Summary for the final exam

Joseph Chen

January 22, 2018

1 traditional PR

1.1 模式

- 定义: 广义地说，存在于时间和空间中可观察的物体，如果我们可以区别它们是否相同或是否相似，都可以称之为模式。模式所指的不是事物本身，而是从事物获得的信息，因此，模式往往表现为具有时间和空间分布的信息。
- 直观特性: 可观察性、可区分性、相似性
- 模式识别的分类：监督学习、概念驱动或归纳假说；非监督学习、数据驱动或演绎假说。
- 模式分类的主要方法：数据聚类、统计分类、结构模式识别、神经网络

1.2 数学预备

- 贝叶斯公式 posterior $\propto$ likelihood $\times$ prior
- 如何从联合分布 (joint distribution) 导出边缘分布 (marginal distribution)? **A:** 对某个随机变量积分或者求和,e.g. $p(a,b) = \sum_c p(a,b,c)$
- 多维高斯分布的概率密度

$$N(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\Sigma})\right\}$$

- 指数函数分布族 (exponential family), 包含高斯, 二项分布 etc. 共轭先验: 给定一个 likelihood, 我们可以找到一个具有相同函数形式的 prior, 从而 posterior 也具有相同的函数形式.
- 拉格朗日乘数法, 用在 (凸) 优化问题求解, 原问题转换成对偶形式.

- 数学期望和方差

$$
\begin{aligned}
E(x) &= \sum_x p(x)x \text{ or } E(x) = \int p(x)x\mathrm{d}x \\
var(x) &= E([x-E(x)]^2) = E(x^2) - [E(x)]^2 \\
cov(x,y) &= E_{x,y}((x-E(x))(y-E(y))) \\
&= E_{x,y}(xy) - E(x)E(y) \\
cov(\boldsymbol{x},\boldsymbol{y}) &= E_{\boldsymbol{x},\boldsymbol{y}}[(\boldsymbol{x}-E(\boldsymbol{x}))(\boldsymbol{y}^T - E(\boldsymbol{y}^T))] \\
&= E_{\boldsymbol{x},\boldsymbol{y}}[\boldsymbol{x}\boldsymbol{y}^T] - E[\boldsymbol{x}]E[\boldsymbol{y}^T]
\end{aligned}
$$

1.3 判别函数

线性分类的 3 种情况:

- 多类情况 1: M 类需要 M 个判别函数, 分界面区分 C_i or notC_i. e.g. $d_i(x) > 0$ 其他小于 0 则 x 属于第 i 类;
- 多类情况 2: M 类需要 $M(M-1)/2$, 分界面区分开 C_i 和 C_j, 若 $\forall d_{ij}(x) > 0, j \neq i$, 则 x 属于类别 i.
- 多类情况 3: M 类需要 M 个判别函数, 取最大的判别函数的下标作为类别

三种学习参数的方法: 最小二乘法 (LMS), Fisher 准则, 感知机

- LMS: l_2 norm loss, 梯度下降或者解析解
- Fisher 准则 (有计算): 思想就是将原数据从 D 维投影到 1 维, 然后找到一个线性分界面, 投影方程

  $$y = \boldsymbol{w}^T\boldsymbol{x}$$

  以二分类为例, 求出均值向量

  $$\boldsymbol{m}_1 = \frac{1}{N}\sum_{n\in\mathcal{C}_1}\boldsymbol{x}_n, \boldsymbol{m}_2 = \frac{1}{N}\sum_{n\in\mathcal{C}_2}\boldsymbol{x}_n$$

  最大化 Fisher 准则

  $$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T\boldsymbol{S}_b\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{S}_w\boldsymbol{w}}$$

  得到

  $$\boldsymbol{w} \propto \boldsymbol{S}_w^{-1}(\boldsymbol{m_2} - \boldsymbol{m_1})$$

  然后分界面就是

  $$\boldsymbol{w}^T\boldsymbol{x} - y_0 = 0$$

  这里 y_0 取值

  $$y_0 = \frac{1}{2}(\boldsymbol{w}^T\boldsymbol{m}_1 + \boldsymbol{w}^T\boldsymbol{m}_2)$$

类间协方差矩阵

$$S_b = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T$$

类内协方差矩阵

$$S_w = \sum_{n \in \mathcal{C}_1} (\boldsymbol{x}_n - \boldsymbol{m}_1)(\boldsymbol{x}_n - \boldsymbol{m}_1)^T + \sum_{n \in \mathcal{C}_2} (\boldsymbol{x}_n - \boldsymbol{m}_2)(\boldsymbol{x}_n - \boldsymbol{m}_2)^T$$

- 感知机算法: 分类器

  $$h_\theta(x) = \mathrm{sign}(\theta^T x)$$

  参数更新 (SGD)

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

基于最小错误率和最小风险的贝叶斯决策

- 最小错误率的情况: given $P(w_1)$ 和 $P(w_2)$, $P(x|w_1)$ 和 $P(x|w_2)$, 求出后验概率 $P(w_1|x)$ 和 $P(w_2|x)$, 基于最小错误率的决策, $P(w_1|x) > P(w_2|x)$ 则 $x \in w_1$.
- 最小风险: $g_1(x) = \lambda_{11}P(w_1|x) + \lambda_{12}P(w_2|x)$, $g_2(x) = \lambda_{21}P(w_1|x) + \lambda_{12}P(w_2|x)$, 选择风险小的

2 KL 变换/PCA

given N 个样本 $\boldsymbol{x}_i$, 求其 KL 变换的步骤

- 计算样本的均值 $\boldsymbol{\mu}$, 所有样本减去均值. (中心平移到原点)
- 计算协方差矩阵 $R = \frac{1}{N}\sum_{i=1}^N \boldsymbol{x}_i \boldsymbol{x}_i^T$
- 求协方差矩阵的特征向量和对应的特征值, 根据要求选择前面的特征向量 (依据特征值大小排序), 构成变换矩阵 U
- U 与原始数据 $\boldsymbol{x}$ 相乘即得到变换后的数据

3 Supervised learning

3.1 几个关系

给定一个训练集 $\{\boldsymbol{x}, \boldsymbol{t}\}$, LMS 可由最大化似然函数 $p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}, \beta)$ 导出,

$$p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}, \beta) = \prod_{n=1}^N \mathcal{N}(t_n | y(x_n, \boldsymbol{w}), \beta^{-1})$$

$$\ln p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}, \beta) = -\frac{\beta}{2}\sum_{n=1}^N \{y(x_n, \boldsymbol{w}) - t_n\}^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi)$$

正则化系数可以通过最大化后验概率得出

$$p(\boldsymbol{w}|\boldsymbol{x},\boldsymbol{t}) \propto p(\boldsymbol{t}|\boldsymbol{x},\boldsymbol{w})p(\boldsymbol{w})$$

设先验

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|0,\alpha^{-1}\boldsymbol{I}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\alpha^{-1}\boldsymbol{I}|^{1/2}}\exp\{-\frac{1}{2}\boldsymbol{w}^T(\alpha^{-1}\boldsymbol{I})^{-1}\boldsymbol{w}\}$$

最大化 $p(\boldsymbol{w}|\boldsymbol{x},\boldsymbol{t})$ 等价于

$$\text{maximize } -\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n,\boldsymbol{w})-t_n\}^2-\frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w}$$

因此 $\lambda=\alpha/\beta$. 完全的贝叶斯观念是将 $\boldsymbol{w}$ 视为一个随机变量, 在整个参数空间积分得到预测分布

$$p(t|x,\boldsymbol{x},\boldsymbol{t}) = \int p(\boldsymbol{w}|\boldsymbol{x},\boldsymbol{t})p(t|x,\boldsymbol{w})\mathrm{d}\boldsymbol{w}$$

3.2 偏差方程的分解

定义几个符号: 给定训练集 D, 训练得到的预测函数 $f_D(x)$, 最优估计 $h(x)$.

$$\begin{aligned}
E_D(L) &= \int\int(f_D(x)-y)^2p(x,y)\mathrm{d}x\mathrm{d}y\\
&= \int\int(f_D(x)-h(x)+h(x)-y)^2p(x,y)\mathrm{d}x\mathrm{d}y\\
&= \int(f_D(x)-h(x))^2p(x)\mathrm{d}x+\int\int(h(x)-y)^2p(x,y)\mathrm{d}x\mathrm{d}y\\
&= \int(f_D(x)-E[f_D(x)]+E[f_D(x)]-h(x))^2p(x)\mathrm{d}x+\int\int(h(x)-y)^2p(x,y)\mathrm{d}x\mathrm{d}y\\
&= \int(f_D(x)-E[f_D(x)])^2p(x)\mathrm{d}x+\int(E[f_D(x)]-h(x))^2p(x)\mathrm{d}x+\int\int(h(x)-y)^2p(x,y)\mathrm{d}x\mathrm{d}y
\end{aligned}$$

上式中第一项为 variance, 第二项为 bias^2, 第三项为 noise.

3.3 SVM

见单独的总结

4 图模型

条件独立性的判断和证明.

5 独立于算法的机器学习

5.1 some philosophy

- No Free Lunch Theorem: 不存在一个与具体应用无关的、普遍适用的“最有分类器/回归器”! 仅在学习算法与问题匹配的情况下才 (即在特定的实际问题或目标函数) 才有所谓“更优”!
- Ugly Duckling Theorem: 世界上不存在分类的客观标准，一切分类标准都是主观的。
- Occam’s Razor: 如无必要, 勿增实体。在相互竞争的假设中, 我们选择条件最少的假设。在模式识别领域, 在拟合数据程度接近的情况下, 我们更加偏向于选择简单的算法或者分类器。

5.2 重采样技术

5.2.1 sampling

Jackknife – 刀切法, leave one out
Bootstrap – 随机选取 n 个点, 重新给予权重.

5.2.2 Bagging

independently bootstrap data sets. 给定一个数据集 D, Bagging 算法如下:

- 独立地采样 m 个子集 $D_1, D_2, ..., D_m$
- 每个子集上训练一个分类器 f_i
- 最后的分类器结果由所有的投票决定

5.2.3 Boosting

dependently bootstrap data sets 举个例子,

- D_1 是随机地从原数据集 X 中选取的一个子集, 训练一个分类器 h_1
- D_2 从剩余样本 X/D_1 中选取, 使得一半被 h_1 正确分类, 一半被 h_1 错误分类;
- D_3 从剩余样本 $X/(D_1 \cup D_2)$ 中选取, 使得 h_1 和 h_2 判决结果不同, 训练一个分类器 h_3
- 总的分类器

$$h(x) = h_1(x), \text{if } h_1(x) = h_2(x); \text{ otherwise } h(x) = h_3(x)$$

5.2.4 AdaBoost

AdaBoost 基本思想: 将弱分类器进行线性带权（权重可以理解为每个弱分类器对最终形成的强分类器的影响因袭）组成形成强分类器。继续训练可以增加 Margin（可以理解为正确分类的置信度），而 Margin 的增加可以降低泛化误差。因此，会导致测试误差下降。
AdaBoost 算法:
给定 N 个样本的训练集 $X = \{(\boldsymbol{x}_1, t_1), (\boldsymbol{x}_2, t_2), \cdots, (\boldsymbol{x}_N, t_N)\}$,$t_i \in \{+1, -1\}$.

1. 初始化样本的权重为 $w_n^{(1)} = 1/N$, $n = 1, 2, \cdots, N$

2. for $m = 1, 2, \cdots, M$:

    - 训练一个弱分类器 $y_m(\boldsymbol{x})$, 使其最小化带权重的误差函数

    $$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n)$$

    这里 $I()$ 是指示函数, $y_m(\boldsymbol{x}_n) \neq t_n$ 为真 $I = 1$ 否则 $I = 0$

    - 计算分类器的话语权

    $$\alpha_m = \ln \frac{1 - \epsilon_m}{\epsilon_m}$$

    这里 ϵ_m 由下式给出

    $$\epsilon_m = \frac{J_m}{\sum_{n=1}^{N} w_n^{(m)}}$$

    - 更新样本的权重

    $$w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(y_m(\boldsymbol{x}_n) \neq t_n)\}$$

3. 得到最后的分类器

$$Y_M(\boldsymbol{x}) = \mathrm{sign}(\sum_{m=1}^{M} \alpha_m y_m(\boldsymbol{x}))$$

6 EM 算法

6.1 K means

6.1.1 与 GMM 的差异

K-means 是 GMM 在协方差矩阵 $\epsilon\boldsymbol{I} \to \boldsymbol{0}$ 的特例, K-means 执行的是 “hard assignment”, 也就是在 E step 将每个样本粗暴地赋予到距离最近的那个类别, 而 GMM 执行的是 “soft assignment”, 对一个样本而言, 赋予其属于某个类别的概率.

6.1.2 算法

easy... 判断收敛的条件是中心不再移动

6.2 GMM

GMM 模型

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \sum_k \pi_k = 1, 0 \leq \pi_k \leq 1$$

对数似然函数

$$\ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{1}$$

无法求得解析解, 可以用梯度法进行数值优化, 这里采用 EM 算法进行最大化似然函数.
引入隐变量 (a latent variable)$\boldsymbol{z}$, $\boldsymbol{z}$ 满足 $z_k \in \{0, 1\}, \sum_k z_k = 1$

$$p(z_k = 1) = \pi_k$$

$$p(\boldsymbol{x}|z_k = 1) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

给定一个样本 $\boldsymbol{x}$, $\boldsymbol{x}$ 属于第 k 类的概率, 也称为 “responsibility”

$$\begin{aligned} \gamma(z_k) \equiv p(z_k = 1|\boldsymbol{x}) &= \frac{p(z_k = 1)p(\boldsymbol{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\boldsymbol{x}|z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \end{aligned}$$

令似然函数对 $\boldsymbol{\mu}_k$ 求偏导等于 0 得

$$\begin{aligned} 0 &= \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) \\ &= \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) \end{aligned}$$

因此

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n, N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

对 $\boldsymbol{\Sigma}_k$ 求偏导数等于 0 得

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T$$

类似可以推出

$$\pi_k = \frac{N_k}{N}$$

所以 GMM 条件下的 EM 算法如下:

1. 初始化参数 $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k

2. **E step** 用当前参数更新 responsibility $\gamma(z_{nk})$
3. **M step** 用当前 responsibility 更新 $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, $\boldsymbol{\pi}_k$
4. 判断算法是否收敛, 否则返回步骤 2