模式识别 第二十周读书报告

第十章 信息论与决策树

10.1 信息论基础

熵是信息论中的核心概念，熵是对信息中的不确定性（或不可预见性）的一种度量。如果要传输的符号不是近似的（也就是一种无损编码）又未被压缩，那么熵提供了一个理论上的最短可能的二进制编码的极限。

信息熵的定义公式为：

$$H(X) = -\sum_{x \in \chi} p(x) \log p(x)$$

H(X)是最短可能的平均或期望编码长度。

10.2 条件熵和联合熵

两个离散随机变量的联合熵定义为：

$$H(X,Y) = -\sum_{x}\sum_{y} p(x,y) \log_2 p(x,y) = -\mathbb{E}_{(X,Y)}[\log_2 p(X,Y)],$$

Y 在 X=x 的条件下的分布的熵为：

$$H(Y|X=x) = -\sum_{y} p_{Y|X=x}(y|x) \log_2 p_{Y|X=x}(y|x)$$

条件熵定义为：

$$H(Y|X) = \sum_{x} p(x) H(Y|X=x) = -\sum_{x,y} p(x,y) \log_2 p(y|x),$$

这是对所有 x 的 H(Y|X=x) 的加权平均。

联合熵为：

$$H(X,Y) = H(X) + H(Y|X)$$

10.3 互信息

在通常情况下 H(X|Y) != H(Y|X)

但是我们有：H(X)-H(X|Y)=H(Y)-H(Y|X)。这个差值可以被当为 X 和 Y 所共有的信息量。X 和 Y 的互信息定义为：

$$\begin{aligned} I(X;Y) &= \sum_{x,y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \\ &= \mathbb{E}_{(X,Y)} \left[\log_2 \frac{p(X,Y)}{p(X)p(Y)} \right] \\ &= I(Y;X). \end{aligned}$$

互信息是对称的，并且可以证明

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y). \end{aligned}$$

10.4 KL 散度和相对熵

两个分布之间的差异通过 KL 散度衡量。对两个（相同定义域）概率质量函数 p(x)和 q(x)而言，KL 散度定义如下：

$$\mathrm{KL}(p\|q) = \sum_x p(x) \log_2 \frac{p(x)}{q(x)}.$$

KL 散度也被称为相对熵，我们有

$$I(X;Y) = \mathrm{KL}\left(p(x,y)\|p(x)p(y)\right).$$

KL 散度的非负性有许多含义和推论：

1、 互信息 I(X;Y)非负。当且仅当 X 和 Y 独立时 I(X;Y)为 0;

2、 U 为有 m 个事件的均匀分布。那么$KL(X||U) = \sum_x p(x) log_2 \frac{p(m)}{\frac{1}{m}} = log_2 m - H(x)$。对任意 x 有

$$H(X) = \log_2 m - \mathrm{KL}(X\|U),$$
$$0 \leqslant H(X) \leqslant \log_2 m.$$

3、 $H(X) \geq H(X|Y)$;

4、 $H(X) + H(Y) \geq H(X,Y)$。

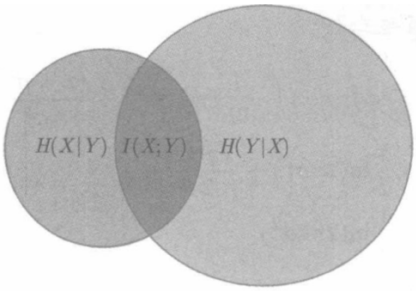


图 10.2 熵、条件熵和互信息之间的关系 (见彩插)

10.5 连续分布的信息论

概率密度为 p(x)的连续随机变量 X，它的微分熵为

$$h(X) = -\int p(x) \ln p(x)\, \mathrm{d}x.$$

令$p(x) = N(x;\mu,\sigma)$为一个正态分布的概率密度函数，它的微分熵为

$$\begin{aligned} h(X) &= -\int p(x) \left(-\frac{1}{2}\ln(2\pi\sigma^2) - \frac{(x-\mu)^2}{2\sigma^2}\right) \mathrm{d}x \\ &= \frac{1}{2}\ln(2\pi\sigma^2) + \frac{\mathrm{Var}(X)}{2\sigma^2} \\ &= \frac{1}{2}\ln(2\pi e\sigma^2). \end{aligned}$$

标准正态分布的熵为$\frac{1}{2}\ln(2\pi e)$。

(X,Y) 的联合熵定义为 $2\pi e$

$$h(X,Y) = -\int p(x,y) \ln p(x,y)\, \mathrm{d}x\, \mathrm{d}y,$$

这还可以扩展到多个随机变量. 相似地, 条件微分熵定义为

$$h(X|Y) = -\int p(x,y) \ln p(x|y)\, \mathrm{d}x\, \mathrm{d}y.$$

以下等式对条件熵也成立

$$h(X|Y) = h(X,Y) - h(Y) \quad 和 \quad h(Y|X) = h(X,Y) - h(X).$$

X 和 Y 之间的互信息为

$$I(X;Y) = \int p(x,y) \ln \frac{p(x,y)}{p(x)p(y)} \, \mathrm{d}x \, \mathrm{d}y,$$

我们仍然有

$$\begin{aligned} I(X;Y) &= h(X) - h(X|Y) \\ &= h(Y) - h(Y|X) \\ &= h(X) + h(Y) - h(X,Y). \end{aligned}$$

对两个概率密度函数 f(x)和 g(x)，在相同域下的 KL 距离（KL 散度，或相对熵） 定义为

$$\mathrm{KL}(f \| g) = \int f(x) \ln \frac{f(x)}{g(x)} \, \mathrm{d}x.$$

多元高斯分布是所有均值存在且协方差矩阵是Σ的分布中熵最大的分布。

10.6 最小交叉熵

在一个 m 分类问题中，会学习 m 个线性方向，每个类别对应一个线性方向β_i.对一个任意样例 x，$x^T\beta_i$通过 softmax 变换构成一个概率估计

$$\Pr(y = j|\boldsymbol{x}) \approx f^j(\boldsymbol{x}) = \frac{\exp(\boldsymbol{x}^T\boldsymbol{\beta}_j)}{\sum_{j'=1}^{m} \exp(\boldsymbol{x}^T\boldsymbol{\beta}_{j'})},$$

其中$f^j(x)$ 是对 x 属于第 j 个类的概率估计。在有 n 个训练样例的问题中，softmax 回归会最大化如下目标函数

$$\sum_{i=1}^{n} \sum_{j=1}^{m} [\![y_i = j]\!] \log_2 f^j(\boldsymbol{x}_i),$$

若 p 是由训练集观察得到的概率$p_{ij} = [\![yi = j]\!]$，q 是由模型估计出的概率$q_{ij} = f^j(xi)$.那么最大化上述目标函数等价于最小化

$$\mathrm{CE}(p,q) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij} \log_2 q_{ij}.$$

因此，多项对数几率回归旨在最小化交叉熵。

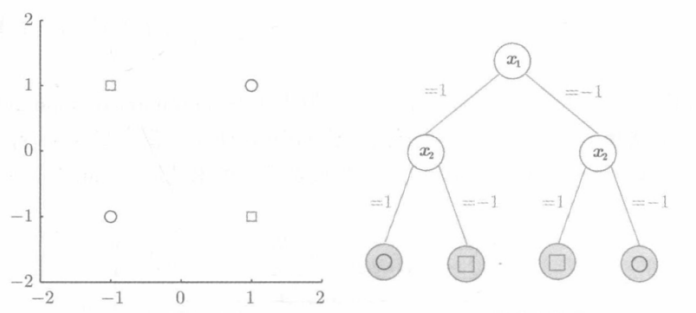
10.7 决策树

决策树又称判定树，是数据挖掘中的一种重要分类方法，它是一种树结构形式来表达的预测分析模型。

通过把实例从根节点排列到某个叶子结点来分类实例;

叶子结点即为实例所属的分类;

树上每个节点说明了对实例的某个属性的测试结点的每个后继分支对应于该属性的一个可能值。



(a) 异或问题　　(b) 异或决策树

图 10.3　异或问题及其决策树模型 (见彩插)

我们需要解决如下的问题

1、 如何选择内部结点的划分准则

2、 何时停止划分？树的深度过大，决策树将过拟合，过浅的决策树将欠拟合

3、 决策树将如何处理实数值变量呢

4、 如果输入的特征向量有许多维，并且每维只是对分类有很小的作用，只用一维进行结点的划分的决策树是不够的

5、 存在一些单一决策树很难解决的问题

10.8 基于信息增益的结点划分

一个系统越是有序，信息熵就越低；反之，一个系统越是混乱，信息熵就越高。信息熵也可以说是系统有序化程度的一个度量。即熵越大，随机变量的不确定性越大。

分类错误率会随着某次结点的划分而降低，这是因为训练样例的标记分布将变得更集中。在信息论中，我们可以称新结点的熵都比根节点小。因此，信息增益准则试图找到一个使得熵的变化最大的划分。

信息增益度量了预测的不确定性减少的程度，定义为

$$H(T) - \sum_{i=1}^{K} w_i H(T_i),$$

即上一层的熵减去当前一层熵的总和。信息增益大的特征具有更强的分类能力。

在内部结点，我们可以测试所有特征并且使用具有最大信息增益的那个特征来进行划分。如果我们使用随机变量 L 来表示分类的标记，并用 F 表示用于信息增益的特征，那么$\sum_{i=1}^{K} w_i H(T_i)$其实是使用训练集计算到的条件熵 H(L|F)。因此信息增益为

$$H(L) - H(L|F) = I(L;F),$$

第 11 章 稀疏数据和未对齐数据

11.1 稀疏机器学习

给定一个向量 x，稀疏机器学习通常将 x 转换为一个新的表示 y，且其学习过程保证这个新的表示 y 是稀疏的。

稀疏 PCA

为了使得 PCA 表示的y_i是稀疏的，我们需要一个正则化项。例如我们可以求解如下的优化的问题

$$\min_{\boldsymbol{y}_i} \|\boldsymbol{y}_i - E_d^T(\boldsymbol{x}_i - \bar{\boldsymbol{x}})\|^2 + \lambda\|\boldsymbol{y}_i\|_0,$$

$||y_i||_0$是ℓ_0范数，由于$||y_i||_0$是y_i中非零元素的个数，最小化这一项会促使其解有许多零元素，即变得稀疏。

由ℓ_1范数诱导稀疏性

正则化项$||x||_1$也可以促使解的元素趋于 0，因此我们经常将 L-0 约束放宽为 L-1 约束。ℓ_1范数对优化友好，它的定义为

$$\|\boldsymbol{x}\| = \sum_{i=1}^{d} |x_i|,$$

字典学习和稀疏编码问题

输入向量 x 与其新的表示α有近似线性关系联系起来

$$\boldsymbol{x} \approx D\boldsymbol{\alpha},$$

其中 D 为分块矩阵的形式，写作

$$D = [\boldsymbol{d}_1|\boldsymbol{d}_2|\ldots|\boldsymbol{d}_k],$$

那么，我们有

$$\boldsymbol{x} \approx \sum_{i=1}^{k} \alpha_i \boldsymbol{d}_i,$$

给定训练集，我们可以将所有训练样例组合成一个 p*n 的矩阵

$$X = [\boldsymbol{x}_1|\boldsymbol{x}_2|\ldots|\boldsymbol{x}_n].$$

同样，他们的新表示也可以组合为一个新矩阵

$$A = [\boldsymbol{\alpha}_1|\boldsymbol{\alpha}_2|\ldots|\boldsymbol{\alpha}_n] \in \mathbb{R}^k \times \mathbb{R}^n.$$

所有样例的线性近似误差构成了一个大小为 p*nd 新矩阵 X-DA

用下式度量近似误差

$$\sum_{i=1}^{p}\sum_{j=1}^{n}[X - DA]_{ij}^2,$$

其可以写作 $\|X - DA\|_F^2$. 对一个 $m \times n$ 的矩阵 X, $\|X\|_F$ 是矩阵的 Frobenius 范数 (Frobenius norm)

$$\|X\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}^2} = \sqrt{\mathrm{tr}(XX^T)}.$$

如果 D 已知，并我们希望重构是稀疏的，我们求解如下的优化问题

$$\min_{\boldsymbol{\alpha}_i} \|\boldsymbol{x}_i - D\boldsymbol{\alpha}_i\|^2 + \lambda\|\boldsymbol{\alpha}_i\|_1.$$

但是 D 是未知的，需要从数据 X 中学到。D 被称为字典，也就是说我们使用字典项的加权组合来近似任意样例。当 X 是满秩的时候，字典不足以完全描述 x_i,也就是说这时的字典是欠完备的。

通常使用过完备的字典，即（p<k)。字典学习的问题可以被形式化的建模为

$$\begin{aligned}\min_{D,A} \quad & \sum_{i=1}^{n}\left(\|\boldsymbol{x}_i - D\boldsymbol{\alpha}_i\|_F^2 + \lambda\|\boldsymbol{\alpha}_i\|_1\right)\\ \text{s.t.} \quad & \|\boldsymbol{d}_j\| \leqslant 1, \forall\, 1 \leqslant j \leqslant k.\end{aligned}$$

如同 PCA 一样，我们不能让字典项无限延长，因此需要添加下面的约束项。
上述优化问题还可以写作:

$$\begin{aligned}\min_{D,A} \quad & \|X - DA\|_F^2 + \lambda\|\mathrm{vec}(A)\|_1\\ \text{s.t.} \quad & \|\boldsymbol{d}_j\| \leqslant 1, \forall\, 1 \leqslant j \leqslant k.\end{aligned}$$

其中 vec(A)是 A 向量化的结果。

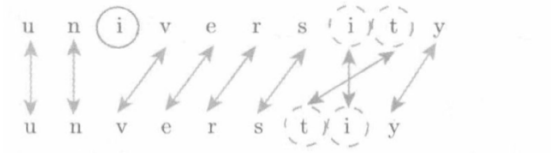
11.2 动态时间调整

以股票价格的变迁为例，我们可以每隔固定一段时间采样某只股票在一天中的价格，从而用一个固定长度的向量表示一天中的股票价格变化。然而我们很难说两个不同的日子的数据（即两天的特征向量）中的相同维度表示相同含义。

再进行对齐操作后，两个向量的相同维度确实表示（或近似表示）相同的含义，随后我们就可以像往常一样计算两个样例的相似性或者距离。

未对齐的时序数据

时序数据是指包含了一组有序元素的数据类型，在现实世界中许多数据是有序的，如英语单词。



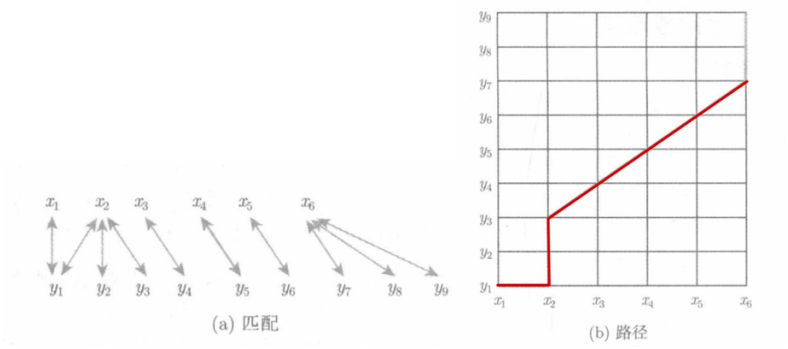
university 和 unverstiy 的对齐 (见彩插)

动态时间调整 DTW

令 x=(x1,x2,⋯,xn),y=(y1,y2,⋯,ym),序列不一定需要有同样的长度。
什么是一个好的匹配?
1、 如果 xi 和 yi 匹配，那么 d(xi,yi)应该很小;
2、 在匹配过程中，x 和/或 y 中的一些元素可以跳过;
3、 我们应当选择总距离最小的匹配

首先，如果每个元素都被跳过，那么总距离就是 0，DWT 对这种情况的补救措施是约束 x(y)中的每个元素都在 y(x)中有一个匹配元素。第二，如果两对匹配的元素发生交叉，优化问题的搜索空间将非常大，因此优化问题会很困难。

针对上述问题 DWT 使用如下准则替换上述第二条准则
1） 每个 xi 和 yi 都必须有一个匹配元素
2） 匹配必须按顺序进行



一条路径可以被一系列坐标(rk,tk),(k=1,2,⋯,K)完全指定。

匹配必须按顺序进行并且每个元素都在匹配中，那么只有以下三种情况：

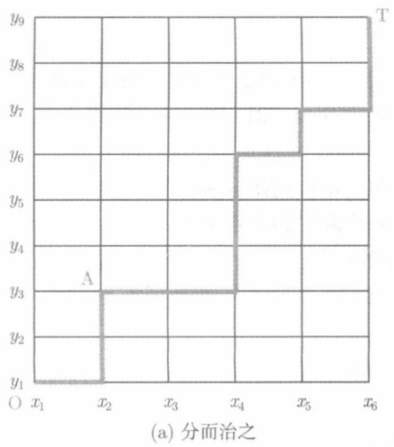
$$\begin{aligned}
&\#1:\ r_{k+1}=r_k,\quad t_{k+1}=t_k+1 && \text{(路径向上)};\\
&\#2:\ r_{k+1}=r_k+1, t_{k+1}=t_k && \text{(路径向右)};\\
&\#3:\ r_{k+1}=r_k+1, t_{k+1}=t_k+1 && \text{(路径向右上)};
\end{aligned}$$

对于选择总距离最小的匹配，只需求解如下的优化问题（Ω为满足约束条件的所有路径组成的集合，D(n,m)为最小总匹配距离）。

$$D(n,m)=\min_{(\boldsymbol{r},\boldsymbol{t})\in\Omega}\sum_{k=1}^{K_{(\boldsymbol{r},\boldsymbol{t})}} d(x_{r_k},y_{t_k}).$$

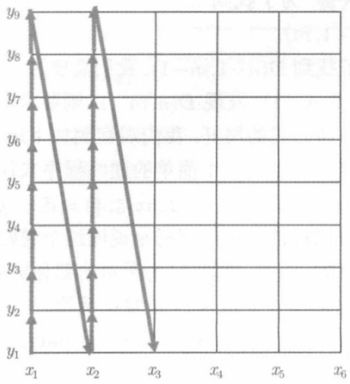
动态规划

分而治之是一种减少 DTW 复杂度有用的策略。如下图所示，若可以知道最优路径是由从 O 到 A 的最优路径的距离和从 A 到 T 的最优距离之和，那么原本的大问题就被分为了两个小问题。



然而使用分治法求解 DTW 存在问题。

动态规划策略通过枚举所有可能的 A 的候选来解决第一个困难，并经常通过把一个大问题分成一个非常小的问题以及另一个问题来解决困难。



(b) 动态规划

A 到 T 的移动有如下三种情况

- $A = (x_{n-1}, y_{m-1})$, 移动是向右上的 (即沿着对角线的). 那么, 我们有

$$D(n,m) = D(n-1,m-1) + d(x_n, y_m),$$

  其中 $D(n-1,m-1)$ 是从 O 到 (x_{n-1}, y_{m-1}) 的最优距离.
- $A = (x_n, y_{m-1})$, 移动是向上的. 那么, 我们有

$$D(n,m) = D(n,m-1) + d(x_n, y_m).$$

- $A = (x_{n-1}, y_m)$, 移动是向右的. 那么, 我们有

$$D(n,m) = D(n-1,m) + d(x_n, y_m).$$

综合上述三种情况，我们有

$$D(u,v) = d(x_u, y_v) + \min\{D(u-1,v), D(u,v-1), D(u-1,v-1)\}.$$

上式描述了一个大问题和一个小问题之间的递归关系。

动态时间规整算法

算法 11.1 动态时间规整算法

输入: 两个序列 $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ 和 $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$
$D(1,1) = d(x_1, y_1)$
for $j = 2$ to m **do**
  $D(1,j) = d(x_1, y_j) + D(1,j-1)$
end for
for $i = 2$ to n **do**
  $D(i,1) = d(x_i, y_1) + D(i-1,1)$
  for $j = 2$ to m **do**
    $D(i,j) = d(x_i, y_j) + \min\{D(i-1,j-1), D(i,j-1), D(i-1,j)\}$
  end for
end for