w = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T w$$

注意到 $(\mu_0 - \mu_1)^T w$ 是标量, 令其等于入. 于是:

$$w = S_w^{-1} (\mu_0 - \mu_1)$$

$$S_w = U \Sigma V^T \text{ 奇异值分解. } S_w^{-1} = V \Sigma^{-1} U^T$$

将 LDA 推广到多分类中. 假定有 N 个类.

$$\text{全局散度矩阵 } S_t = S_b + S_w = \sum_{i=1}^m (x_i - \mu) (x_i - \mu)^T$$

$$S_w = \sum_{i=1}^N S_{wi} \quad S_{wi} = \sum_{x \in X_i} (x - \mu_i) (x - \mu_i)^T$$

$$S_b = S_t - S_w = \sum_{i=1}^N m_i (\mu_i - \mu) (\mu_i - \mu)^T$$

常 KKT 梯度目标:

$$\max_w \frac{\text{tr}(w^T S_b w)}{\text{tr}(w^T S_w w)} \quad w \in \mathbb{R}^{d \times (N-1)}$$

w 是 $S_w^{-1} S_b$ 的 d' ($d' \leq N-1$) 个最大非零广义特征值对应的特征向量组成的矩阵.

LDA 被视为一种监督降维技术.

3.5 多分类学习

二分类方法推广到多分类. 对问题进行拆分.

拆分策略 $\left\{ \begin{array}{l} \text{一对一: OvO} \\ \text{一对其余: OvR} \\ \text{多对多: MvM} \end{array} \right.$

(1) OvO: N 个类两两面对. $N(N-1)/2$ 个二分类任务.

(2) OvR: 某一个作为正例, 其他反例. N 个二分类任务.

$\left\{ \begin{array}{l} \text{OvO 在训练阶段、测试时间开销比 OvR 大.} \\ \text{OvO 每次训练时间比 OvR 小.} \end{array} \right.$

性能两个差不多.

(3) MvM: 每次将若干个类作为正类, 若干个其他类作为反类.

一种技术: 纠错输出码 (ECOC).

① 编码: 对 N 个类别进行 M 次划分, 每次划分一部分正, 一部分反, 产生 M 个训练集, 训练 M 个分类器.

② 解码: M 个分类器分别对测试样本预测, 预测组成编码, 将预测编码和类别编码比较, 返回距离最小的为最终预测.

一般而言, ECOC 编码越长, 错误能力越强.

同等长度编码. 理论上, 任意两类之间编码距离越远, 错误能力越强.

3.6 类不平衡问题.

不同类别的样本比例相差很大.

基本思路: 若 $\frac{y}{1-y} > 1$ 则预测为正例.

\Rightarrow 若 $\frac{y}{1-y} > \frac{m^+}{m^-}$ 则预测为正例.

基本策略: 再缩放: $\frac{y'}{1-y'} = \frac{y}{1-y} \times \frac{m^-}{m^+}$

三类方法: ① **过采样**: 增加一些正例. SMOTE

② **欠采样**: 去除一些反例.

③ **阈值移动**.



3.7 总结

古今模型优化目标:

{最小二乘法, 最小化均方误差 MSE

{对数几率回归: 最大化样本分布似然

LDA: 投影空间内最小(大)化类内(间)散度. 矩阵论、广义逆矩阵

参数优化方法:

线性代数

凸优化梯度下降, 牛顿法

Chapter 4 决策树

4.1 基本流程

决策树基于“树”结构进行决策.

① 每个内部结点对应于某个属性上的测试

② 每个结点对应于该测试的一种可能结果(一个取值).

③ 每个叶结点对应一个预测结果.

策略：先分后合，自底至叶的逆向过程.

三种停止条件：

① 当前结点包含样本属同一类别，无需划分

② 当前属性集为空，所有样本在所有属性上取值相同，无法划分.

③ 当前结点包含的样本集合为空，不能划分.

决策树学习的目的是为了产生一棵泛化能力强，即处理未见示例能力强的决策树.

4.2 划分选择

4.2.1 信息增益 (ID3)

(假定当前样本集合 D 中第 k 类样本所占的比例为 p_k , 则 D 的信息熵：

$$Ent(D) = - \sum_{k=1}^{|Y|} p_k \log_2 p_k$$

$Ent(D)$ 的值越小，则 D 的纯度越高.

离散属性 a 的取值： $\{a^1, a^2, \dots, a^V\}$. D_v : D 中在 a 上取值 = a^v 的样本集合.

以属性 a 对数据集 D 进行划分所获得的信息增益为：

$$Gain(D, a) = Ent(D) - \sum_{v=1}^V \frac{|D^v|}{|D|} Ent(D^v)$$

↑
划分前信息熵 ↓
第 v 个分支权重 ↑
划分后

信息增益越大，纯度提升越大.

信息增益对可取值数目较多的属性有所偏好

4.2.2 增益率 (C4.5)

$$Gain-ratio(D, a) = \frac{Gain(D, a)}{IV(a)}$$

$$IV(a) = - \sum_{v=1}^V \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|}$$

属性 a 的可能取值数目越多，则 $IV(a)$ 的值通常越大.

启发式：先从候选划分属性中找出信息增益高于平均水平的，再从中选取增益率最高的.

4.2.3 基尼指数 (CART)

$$Gini(D) = \sum_{R=1}^{|Y|} \sum_{k \neq k'} P_R P_{R'} = 1 - \sum_{k=1}^{|Y|} P_k^2$$

反映了从D中随机抽取两个样本，其类别标记不一致的概率.

$Gini(D)$ 越小，数据集D纯度越高.

属性a的基尼指数： $Gini_index(D, a) = \sum_{v=1}^{|D'|} \frac{|D'_v|}{|D|} Gini(D'_v)$

在候选属性集合中，选取那个使划分后基尼指数最小的属性.

$$a^* = \operatorname{argmin}_{a \in A} Gini_index(D, a).$$

4.3 剪枝处理

剪枝是用来解决“过拟合”，比如分支过多，把训练集自身的一些特点当作所有数据的一般性质.

基本策略：

① 预剪枝：提前终止某些分支的生长.

② 后剪枝：生成一棵完全树，再回头剪枝.

4.3.1 预剪枝

决策树生成过程中，对每个结点在划分前先进行估计。若当前结点的划分不能带来决策树泛化性能提升，则停止划分并将当前结点记为叶结点，其类别标记为训练样例数最多的类别.

优点：①降低过拟合风险.

②显著减少训练时间和测试时间开销.

缺点：欠拟合风险，贪心！

4.3.2 后剪枝

先从训练集生成一棵完整的决策树，然后自底向上地对非叶结点进行考察，若将该结点对应的子树替换为叶结点能带来决策树泛化性能提升，则将该子树替换为叶结点.

优点：欠拟合风险小，泛化性能往往优于预剪枝决策树.

缺点：训练开销大.

4.4 连续与缺失值

4.4.1 连续值处理

连续属性离散化：二分法.

N个属性值可形成N-1个候选划分，然后即可将它们去做N-1个离散属性值处理.

给定样本集 D , 连续属性 a , 假设 a 在 D 上出现了 n 个不同的取值, 从小到大排序:

$$\{a^1, a^2, \dots, a^i, a^{i+1}, \dots, a^n\}$$

$\underbrace{\quad}_{D_t^-} \quad \underbrace{\quad}_{t \text{ 划分点}} \quad \underbrace{\quad}_{D_t^+}$

若包含 $n-1$ 个元素的候选划分集合:

$$T_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid 1 \leq i \leq n-1 \right\}$$

即把区间 $[a^i, a^{i+1}]$ 的中位点 $\frac{a^i + a^{i+1}}{2}$ 作为候选划分.

$$Gain(D, a) = \max_{t \in T_a} Gain(D, a, t)$$

$$= \max_{t \in T_a} Ent(D) - \sum_{A \in \{+, -\}} \frac{|D_A|}{|D|} Ent(D_A)$$

选择使 $Gain(D, a, t)$ 最大的才作为划分点.

与离散属性不同, 若当前结点划分属性为连续属性, 该属性还可作为其后代结点的划分属性.

4.4.2 缺失值处理

样本赋权, 权重划分

\hat{D} 表示 D 中在属性 a 上没有缺失值的样本子集, \hat{D}^v 表示 D 中在属性 a 上取值为 a^v 的样本子集, B_k 表示 D 中属于第 k 类的样本子集. 为每个样本 x 赋予一个权重 w_x , 并定义:

① 无缺失值样本所占的比例:

$$p = \frac{\sum_{x \in \hat{D}} w_x}{\sum_{x \in D} w_x}$$

② 无缺失值样本中第 k 类所占比例:

$$\hat{p}_k = \frac{\sum_{x \in \hat{D}_k} w_x}{\sum_{x \in \hat{D}} w_x} \quad 1 \leq k \leq |Y|$$

③ 无缺失值样本中在属性 a 上取值为 a^v 的样本所占比例:

$$\hat{r}_v = \frac{\sum_{x \in \hat{D}^v} w_x}{\sum_{x \in \hat{D}} w_x} \quad 1 \leq v \leq V$$

例: $Gain(D, a) = p \times Gain(\hat{D}, a)$

$$= p \times \left(Ent(\hat{D}) - \sum_{v=1}^V \hat{r}_v Ent(\hat{D}^v) \right)$$

其中 $H(D) = - \sum_{k=1}^{|Y|} p_k \log_2 \tilde{p}_k$

给定划分属性，若样本在该属性上的值缺失，如何进行划分？

①若样本 x 在划分属性 a 上的取值已知，则将 x 划入与其取值对应的孩子结点，且样本权值在子结点中保持为 w_x 。

②若样本 x 在划分属性 a 上的取值未知，则将 x 同时划入所有子结点，且样本权值在与属性 a^v 对应的子结点中调整为 $F_v \cdot w_x$ 。

4.5 多变量决策树

单变量决策树分类边界：轴平行。

多变量：非叶节点是属性的线性组合。每个非叶节点是一个形如

$$\sum_{i=1}^d w_i a_i = t$$

的线性分类器。 w_i 是 a_i 的权值。

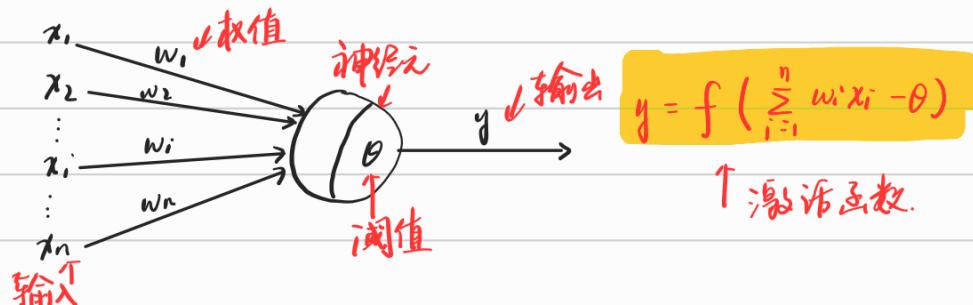
Chapter 5 神经网络

5.1 神经元模型 — 神经网络基本概念

神经网络是由具有适应性的简单单元组成的广泛并行互连的网络，它的组织能够模拟生物神经系统对真实世界物体所作出的交互反应。

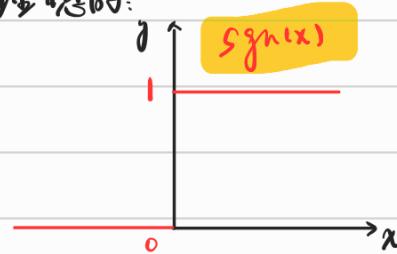
机器学习中谈论神经网络指“神经网络层”。

1943 年, McCulloch 和 Pitts: M-P 神经元模型



激活函数。

理想的:

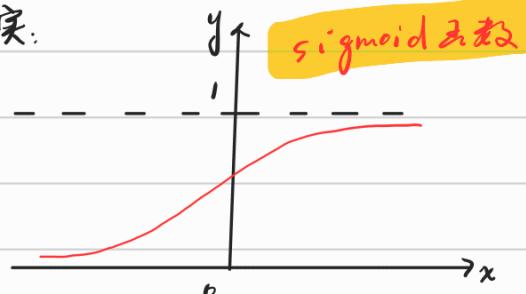


$$\text{sgn}(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

1 激活, 0 抑制。

但不连续不光滑。

现实:



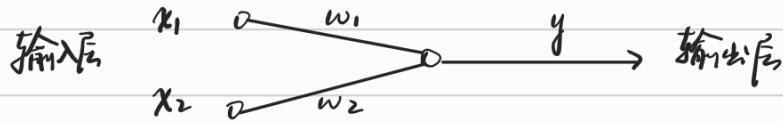
$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

0 与 1 之间，可微连续。

$$f(x) = f(x) (1 - f(x))$$

5.2 感知机与多层网络

Perceptron 由两层神经元组成，输入层接受外界输入信号传递给输出层，输出层是 M-P 神经元，又叫阈值逻辑单元。



感知机可实现与、或、非运算。

给定数据集，权重 $w_i (i \in [n])$ 及值可通过学习得到。

对训练样例 (x, y) ，若当前感知机的输出为 \hat{y} ，则感知机权重调整规则为：

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i = \eta (y - \hat{y}) x_i$$

$\eta \in (0, 1)$ 叫学习率

单层感知机只能解决线性可分问题，非线性“或”“解决不了，多层感知机可以。输入层与输出层之间的一层神经元，被称为隐层，隐层与输出层神经元都是具有激活函数的功能神经元。function unit

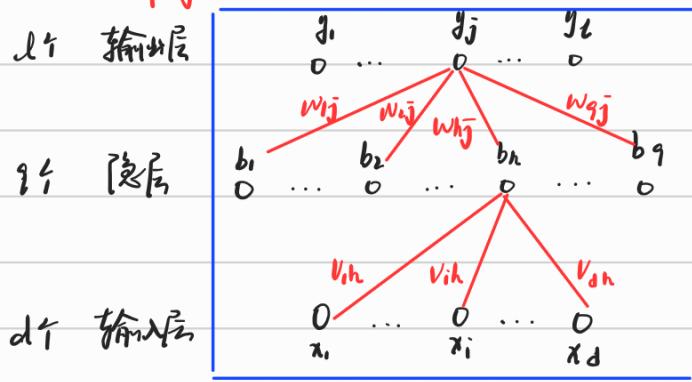
多层网络包含隐层。

前馈网络：神经元之间不存在同层连接以及跨层连接。

多层前馈网络有强大的表示能力，但是如何设置隐层神经元数是未决问题，实际常用“试错法”。

5.3 BP 算法

BackPropagation：误差逆传播。



第 j 个输出神经元的输入：

$$\beta_j = \sum_{h=1}^q w_{jh} b_h$$

第 h 个隐层神经元输入：

$$\alpha_h = \sum_{i=1}^d v_{ih} x_i$$

训练例 (x_k, y_k) ，假设输出为 $\hat{y}_k = (\hat{y}_1^k, \hat{y}_2^k, \dots, \hat{y}_l^k)$ ， $\hat{y}_j^k = f(\beta_j - \theta_j)$

则网络在 (x_k, y_k) 上的 MSE 为： $E_k = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j^k - y_j^k)^2$

需要多少个参数？ $d \times q$ 个权值， $q \times l$ 个权值， q 个偏置， l 个偏置。

$$d \times q + q \times l + q + l = (d+1+l)q + 1 \text{ 个参数。}$$

任意参数更新公式: $v \leftarrow v + \Delta v$.

BP 基于梯度下降策略, 以目标负梯度方向对参数调整. 以 w_{kj} 为例.

对误差 E_k , 给定学习率 η , 有:

$$\Delta w_{kj} = -\eta \frac{\partial E_k}{\partial w_{kj}}$$

$$\frac{\partial E_k}{\partial w_{kj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{kj}}$$

$$= (\hat{y}_j^k - y_j^k) \cdot \hat{y}_j^k (1 - \hat{y}_j^k) \cdot b_h \quad \text{即 } \Delta w_{kj} = \eta g_j b_h.$$

输出层神经元梯度项

$$\text{令 } \delta_j = -\frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j}$$

同理:

$$\Delta \theta_j = -\eta g_j$$

$$\text{其中 } e_h = -\frac{\partial E_k}{\partial b_h} \cdot \frac{\partial b_h}{\partial x_i} \quad \text{顶层神经元梯度项}$$

$$\Delta v_{ih} = \eta e_h x_i$$

$$= b_h (1 - b_h) \sum_{j=1}^l w_{hj} g_j$$

$$\Delta r_h = -\eta e_h$$

BP 目标是最小化训练集上的累积误差 $E = \frac{1}{m} \sum_{k=1}^m E_k$.

{
标准BP: 每次针对单个训练样例更新权值与阈值. 参数更新频繁, 不同样例可能抵消, 需要多次迭代.

累积BP: 其优化目标是最小化整个训练集上的累积误差, 读取整个训练集一遍才对参数进行更新, 参数更新频率较低.

如何缓解 BP 网络过拟合?

① 早停: i) 若训练误差连续 a 步的变化小于 b

ii) 使用验证集; 若训练误差小, 验证误差大

② 正则化: 在误差目标函数中增加一项描述网络复杂度.

$$E = \lambda \frac{1}{m} \sum_{k=1}^m E_k + (1-\lambda) \sum_i w_i^2$$

5.4 全局最小与局部极小.

神经网络的训练过程可看作一个参数优化过程: 在参数空间中寻找一组最优参数使得误差最小.

“跳出”局部极小的常见策略: 不同的初始化参数、随机过大、随机扰动演化算法……

5.5 其他神经网络.

- 比底层、识别层、识别阈值重>置模块。**
- **RBF**: 分类任务中除BP外最常用，使用径向基函数。单隐层。
 - **ART**: 竞争学习代表。(无监督学习策略)。胜者通吃。
 - 优点：可进行增量学习和在线学习。可塑性、稳定性好。
 - **SOM**: 最常用的聚类方法之一。竞争学习型无监督。高维→低维。
 - **级联相关网络**: “构造性”神经网络代表。
 - • **Hopfield 网络**: 反馈型神经网络。
 - **Elman 网络**: 递归型神经网络代表，允许环形结构。
 - **Boltzmann 机**: 基于能量的模型代表。
- 受限 Boltzmann 机使用反批量化算法训练。**

5.6 深度学习

(1) 典型的深度学习模型就是很深层的神经网络。

然而误差在多隐层内逆传播时，往往会出现梯度消失。

(2) tricks:

预训练: 监督逆向传播，每次训练一层隐层。

微调: 预训练全部完成后，对全网络进行微调训练。

权共享: 一组神经元使用相同的连接权值。

Dropout: 在每轮训练时随机选择一些参数令其不被更新。

ReLU: 更直观的激活函数。

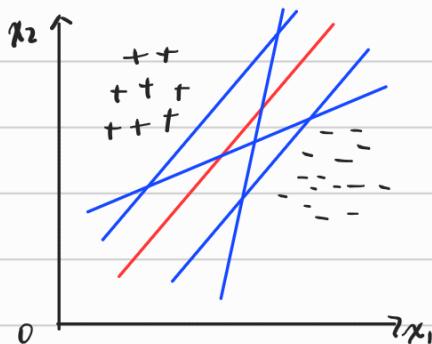
交叉熵: BP中用交叉熵 $-\frac{1}{m} \sum_{i=1}^m y_i \log \hat{y}_i$ 代替 MSE。

(3) 深度学习又叫“特征学习”或“表示学习”。

Chapter 6 支持向量机

6.1 间隔与支持向量

给定训练样本 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, $y_i \in \{-1, +1\}$ ，分类学习最基本的想法是基于训练集 D 在样本空间找到一个划分超平面，将不同类别样本分开。



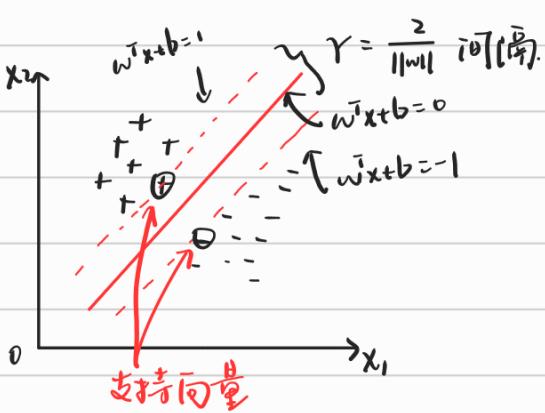
正中间的泛化能力最强，鲁棒性最好。

超平面方程: $w^T x + b = 0$

$w = (w_1, w_2, \dots, w_d)$ 为法向量
任意点 x 到超平面距离为:

$$r = \frac{|w^T x + b|}{\|w\|}$$

位移量，决定超平面与原距离。



SVM 基本型:

最大间隔. 寻找 w 和 b , 使得 Y 最大.

$$\underset{w, b}{\operatorname{argmax}} \frac{2}{\|w\|}$$

$$\text{s.t. } y_i(w^T x_i + b) \geq 1, i \in [m].$$

↓ LP

$$\underset{w, b}{\operatorname{argmin}} \frac{1}{2} \|w\|^2$$

$$\text{s.t. } y_i(w^T x_i + b) \geq 1, i \in [m].$$

6.2 对偶问题

拉格朗日乘子法:

① 引入拉格朗日乘子得到拉格朗日函数.

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i(w^T x_i + b))$$

$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)$

② 令 $L(w, b, \alpha)$ 对 w 和 b 的偏导数为零可得:

$$w = \sum_{i=1}^m \alpha_i y_i x_i, \quad 0 = \sum_{i=1}^m \alpha_i y_i$$

③ 回代得:

$$\max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^T x_j, \text{ s.t. } \sum_{i=1}^m \alpha_i y_i = 0, \alpha_i \geq 0, i \in [m].$$

解出 α 后, 求出 w 与 b 即可得模型:

$$f(x) = w^T x + b = \sum_{i=1}^m \alpha_i y_i x_i^T x + b$$

KKT 条件: $\begin{cases} \alpha_i \geq 0 \\ 1 - y_i f(x_i) \leq 0 \\ \alpha_i (1 - y_i f(x_i)) = 0 \end{cases} \Rightarrow \text{只有 } \alpha_i = 0 \text{ 或 } y_i f(x_i) = 1$

若 $\alpha_i = 0$, 则样本不会在模型中出现, 不会对 $f(x)$ 有影响.

若 $\alpha_i > 0$, 则 $y_i f(x_i) = 1$, 对应样本点位于最大间隔边界上, 是支持向量.

解的稀疏性: 训练完成后, 最终模型仅与支持向量有关.

如何求解③中式子? 是一个二次规划问题, 正比于训练样本数, 会造成较大开销.

SMO 思路: 不断执行如下两个步骤直至收敛:

step1: 选取一对需更新的变量 α_i 和 α_j .

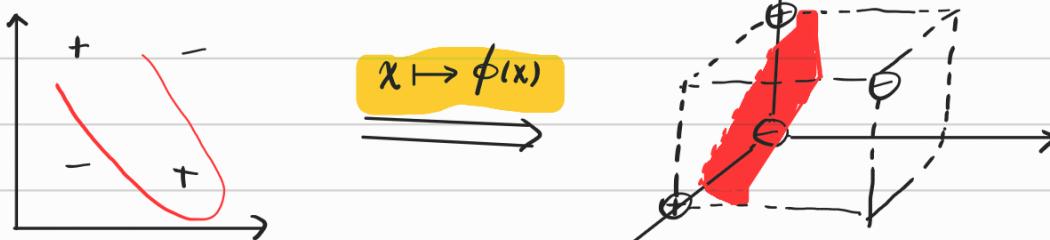
Step 2: 固定 α_i 和 α_j 以外的参数, 求解对偶问题更新 α_i 和 α_j .

仅考虑 α_i 和 α_j 时, 对偶问题的约束 $0 = \sum_{i=1}^m \alpha_i y_i$ 变为 $\alpha_i y_i + \alpha_j y_j = c$, $\alpha_i, \alpha_j \geq 0$.

用 α_i 表示 α_j , 带入对偶问题是有可能解. 对任意支持向量 (x_s, y_s) 有 $y_s f(x_s) = 1$.
由此可解出 b . (通常使用所有支持向量求解的平均值).

6.3 核函数.

特征空间映射:



若不存在一个能正确划分两类样本的超平面怎么办?

将样本从原始空间映射到一个更高维的特征空间.

设样本 x 映射后的向量为 $\phi(x)$, 划分超平面为 $f(x) = w^\top \phi(x) + b$.

原始问题、对偶问题及预测中才的对应位置全部变为 $\phi(x)$ 即可.

$$\text{核函数: } K(x_i, x_j) = \phi(x_i)^\top \phi(x_j)$$

绕过显式高维映射, 以及计算高维内积的困难.

Mercer 定理: 若一个对称函数所对应的核矩阵半正定, 则它就能作为核函数
来使用.

常用核函数:

$$\left\{ \begin{array}{ll} \text{线性核} & K(x_i, x_j) = x_i^\top x_j \\ \text{多项式核} & K(x_i, x_j) = (x_i^\top x_j)^d \quad d \geq 1 \\ \text{高斯核} & K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad \sigma > 0 \text{ 带宽} \\ \text{拉普拉斯核} & K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right) \quad \sigma > 0 \\ \text{Sigmoid 核} & K(x_i, x_j) = \tanh(\beta x_i^\top x_j + \theta) \quad \beta > 0, \theta < 0 \end{array} \right.$$

文本数据常用线性核, 情况不明时可先尝试高斯核.

6.4 软间隔和正则化.

引入软间隔 soft margin, 允许在一些样本上不满足约束: $y_i(w^\top x_i + b) \geq 1$.

最大化间隔同时, 不满足约束的样本尽可能少, 优化目标可写为:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \ell_{0/1}(y_i(w^\top x_i + b) - 1)$$

其中 $\ell_{0/1}$ 是 0/1 损失函数: $\ell_{0/1}(z) = \begin{cases} 1 & \text{if } z < 0 \\ 0 & \text{o.w.} \end{cases}$ / 非凸非连续，数学性质不好.

替代损失函数: 一般是 0/1 损失函数的上界.

① hinge 损失: $\ell_{\text{hinge}}(z) = \max(0, 1-z)$

② 指数损失: $\ell_{\text{exp}}(z) = \exp(-z)$

③ 对数损失: $\ell_{\log}(z) = \log(1 + \exp(-z))$

Q: 求解替代函数得到的解是否仍是原问题的解?

以 hinge 函数为例, 原始问题: $\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \max(0, 1 - y_i(w^T x_i + b))$

引入松弛变量 ξ_i :

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

$$\text{s.t. } y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i \in [m].$$

$$\text{对偶问题: } \max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$\text{s.t. } \sum_{i=1}^m \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i \in [m].$$

根据 KKT 条件可知最终模型仅与支撑向量有关.

能运用对数损失函数 ℓ_{\log} 来替代损失函数?

① 支持向量机与对称回归优化目标相近, 通常性能相当.

② 优势在于输出具有自然的概率意义.

③ 对称回归可直接用于多分类任务.

④ 对称回归的解依赖于更多的训练样本, 预测开销更大.

一般形式:

$$\min_f \underbrace{\mathcal{L}(f)}_{\text{结构风险}} + C \sum_{i=1}^m \underbrace{l(f(x_i), y_i)}_{\text{经验风险}}$$

结构风险: 描述模型本身
的某些性质.

经验风险: 描述模型与训练数据的
契合程度 (误差)

L_p 范数基常用的正则化项, 其中 L_2 范数 $\|w\|_2$ 倾向于 w 的分量取值尽量均衡, 即非零分量尽量稠密, 而 L_1 和 L_∞ 则倾向于 w 的分量尽量稀疏, 即非零分量个数尽量少.

6.5 支持向量回归(SVR).

允许模型输出与实际输出间存在 ε 的差值, 落入 ε 间隔带的样本不计算损失, 从而使得模型获得稀疏性.

$$\text{原始问题: } \min_{w,b,\xi_i} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m (\xi_i + \hat{\xi}_i)$$

$$\text{s.t. } f(x_i) - y_i \leq \varepsilon + \xi_i,$$

$$y_i - f(x_i) \leq \varepsilon + \hat{\xi}_i,$$

$$\xi_i \geq 0, \quad \hat{\xi}_i \geq 0, \quad i \in [m].$$

对偶问题: $\max_{\alpha_1, \alpha_2} \sum_{i=1}^m y_i (\hat{\alpha}_i - \alpha_i) - \varepsilon (\hat{\alpha}_i + \alpha_i) - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m (\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j) \hat{x}_i^\top x_j$

s.t. $\sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) = 0, 0 \leq \alpha_i, \hat{\alpha}_i \leq C$

预测: $f(x) = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) \hat{x}_i^\top x + b$.

6.6 核方法

表达定理: 对于任意单调递增函数 $\varphi: [0, +\infty) \mapsto \mathbb{R}$ 和任意非负损失函数 ℓ : $\mathbb{R}^m \mapsto [0, +\infty)$, 优化问题:

$$\min_{h \in H} F(h) = \varphi(\|h\|_H) + \ell(h(x_1), h(x_2), \dots, h(x_m))$$

的解总可以写为 $h^*(x) = \sum_{i=1}^m \alpha_i K(x, x_i)$.

KLDA: 核线性判别分析.

核技巧是机器学习处理非线性问题的基本技术之一.

Chapter 7 贝叶斯分类器

7.1 贝叶斯决策论

对分类任务而言, 在所有相关概率已知的理想情形下, 贝叶斯决策论考虑如何基于这些概率和误判损失来选择最优的类别标记.

给定 N 个类别, 令 λ_{ij} 代表将第 j 类样本误分类为第 i 类所产生的损失, 则基于后验概率将样本 x 分到第 i 类的条件风险为:

$$R(c_i|x) = \sum_{j=1}^N \lambda_{ij} P(c_j|x)$$

贝叶斯准则:

$$h^*(x) = \arg \min_{c \in \mathcal{C}} R(c|x).$$

h^* 称为贝叶斯最优分类器, 其总体风险称为贝叶斯风险.

机器学习所要实现的是基于有限训练样本集尽可能准确地估计出后验概率 $P(c|x)$, 主要有两种策略:

(1) **判别式**: 直接对 $P(c|x)$ 建模. 如: 决策树、BP 神经网络、SVM.

(2) **生成式**: 首先对 $P(x, c)$ 建模, 再由此 $P(c|x) = \frac{P(x, c)}{P(x)}$ 获得.

如: 贝叶斯分类器.

贝叶斯定理:

$$P(c|x) = \frac{P(c) P(x|c)}{P(x)}$$

先验
似然
后验
边缘

7.2 极大似然估计

先假设某种概率分布形式，再基于训练样例对参数进行估计。

假定 $P(x|c)$ 具有确定的分布形式，且被参数 θ_c 唯一确定，则任务就是利用训练集 D 来估计参数 θ_c 。

θ_c 对于训练集 D 中第 c 类样本组成的集合 D_c 的似然为：

$$P(D_c|\theta_c) = \prod_{x \in D_c} P(x|\theta_c)$$

连乘易造成下溢，因此用对数似然：

$$\text{LL}(\theta_c) = \log P(D_c|\theta_c) = \sum_{x \in D_c} \log P(x|\theta_c)$$

于是 θ_c 的极大似然估计 $\hat{\theta}_c$ 为： $\hat{\theta}_c = \arg \max_{\theta_c} \text{LL}(\theta_c)$

估计结果的准确性严重依赖于所假设的概率分布形式是否符合潜在的真实数据分布。

7.3 朴素贝叶斯分类器. Naive Bayes

困难：所有属性上的联合概率难以从有限训练样本来估计获得。

朴素贝叶斯分类器采用“属性条件独立性假设”，i.i.d.？

$$P(c|x) = \frac{P(c) P(x|c)}{P(x)} = \frac{P(c)}{P(x)} \prod_{i=1}^d P(x_i|c)$$

c 为属性类， x_i 为 x 在第 i 个属性上的取值。

$P(x)$ 对所有类别相同，有：

$$h_{nb}(x) = \arg \max_{c \in \mathcal{Y}} P(c) \prod_{i=1}^d P(x_i|c)$$

朴素贝叶斯分类器的训练过程就是基于训练集 D 来估计类先验概率 $P(c)$ ，并为每个属性估计条件概率 $P(x_i|c)$ 。

D_c 表示训练集 D 中第 c 类集合。样本充足，则类先验概率：

$$P(c) = \frac{|D_c|}{|D|}$$

离散属性而言， D_{c,x_i} 表示 D_c 中在第 i 个属性上取值为 x_i 的样本组成的集合。

$$P(x_i|c) = \frac{|D_{c,x_i}|}{|D_c|}$$

对连续属性，可用 pdf：假设 $P(x_i|c) \sim N(\mu_{c,i}, \sigma_{c,i}^2)$

$$P(x_i|c) = \frac{1}{\sqrt{2\pi} \sigma_{c,i}} \exp\left(-\frac{(x_i - \mu_{c,i})^2}{2\sigma_{c,i}^2}\right)$$

拉普拉斯修正:

若某个属性值在训练集中没有与某个类同时出现过，则直接计算会出现问题，其他属性提供的信息会被抹去。

令 N 表示训练集 D 中可能的类别数， N_i 表示第 i 个属性可能的取值数。

$$\hat{P}(c) = \frac{|D_c| + 1}{|D| + N}$$

$$\hat{P}(x_i|c) = \frac{|D_{c,x_i}| + 1}{|D_c| + N_i}$$

7.4 半朴素贝叶斯分类器

基本思路：逐步考虑一部分属性之间的相互依赖关系。

独依赖估计 (ODE): 假设每个属性在类别上最多依赖一个属性。

$$P(c|x) \propto P(c) \prod_{i=1}^d P(x_i|c, p_{ai}) \quad \text{x}_i \text{ 的父属性}$$

父属性的确定：

SPODE：假设所有属性都依赖于同一属性（超父）。

TAN：计算任意两个属性之间的条件互信息。

$$I(x_i, x_j | y) = \sum_{x_i, x_j: c \in y} P(x_i, x_j | c) \log \frac{P(x_i, x_j | c)}{P(x_i | c) P(x_j | c)}$$

以属性为结点构建完全图，任意两个结点之间边的权重设为 $I(x_i, x_j | y)$ ，构建完全图的最大带权生成树。

AODE (Averaged): 尝试将每个属性作为超父构建 SPODE。

将拥有足够训练数据支撑的 SPODE 集成起来作为最终结果。

无需模型选择，既能通过预计算节省预测时间，又能采取懒惰学习方法在预测时再进行计数，易于实现增量学习。

7.5 贝叶斯网

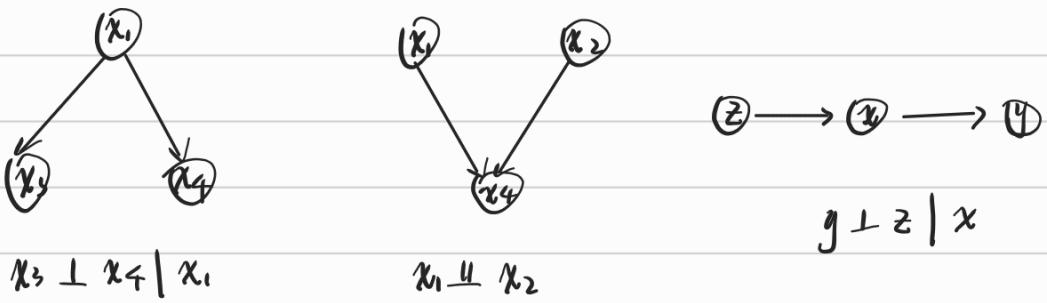
又叫信念网，借助有向无环图刻画属性依赖关系，并使用条件概率表描述属性联合概率分布。

$B = \langle G, \theta \rangle$ ， G 结构：有向无环图， θ ：参数。

给定父结点集，贝叶斯网假设每个属性与其非后裔属性独立。

$$P_B(x_1, x_2, \dots, x_d) = \prod_{i=1}^d P_B(x_i | \pi_i) = \prod_{i=1}^d \theta_{x_i | \pi_i}$$

三变量间的依赖关系：



若 x_4 已知，则 $x_1 \perp\!\!\! \perp x_2 | x_4$

若 x_4 未知，则 $x_1 \perp\!\!\! \perp x_2 | x_4$

分析条件独立性：有向分离。得到道德图。令父结点相连称道德化

贝叶斯网学习的主要任务就是根据训练集来找出结构最恰当的贝叶斯网。评估函数评估贝叶斯网与训练数据的契合程度。例如最小描述长度 MDL。

推断：基于已知属性变量的观测值，推测其他属性变量的取值。

常见做法：① 吉布斯采样

② 变分推断

7.6 EM 算法

令 X 表示已观测变量集， Z 表示隐变量集，欲对模型参数 θ 做极大似然估计，则应最大化对数似然函数：

$$LL(\theta | X, Z) = \ln P(X, Z | \theta)$$

对隐变量 Z 计算期望，根据训练数据最大化对数边际似然。

$$L((\theta | X) = \ln P(X | \theta) = \ln \sum P(X, Z | \theta)$$

以初值 θ^0 为起点，迭代执行以下步骤直到收敛：

① 基于 θ^t 推断隐变量 Z 的期望，记为 Z^t 。

② 基于已观测变量 X 和 Z^t 对参数 θ 做极大似然估计，记为 θ^{t+1} 。

E步：当 θ 已知 \rightarrow 根据训练数据推断 Z 。

M步：当 Z 已知 \rightarrow 对 θ 做极大似然估计。

一般形式：E-M步交替计算，直到收敛。

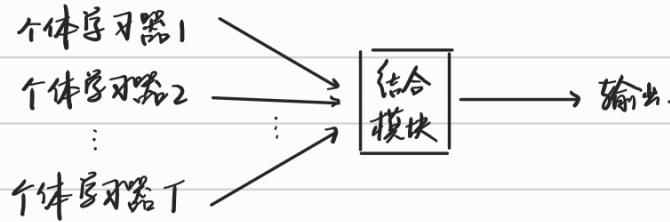
E步：计算期望，利用前一步估计的参数值计算对数似然的期望值。

M步：最大化，寻找能使E步产生的似然期望最大化的参数值。

Chapter 8 集成学习

8.1 个体与集成

集成学习通过构建并结合多个学习器来完成学习任务，有时也被称为多分类器系统，基于委员会的学习。



同质集成: 只包含同种类型的个体学习器，个体学习器也叫基学习器，算法叫基学习算法。

异质集成: 由不同学习算法生成，不再有基学习法，称组件学习器。

集成学习常获得比单一学习器显著优越的泛化性能

要想获得好的集成个体学习器应“好而不同”，即个体学习器要有一定的准确性，并且也有着异样性。

考虑二分类问题，假设基分类器错误率为：

$$P(h_i(x) \neq f(x)) = \varepsilon$$

假设集成通过简单投票法结合 T 个分类器，若有超过半数的基分类器正确则分类正确。

$$F(x) = \text{sign}\left(\sum_{i=1}^T h_i(x)\right)$$

假设基分类器的错误率相互独立，则：

$$P(F(x) \neq f(x)) = \sum_{k=0}^{\lfloor T/2 \rfloor} \binom{T}{k} (1-\varepsilon)^k \varepsilon^{T-k} = \exp\left(-\frac{1}{2}T(1-2\varepsilon)^2\right) \rightarrow 0$$

如何产生“好而不同”的个体学习器是集成学习研究的核心。

集成学习分成两大类：

① 个体学习器存在强依赖关系，必须串行生成序列化方法，如：Boosting。

② 不存在强依赖关系，可同时生成的并行化方法，如 Bagging 和 RF。

8.2 Boosting

先从初始训练集中训练一个基学习器，再根据基学习器表现对训练样本分布调整，使先前错分样本后续得到更大关注，基于调整后的样本训练下一个基学习器，反复直到达到指定值 T，最终将 T 个学习器加权结合。

AdaBoost:

基学习器的线性组合： $H(x) = \sum_{t=1}^T \alpha_t h_t(x)$

训练的每一阶段都检查先前生成的基分类器是否比随机猜测好。

对特定的数据分布进行学习：{重采样法
重叠样法}

8.3 Bagging 和 Random Forest

基于自助重样法，给定包含 m 个样本数据集，使得下次采样该样本仍能被选中。经过 m 个随机采样，得到 m 个样本的采样集。

初始训练集中有的样本在采样集中多次出现，有的从未出现。

Bagging 流程：

① 可采样出 m 个含 m 个训练样本的采样集，然后基于每个采样集训练一个基学习器，再将这些学习器结合。

② 对分类任务使用简单投票法，票数一样随机取一个。

③ 对回归任务使用简单平均法。

优点：

① 时间复杂度低。

② 可使用包外估计。（剩下的 36.8% 的样本作为验证集）

从偏差一方差：

Boosting：降低偏差，可对泛化性能相当弱的学习器构造出很强的集成。

Bagging：降低方差，在不剪枝的决策树、神经网络等易受样本影响的学习器上效果更好。

随机森林(RF)：Bagging 的一个扩展变体。

RF 以决策树为基学习器构建 Bagging 集成，在决策树训练过程中引入了随机属性选择。

传统决策树在选择划分属性时是在当前结点的属性集合选一个最优属性。

RF 中，对基决策树的每个结点，先从该结点属性集合随机选择一个包含 k 个属性的子集，然后从子集中选择最优属性划分。

$k=d$ ，与传统决策树相同。

$k=1$ ，随机选择一个属性用以划分。

一般 $k = \log_2 d$ 。

8.4 结合策略

学习器结合带来的好处：

① 从统计方面，单学习器可能因误差而导致泛化性能下降，结合多个学习器会降低。

② 从计算方面，多次运行之后结合，可降低陷入局部极小点风险。

③ 从表示方面，结合多个学习器，相应的假设空间有所扩大，有可能学到更好的近似。

结合方法：

(1) 平均法：

· 简单平均： $H(x) = \frac{1}{T} \sum_{i=1}^T h_i(x)$

· 加权平均： $H(x) = \sum_{i=1}^T w_i h_i(x), w_i \geq 0, \sum_{i=1}^T w_i = 1$

集成学习的各种结合方法都可以看成是加权平均法的变种或特例。

(2) 投票法：

· 绝对多数投票法： $H(x) = \begin{cases} j & \text{if } \sum_{i=1}^T h_i^j(x) > \frac{1}{2} \sum_{k=1}^T \sum_{i=1}^T h_i^k(x) \\ \text{rejection.} & \text{o.w.} \end{cases}$

· 相对多数投票法： $H(x) = \underset{j}{\operatorname{argmax}} \sum_{i=1}^T h_i^j(x)$
(得票最多的，相同随机选一个)

· 加权投票法： $H(x) = \underset{j}{\operatorname{argmax}} \sum_{i=1}^T w_i h_i^j(x)$

不同类型的个体学习器可能产生不同类型的 $h_i^j(x)$ 的值，有：

{
· 硬标记： $h_i^j(x) \in \{0, 1\}$. 硬投票。
· 软概率： $h_i^j(x) \in [0, 1]$. 软投票。

若某学习器的类型不同，可将软概率输出转化为硬标记输出再投票。

(3) 学习法： Stacking

将个体学习器称为初级学习器，用于结合的学习器称为次级学习器。Stacking 先从初始数据集训练出初级学习器，将初级学习器的输出软概率作为次级学习器的输入。用响应线性规则 (MLR) 作为次级学习算法效果较好。

贝叶斯模型平均 (BMA) 基于后验概率来为不同模型赋予权重，可视为加权平均的一种实现。Stacking 通常优于 BMA，鲁棒性好，对近似误差敏感。

8.5 多样性

8.5.1 误差-分歧分解

定义学习器 h_i 的分歧： $A(h_i|x) = (h_i(x) - H(x))^2$

$$\text{集成的分类: } \bar{A}(h|x) = \sum_{i=1}^T w_i A(h_i|x) = \sum_{i=1}^T w_i (h_i(x) - H(x))^2$$

公式不仅代表了个体学习器在样本 x 上的不一致，即在一定程度上反映了个体学习器的多样性。个体学习器 h_i 和集成 H 的平方误差差值为：

$$E(h_i|x) = (f_i(x) - h_i(x))^2$$

$$E(H|x) = (f(x) - H(x))^2$$

令 $\bar{E}(h|x) = \sum_{i=1}^T w_i E(h_i|x)$ 表示个体学习器误差的加权均值，有

$$\bar{A}(h|x) = \sum_{i=1}^T w_i \bar{E}(h_i|x) - \bar{E}(H|x)$$

$$= \bar{E}(h|x) - E(H|x)$$

上式对所有样本 x 均成立，令 $p(x)$ 表示样本的频率密度，则在全样本上有：

$$\sum_{i=1}^T w_i \int A(h_i|x) p(x) dx = \sum_{i=1}^T w_i \int E(h_i|x) p(x) dx - \int E(H|x) p(x) dx.$$

个体学习器 h_i 在全样本上的泛化误差和为：

$$E_i = \int E(h_i|x) p(x) dx \quad A_i = \int A(h_i|x) p(x) dx$$

集成的泛化误差：

$$E = \int E(H|x) p(x) dx.$$

令 $\bar{E} = \sum_{i=1}^T w_i E_i$ 表示个体学习器泛化误差的加权均值。

$\bar{A} = \sum_{i=1}^T w_i A_i$ 表示个体学习器的加权分歧值，有：

$$\bar{E} = \bar{E} - \bar{A}$$

显式：个体学习器准确率越高，多样性越大，则集成效果越好。

Q: 为什么不能把 $\bar{E} - \bar{A}$ 作为优化目标求解？

① 它们定义在整个样本空间上。

② \bar{A} 不是一个可直接操作的多样性度量。

③ 上面的推导过程只适用于回忆学习，难以扩展到分类上。

8.5.2 多样性度量

一般通过两台决策器的预测结果列联表定义。

	$h_i = +1$	$h_i = -1$
$h_j = +1$	a	c
$h_j = -1$	b	d

$$a+b+c+d=m$$

· 不合度量:

$$dis_{ij} = \frac{b+c}{m}$$

· QI度量: $Q_{ij} = \frac{ad-bc}{ad+bc}$

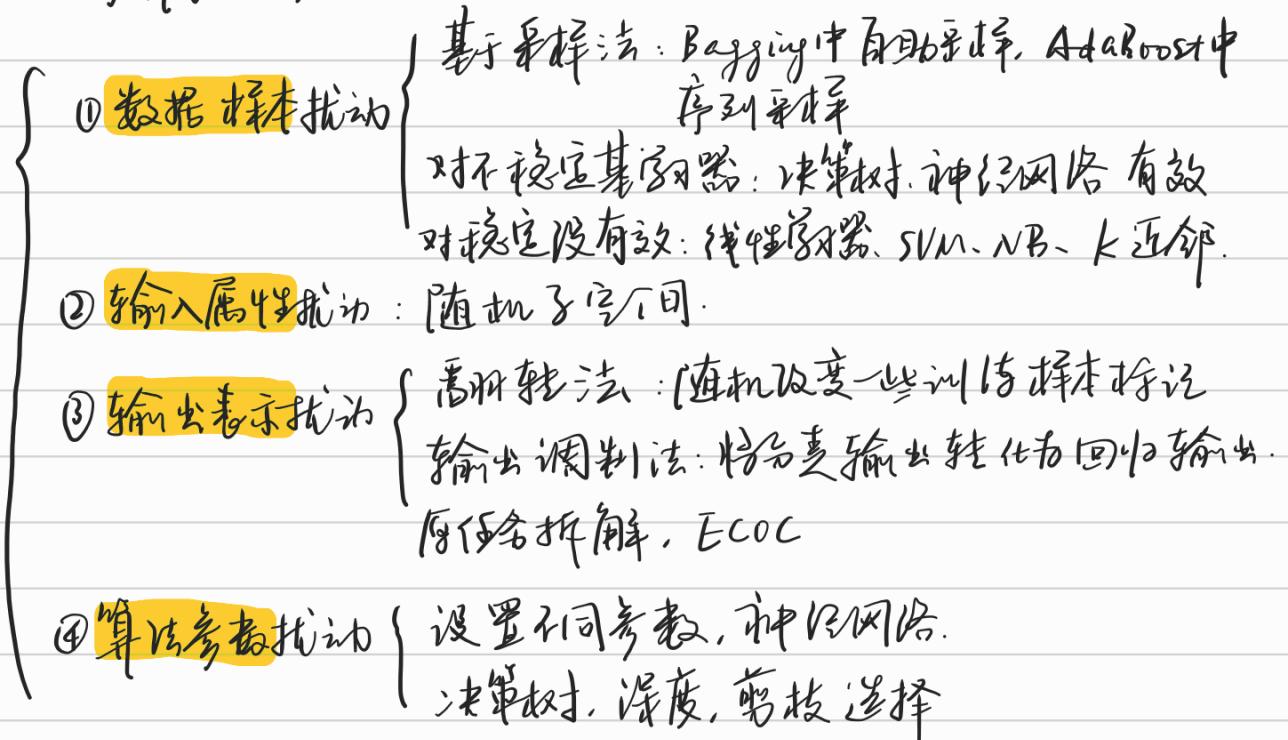
$$|Q_{ij}| \geq |P_{ij}|$$

· 相关系数: $\rho_{ij} = \frac{ad - bc}{\sqrt{(a+b)(a+c)(c+d)(b+d)}}$

· K-统计量: $K = \frac{P_1 - P_2}{1 - P_2}$

$$P_1 = \frac{a+d}{m}, \quad P_2 = \frac{(a+b)(a+c) + (c+d)(b+d)}{m^2}$$

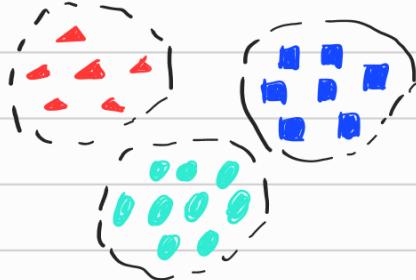
8.5.3 多样性增强.



Chapter 9 聚类

9.1 聚类任务

将数据样本划分为若干个通常不相交的簇 (cluster).



9.2 评估度量, 也叫有效性指标.

聚类结果是 簇内相似度高且 簇间相似度低.

外部指标: 将聚类结果与某个“参考模型”进行比较.

如: Jaccard 指数, FM 指数, Rand 指数.

内部指标: 直接考察聚类结果.

如 DB 指数, Dunn 指数.

9.3 距离计算:

距离度量性质

非负
同一对称
直进

常用距离形成: 闵可夫斯基距离.

$$dist_{mk}(x_i, x_j) = \left(\sum_{n=1}^n |x_{in} - x_{jn}|^p \right)^{\frac{1}{p}}$$

$p=2$: 欧氏距离.

$p=1$: 曼哈顿距离

属性分类: 连续、离散、有序、无序.

对无序属性, 使用 VDM (Value Difference Metric).

对混合属性, 使用 Minkowski DM. (含有 VDM).

加权距离: 样本中不同属性的重要性不同时.



聚类方法分类:

(1) 原型聚类: 聚类结构能通过一组原型刻画. 先对原型初始化, 然后对 I²型迭代求解
K-means, 序向量量化 (LVQ), 高斯混合聚类

(2) 密度聚类: 聚类结构能通过样本分布的紧密程度确定. 从样本密度的角度考察样本的可连接性, 并基于可连接样本不断扩展.
DBSCAN, OPTICS, DENCLUE

(3) 层次聚类: 能够产生不同精度的聚类结果. 在不同层次上对数据集进行划分, 从而形成树形的聚类结构.
AGNES (自底向上), DIANA (自顶向下)

9.4 原型聚类

(1) K-Means 步骤:

Step 1. 随机选取 k 个样本点作为簇中心.

Step 2. 将其他样本点根据其与簇中心的距离, 划分给最近的簇.

Step 3. 更新各簇的均值向量, 将其作为新的簇中心.

Step 4. 若所有簇中心未发生改变, Stop; 否则执行 Step 2.

(2) LVQ: 假设数据样本带有类别标注, 也是试图找到用序向量来刻画聚类结构.

实际上是由聚类来形成更细的“子类”结构.

(3) 高斯混合聚类: 采用概率模型. 高斯混合分布. $M\mathcal{L}B + EM$.

9.5 密度聚类.

DBSCAN: 基于邻域参数刻画样本分布的紧密程度.

$(\epsilon, \text{MinPts})$

{
ε邻域
核心对象

密度直达

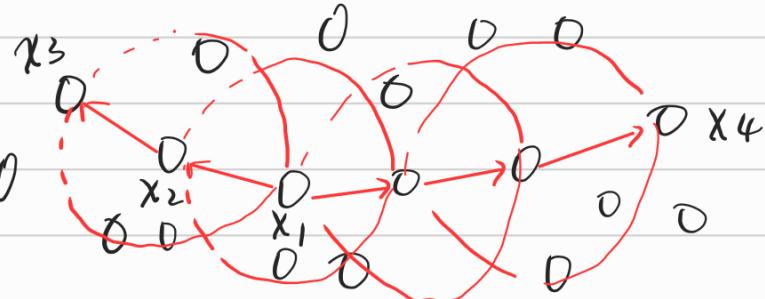
$x_1 \rightarrow x_2$

密度可达

$x_1 \rightarrow x_3$

密度相连

$x_3 \leftrightarrow x_4$



簇的定义: 由密度可达关系引出的最大密度相连样本集合.

{
连接性
最大性

9.6 层次聚类

AGNES: 自底向上.

将样本中的每一个样本看做一个初始聚类簇, 然后在算法运行的每一步中找出距离最近的两个聚类簇进行合并, 该过程不断重复, 直到达到预设的簇个数.

Chapter 10 降维与度量学习

10.1 K近邻学习

监督学习方法. 需要训练样本, 以及某种距离度量.

对于给定的测试样本, 找到训练集中距离最近的k个样本, 对于分类问题使用“投票法”获得预测结果, 对于回归问题使用“平均法”. 还可以加权.

KNN是懒惰学习代表, 训练开销为零, 待收到测试样本再进行处理.

10.2 低维嵌入

维数灾难: 在高维情形下出现的数据样本稀疏, 距离计算困难等问题.

缓解方法：降维

Q: 为什么可以降维？

数据样本虽然高维的，但与学习任务密切相关的也许仅仅是某低维分布，即高维空间的一个低维嵌入。

多维缩放 MDS. 要求原始空间中样本之间的距离在低维空间保持。
线性降维方法。

10.3 主成分分析.

Q: 对已知属性空间中的样本点，如何用一个超平面将所有样本进行表达？

① 最近重构性：样本点到这个平面的距离都足够近。

② 最大可分性：样本点在这个超平面的投影尽可能分开。

PCA 的求解：

$$\max_W \text{tr}(W^T X X^T W)$$

$$W \text{ s.t. } W^T W = I$$

流程：输入：样本集 $D = \{x_i\}_{i=1}^m$ ，低维空间维数 d' 。

过程：① 对所有样本进行中心化： $x_i \leftarrow x_i - \frac{1}{m} \sum_{j=1}^m x_j$ 。

② 计算样本的协方差矩阵 $X X^T$ 。

③ 对 $X X^T$ 做特征值分解。

④ 取最大的 d' 个特征值所对应的特征向量

$$w_1, w_2, \dots, w_{d'}$$

输出：投影矩阵 $W = (w_1, w_2, \dots, w_{d'})$ 。

重构阈值。

10.4 核化线性降维

非线性降维的一种方法，基于核技巧。

KPCA

10.5 流形学习

流形学习是一类借鉴了拓扑流形概念的降维，“流形”是在局部与欧式空间同胚的空间，局部具有欧氏空间的性质。

10.5.1 等度量映射 (Isomap).

低维嵌入流形上的测地线距离不能用高维空间的直线距离计算，但能用近邻距离来近似。在近邻连接图上计算两点之间最短路径 (Dijkstra-Floyd)。 \rightarrow KNN, ϵ -NN.

10.5.2 局部线性嵌入 (LLE)

LLE保持样本之间的线性关系。

假设样本点 x_i 能通过邻域样本 x_j, x_k, x_l 的线性组合重构：

$$x_i = w_{ij} x_j + w_{ik} x_k + w_{il} x_l$$

LLE先为每个样本 x_i 找到其近邻下标集合 Q_i ，然后计算出基于 Q_i 的样本点对 x_i 进行线性重构的系数 w_i 。

LLE在低维空间 w_i 不变，于是 x_i 对应的低维空间坐标 z_i 可由：

$$\min_{w_1, w_2, \dots, w_m} \sum_{i=1}^m \left\| z_i - \sum_{j \in Q_i} w_{ij} z_j \right\|_2^2$$

令 $Z = (z_1, z_2, \dots, z_m) \in \mathbb{R}^{d' \times m}$, $(W)_{ij} = w_{ij}$, $M = (I - W)^T (I - W)$.

优化式可重写为： $\min_Z \text{tr}(Z M Z^T)$

$$\text{s.t. } Z Z^T = I$$

并通过特征值分解求解。最后取 M 的最小 d' 个特征值的特征向量。

10.6 度量学习

对两个 d 维样本 x_i 和 x_j ，它们之间的平方欧式距离：

$$\text{dist}_{\text{eu}}^2(x_i, x_j) = \|x_i - x_j\|_2^2 = \text{dist}_{ij,1}^2 + \text{dist}_{ij,2}^2 + \dots + \text{dist}_{ij,d}^2$$

其中 $\text{dist}_{ij,k}$ 表示 x_i 与 x_j 在第 k 维上的距离。带权重的：

$$\text{dist}_{\text{wed}}^2(x_i, x_j) = \|x_i - x_j\|_2^2 = \sum_{k=1}^d w_k \cdot \text{dist}_{ij,k}^2$$

$$= (x_i - x_j)^T W (x_i - x_j)$$

$w_i \geq 0$, $W = \text{diag}(w)$ 是一个对角矩阵, $(W)_{ii} = w_i$, 可通过学习确定。

马氏距离：

$$\text{dist}_{\text{mah}}^2(x_i, x_j) = (x_i - x_j)^T W (x_i - x_j) = \|x_i - x_j\|_M^2$$

M 亦称度量矩阵，度量学习就是对 M 的学习。川以恒是牛逼

对称矩阵，即必有正交基 P 使得 $M = P P^T$.

近邻成员分析 NCA.

Chapter 11 特征选择与潮流学习

11.1 子集搜索与评价

特征 $\begin{cases} \text{定义：描述物体的属性} \\ \text{分类：相关、无关、冗余} \end{cases}$

特征选择：从给定的特征集合中选出一个或多个特征子集，必须确保不丢失重要特征。

原因：① 减轻维数灾难：在少量属性上构建模型。
② 降低学习难度：留下关键信息

子集搜索：用贪心策略选择包含重要信息的特征子集。

$\begin{cases} \text{前向} : \text{逐渐增加相关特征.} \\ \text{后向} : \text{从完整的特征集合开始，逐渐减少特征.} \\ \text{双向} : \text{每一步逐渐增加相关特征，同时减少无关特征.} \end{cases}$

子集评价：与样本标记对应的划分距离越小，则说明当前特征子集越好。特征子集响应了对数据集的一个划分。

用信息熵进行子集评价。

* 常见的特征选择方法：

$\begin{cases} \text{过滤法} & \begin{cases} \text{方差过滤} \\ \text{相关性过滤} \end{cases} \\ & \begin{cases} \text{卡方过滤} \\ \text{F检验} \\ \text{互信息法} \end{cases} \end{cases}$

包裹法：递归特征消除法 RFE

$\begin{cases} \text{正则化(最小覆盖)} & \begin{cases} L_1 \\ L_2 \end{cases} \\ \text{嵌入法} & \begin{cases} \text{决策树} \\ RF \\ \text{梯度提升树} \end{cases} \\ \text{树模型} & \end{cases}$

11.2 过滤式选择

利用特征选择过程过滤原始数据，再用过滤后的特征训练模型。

Relief：为每个原始特征赋予一个“相关度量”：二范数。

Relief-F：多分类。

11.3 包裹式选择

包裹式选择直接把最终将要使用的学习器的性能作为特征子集的评价准则。

目的是为给定学习器选择最有利于其性能、“量身定做”的特征子集。

包裹式选择性能比过滤式好，但是计算开销大。

11.4 嵌入式选择

LVW 包裹式特征选择方法

将特征选择过程与学习器训练过程融为一体，两者在同一个优化过程中完成。

在学习器训练过程中自动地进行特征选择。

考虑最简单的线性回归模型，以平方误差为损失函数，并引入L₂正则化项。则有：

$$\min_w \sum_{i=1}^m (y_i - w^T x_i)^2 + \lambda \|w\|_2^2 \quad \text{又叫岭回归}$$

将L₂替换为L₁，则有 LASSO：

$$\min_w \sum_{i=1}^m (y_i - w^T x_i)^2 + \lambda \|w\|_1 \quad \text{易获得稀疏解}$$

11.5 稀疏表示与字典学习

稀疏表示：将数据集看成一个矩阵，每行对应一个样本，每列对应一个特征。矩阵中有很多零元素，且非零元素列出现。

优势：①文本数据线性可分

②存储高效。

字典学习：为普通稠密表达的样本找到合适的字典，将样本转化为稀疏表示。

11.6. 压缩感知

能否利用部分数据恢复全部数据？——压缩感知。

限定等距性。 $L_0 \rightarrow L_1$

矩阵与特征值

Chapter 12 计算学习理论

12.1 概述

关注的问题

- 怎么刻画学习这个过程?
- 什么样的问题是可学习的?
- 什么样的问题是易学习的?
- 对于给定的学习算法,能在理论上预测其性能?
- 理论结果如何指导现实问题的算法设计?

12.2 可学习性

12.2.1 什么是“学习”

概念: 概念是从样本空间 X 到标记空间 Y 的映射, 它决定标注 X 的真实标记 y .

目标概念: 如果对任何样例 (x, y) 均有 $c(x) = y$ 成立, 则称 c 为目标概念.

概念类: 所有我们希望学习的目标概念所构成的集合, 用 C 表示.

假设空间: 给定学习算法 L , 它所考虑的所有可能概念的集合, 用 H 表示.

C 与 H 通常是不同的.

可学的: 若目标概念 $c \in C$, 即 H 中存在假设能将所有的示例完全正确分类 (推断与真实标记一致的方式), 则称该问题对学习算法 L 可学的.

不可学的:

对于给定训练集 D , 我们希望基于学习算法 L 得到的模型所对应的假设尽可能接近目标概念 c .

12.2.2 什么是“可学习的”

概率近似正确(PAC): 以较大概率使得误差满足假设上界的模型.

令 δ 表示置信度, 即:

(PAC辨识): 对 $0 < \varepsilon, \delta < 1$, $\forall c \in C$ 和分布 D , 若学习算法 L , 其输出假设 $h \in H$ 满足

$$P(E(h) \leq \varepsilon) \geq 1 - \delta$$

则称 L 能从假设空间 H 中 PAC 辨识概念类 C .

(PAC可学习): 令 m 表示从分布 D 中 i.i.d. 抽样得到的样例数目, $0 < \varepsilon, \delta < 1$, 对所有分布 D , 若存在学习算法 L 和多项式函数 $\text{poly}(\cdot, \cdot, \cdot)$, 使得对于任何 $m = \text{poly}(1/\varepsilon, 1/\delta, \text{size}(x), \text{size}(c))$, L 能从假设空间 H 中 PAC 辨识概念类 C , 则称

概念类C对假设空间H而言是PAC可学习的.

(PAC学习算法): 若L使C为PAC可学习, 且L的运行时间也是poly, 则称L是高效PAC可学习的, L为C的PAC学习算法.

样本复杂度: 满足PAC学习算法上所需的 $m = \text{poly}(1/\epsilon, 1/\delta, \text{size}(x), \text{size}(c))$ 中最小的m.

PAC学习的意义:

①给出了一个抽象地刻画机器学习能力的框架, 基于这个框架可以对很多重要问题进行理论探讨.

②把对复杂算法的时间复杂度的分析转为对样本复杂度的分析.

假设空间H的复杂度是影响可学习性的重要因素之一.

| H | 有限时称为有限假设空间

12.2.3 假设空间复杂性对可学习性的影响

12.2.3.1 有限假设空间

可学
不可学
经验风险最小化(ERM)
不可知 PAC 可学习

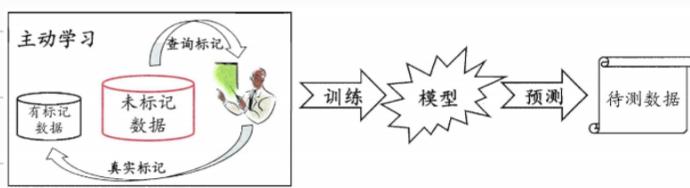
12.2.3.1 无限假设空间 { VC维 与分布无关, 数据独立
Rademacher 复杂度 与分布D或样本D有关

12.3 稳定性

若学习算法上是ERM且稳定的, 则假设空间H可学习.

Chapter 13 半监督学习

13.1 背景



13.2 生成式方法

生成式方法是基于生成式模型的方法。此类方法假设所有数据（无论是否有标记）都是由同一个潜在的模型生成的。未标记数据的标记可看作是模型的缺失参数，通常可基于 EM 算法进行极大似然估计求解。

此类方法简单、易于实现，在有标记数据较少的情况下往往比其他方法性能更好。

关键：模型假设必须准确。

13.3 半监督 SVM (S3VM)

S3VM 尝试找到能将两类有标记样本分开，且穿过数据低密度区域的划分平面。

T3VM：对未标记样本进行各种可能的标记指派，尝试将每个未标记样本分别作为正例或反例。利用局部搜索迭代求解。可能出现类别不平衡。
计算开销大。

13.4 图半监督学习

给定一个数据集，我们将其映射为一个图，数据集中每个样本对应于图中一个结点，若两个样本之间的相似度很高（或相关性很强），则对应的结点之间存在一条边，边的强度正比于样本之间的相似度（或相关性）。我们可以将有标记样本对应的结点染色为深蓝色，而未标记样本结点尚未染色。于是，半监督学习就对应于颜色在图上传播的过程。

通过矩阵乘法。

存储开销高。

仅能考虑训练样本集。

13.5 基于分歧的方法

使用多学习器。

协同训练，针对单视图。（相容互补性）。

13.6 半监督聚类

使用额外的监督信息。

↓ 半监督 K-Means

① 一种是必连与勿连约束，前者指样本必须属于同一个簇，后者指必须不属于。

② 另一种则是少量的有标记样本。→ 半监督 K-Means

Chapter 14 概率图模型

14.1 概率图模型

概率图模型提供了一种描述框架，将学习任务归结为计算变量的概率。在概率图模型中，利用已知变量推测未知变量的分布称为“推断”，其核心是如何基于可观测变量推演出未知变量的条件分布。

生成式：计算联合分布： $P(Y, R, D)$

Y 关心的变量集合， D 为可观测变量集

判别式：计算条件分布： $P(Y|R|D)$

R 为其他变量集合

上述模型时间复杂度高达 $O(2^{|\mathcal{Y}|+|\mathcal{R}|})$ 。

概率图模型是一类用图来表达变量相关关系的概率模型。

{ 结点：随机变量（集合）。

| 边：变量间的概率相关关系

| 有向图：贝叶斯网（Chapter 7 贝叶斯后验器）

| 无向图：马尔可夫网

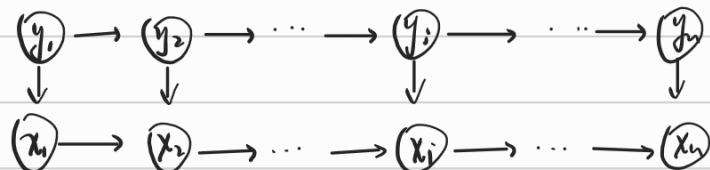
14.2 图模型的两种表示

14.2.1 隐马尔可夫模型（动态贝叶斯网）。

HMM组成：

状态变量： $\{y_1, y_2, \dots, y_n\}$ 隐藏的不可被观测

观测变量： $\{x_1, x_2, \dots, x_n\}$ 表示第*t*时刻的观测值集合。



*t*时刻的状态 y_t 仅依赖于 y_{t-1} 。

$$P(x_1, y_1, \dots, x_n, y_n) = P(y_1) \prod_{i=2}^n P(y_i | y_{i-1}) P(x_i | y_i)$$

HMM的参数： $\lambda = [A, B, \pi]$ 。

• 状态转移概率：在任意时刻*t*，若状态为 s_i ，下-状态为 s_j 的概率。

$$A = [a_{ij}]_{N \times N}, \quad a_{ij} = P(y_{t+1} = s_j | y_t = s_i), \quad 1 \leq i, j \leq N.$$

• 输出观测概率：

$$B = [b_{ij}]_{N \times M}, \quad b_{ij} = P(x_t = o_j | y_t = s_i), \quad 1 \leq i \leq N, \quad 1 \leq j \leq M.$$

• 初始状态概率： $\pi = [\pi_1, \pi_2, \dots, \pi_N], \quad \pi_i = P(y_1 = s_i), \quad 1 \leq i \leq N.$

HMM的生成：给定状态空间 y ，观测空间 X 及 $\pi = [A, B, \pi]$

- ①设置 $t=1$ ，根据 π 选择初始状态 y_1 。
- ②根据状态 y_t 和输出观测概率 B 选择 x_t 。
- ③根据 y_t 和 A 确定 y_{t+1} 。
- ④若 $t < n$ ，设置 $t=t+1$ ，转到②，否则停止。

14.2.2 马尔可夫随机场/条件随机场

结点表示变量(集)、边表示依赖关系。

有一组势函数，亦称因子，主要用于定义概率分布函数。

MRF：分布形式化，使用基于极大团的势函数。

团：图中任意的一个子集，其中任意两个结点间都有边连接。

极大团：图中加入另外任一个结点都不再形成团。

多个变量的连簇分布可基于团分解为多个因子的乘积。

全局可观测性：在给定局部离散的条件下，两个变量子集条件独立。



$$P(X_A, X_B, X_C) = \frac{1}{2} \psi_{AC}(X_A, X_C) \psi_{BC}(X_B, X_C)$$

验证： $P(X_A, X_B | X_C) = P(X_A | X_C) P(X_B | X_C)$

局部可观测性：在给定邻接变量的情况下，一个变量条件独立于其他所有变量。

成对可观测性：在给定其他变量的情况下，两个非邻接变量相互独立。

CRF：给定观测值的MRF

14.3 图模型推断

基于概率图模型定义的分布，能对目标变量的边际分布或某些可观测变量为条件的条件分布进行推断。

参数估计或参数学习问题。

推断方法：

①精确推断：即算法进行变量消去；信念传播（避免冗余计算）

未来点计算开销大.

② 近似推断:

(i) 采样法: 关注概率分布的期望.

MCMC 采样 (Metropolis-Hastings 算法). → 得到平稳分布.

↓ Gibbs 采样 (MH 的特例).

(ii) 变分推断: 使用确定性近似得到近似推断.

(已知简单分布) → 得到局部最优, 但有确定解的
近似后验分布 (EM).

最重要的是考虑如何对隐变量进行拆解, 以及假设各变量簇服从何种分布.

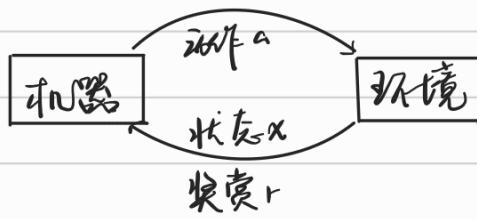
14.4 话题模型

话题模型是一类生成式有向图模型, 主要用来处理离散型的数据集
合 (如文本). 作为一种非监督生成式模型, 话题模型能够有效利用海量
数据发现文档集合中隐含的语义. 代表: 隐狄里克雷分布模型 (LDA).

LDA 基本单元: 词、文档、主题.

Chapter 16 强化学习 reinforcement learning

16.1 什么是强化学习



RL 常用马尔可夫决策过程描述 (MDP):

- 机器所在的环境 E
- 状态空间 X : $x \in X$ 是机器感知到的环境的描述.

- 智能体所采取的行为空间 A
- 潜在的状态转移函数 $P: X \times A \times X \rightarrow \mathbb{R}$.
- 潜在的奖赏函数 $R: X \times A \times X \rightarrow \mathbb{R}$
- 策略: $\pi: X \rightarrow A$

RL 定义了四元组 $\langle X, A, P, R \rangle$.

目标: 机器通过环境不断尝试从而学到一个策略 π , 使得长期执行该策
略后得到的期望累积奖赏最大.

T 步期望累积奖赏: $E\left[\frac{1}{T} \sum_{t=1}^T r_t\right]$

γ 期望折扣累积奖赏: $E\left[\sum_{t=0}^{+\infty} \gamma^t r_{t+1}\right]$

$$a = \pi(x) \quad \langle x_0, a_0, r_0, x_1, a_1, r_1, \dots, x_{T-1}, a_{T-1}, r_{T-1}, x_T \rangle.$$

$$\text{最大化 } E\left[\frac{1}{T} \sum_{t=1}^T r_t\right]$$

∠ RL: 没有有标记样本, 通过执行动作之后的反馈的奖赏来学习.
RL在某种意义上可以认为是具有“延迟标记信息”的监督学习.
RL的样本可来自于与环境的交互过程.

16.2 K-摇臂赌博机

{ 只有一个状态, K个动作
每个摇臂的奖赏服从斯期望未知的分布
执行有限次动作
最大化累积奖赏

探索-利用窘境

{ 探索: 估计不同摇臂的优劣
利用: 选择当前最优的摇臂.

在探索与利用之间折中:

① **ε贪心**: 以 ϵ 概率探索均匀随机选取一个摇臂; 以 $1-\epsilon$ 概率利用选择当前平均奖赏最高的摇臂.

② **Softmax**: 基于当前已知的摇臂平均奖赏来对探索和利用折中.

若某个摇臂当前的平均奖赏越大, 则它被选择的概率越高
概率分配已使用 Boltzmann 分布.

16.3 有模型学习

$$E = \langle X, A, P, R \rangle \text{ 均已知}$$

策略评估: 找到使累积奖赏最大的策略 π .

$$\begin{cases} V_T^\pi(x) = E_\pi \left[\frac{1}{T} \sum_{t=1}^T r_t \mid x_0 = x \right] \\ V_T^\pi(x) = E_\pi \left[\sum_{t=0}^{+\infty} \gamma^t r_{t+1} \mid x_0 = x \right] \end{cases}$$

状态-动作值函数:

$$\begin{cases} Q_T^\pi(x, a) = E_\pi \left[\frac{1}{T} \sum_{t=1}^T r_t \mid x_0 = x, a_0 = a \right] \\ Q_T^\pi(x, a) = E_\pi \left[\sum_{t=0}^{+\infty} \gamma^t r_{t+1} \mid x_0 = x, a_0 = a \right] \end{cases}$$

值函数的递归形式 (Bellman 方程):

$$V_T^\pi(x) = \sum_{a \in A} \pi(a|x) \sum_{x' \in X} P_{x \rightarrow x'}^a \left(\frac{1}{T} R_{x \rightarrow x'}^a + \frac{T-1}{T} V_{T-1}^\pi(x') \right)$$

$$V_T^{\pi}(x) = \sum_{a \in A} \pi(x, a) \sum_{x' \in X} P_{x \rightarrow x'}^a (R_{x \rightarrow x'}^a + \gamma V_T^{\pi}(x')).$$

$$Q_T^{\pi}(x, a) = \sum_{x' \in X} P_{x \rightarrow x'}^a \left(\frac{1}{T} R_{x \rightarrow x'}^a + \frac{T-1}{T} V_{T-1}^{\pi}(x') \right)$$

$$Q_T^{\pi}(x, a) = \sum_{x' \in X} P_{x \rightarrow x'}^a (R_{x \rightarrow x'}^a + \gamma V_T^{\pi}(x'))$$

最大化累积奖赏: $\pi^* = \arg \max_{\pi} \sum_{x \in X} V^{\pi}(x)$.

评估- π 策略的值函数:

$$V^{\pi}(x) = R(x, \pi(x)) + \gamma \sum_{x'} P_{x \rightarrow x'} V^{\pi}(x')$$

求解方法:

$$V_t^{\pi}(x) = R(x, \pi(x)) + \gamma \sum_{x'} P_{x \rightarrow x'} V_{t-1}^{\pi}(x').$$

最优值函数 / 最优策略满足 (最优 Bellman 方程):

$$\left\{ \begin{array}{l} V_T^*(x) = \max_{a \in A} \sum_{x' \in X} P_{x \rightarrow x'}^a \left(\frac{1}{T} R_{x \rightarrow x'}^a + \frac{T-1}{T} V_{T-1}^*(x') \right) \\ V_T^*(x) = \max_{a \in A} \sum_{x' \in X} P_{x \rightarrow x'}^a (R_{x \rightarrow x'}^a + \gamma V_T^*(x')) \end{array} \right.$$

↓

$$V^*(x) = \max_{a \in A} Q^{\pi^*}(x, a).$$

$$\pi^*(x) = \arg \max_{a \in A} Q^{\pi^*}(x, a).$$

RL可以归结为基于 dp 的最优问题. 与监督学习不同, 在有模型学习时, 我们不关注策略的泛化能力, 而是通过 dp 的方式为每一个状态找到最好的动作.

16.4 免模型学习

困难: ① 策略无法评估

② 无法通过值函数计算状态-动作值函数.

③ 机器只能从一个起始状态开始探索环境.

方法: ① 多次采样

② 直接估计每一对状态-动作的值函数

③ 在探索过程中逐渐发现各个状态

高特卡罗RL：采样轨迹，用样本均值近似期望

① 可能的问题：轨迹的单一性

解决方法： ϵ -贪心法（同策略、异策略）。

② 可能的问题：低效

解决方法：时序差分学习 (TD)：增量地进行状态-动作值函数更新。

ϵ -贪心法：

{ 同策略：Sarsa 算法
异策略： α -learning }

16.5 值函数近似

Q: 若状态空间连续(无限)怎么办？ \rightarrow 值函数近似

线性近似：将值函数表达为状态的线性函数。 $V_\theta(x) = \theta^T x$

用最小二乘法度量学到的值函数与真实的值函数 V^* 的逼近程度。

$$\mathcal{E}_\theta = \mathbb{E}_{x \sim \pi} [(V^*(x) - V_\theta(x))^2]$$

用梯度下降法更新参数向量，求解优化问题。

非线性值函数近似：核方法、神经网络。

16.6 一些首向

模仿学习：直接模仿人类专家的状态-动作对来学习策略。

引入了监督信息来学习策略。

直接模仿学习。

递强化学习：奖赏函数的设计！

迭代式递强化学习。