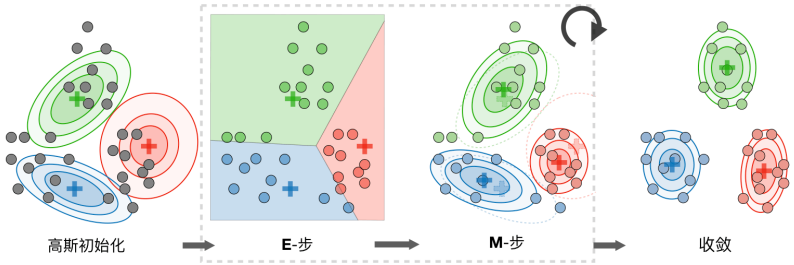
VIP Cheatsheet: Unsupervised Learning

Afshine Amidi and Shervine Amidi

October 13, 2018



翻译: 朱小虎

无监督学习导引

❒ **动机** – 无监督学习的目标是找到在未标记数据$\{x^{(1)},...,x^{(m)}\}$ 中的隐含模式。

❒ **Jensen 不等式** – 令f 为一个凸函数而X 为一个随机变量。我们有下列不等式:

$$\boxed{E[f(X)] \geqslant f(E[X])}$$

E-M 算法

❒ **隐变量** – 隐变量是隐含/不可观测的变量，使得估计问题变得困难，通常被表示成z。这里是包含隐变量的常见设定:

设定	隐变量z	$x\|z$	评论
k 元混合高斯分布	Multinomial(ϕ)	$\mathcal{N}(\mu_j,\Sigma_j)$	$\mu_j \in \mathbb{R}^n, \phi \in \mathbb{R}^k$
因子分析	$\mathcal{N}(0,I)$	$\mathcal{N}(\mu+\Lambda z,\psi)$	$\mu_j \in \mathbb{R}^n$

❒ **算法** – E-M 算法给出了通过重复构建似然函数的下界（E-步）和最优化下界（M-步）进行极大似然估计给出参数θ 的高效估计方法:

- E-步: 计算后验概率$Q_i(z^{(i)})$，其中每个数据点$x^{(i)}$ 来自特定的簇$z^{(i)}$，过程如下:

$$\boxed{Q_i(z^{(i)}) = P(z^{(i)}|x^{(i)};\theta)}$$

- M-步: 使用后验概率$Q_i(z^{(i)})$ 作为簇在数据点$x^{(i)}$ 上的特定权重来分别重新估计每个簇模型，过程如下:

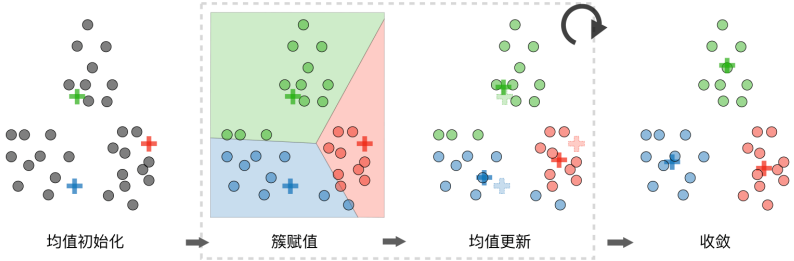
$$\boxed{\theta_i = \underset{\theta}{\text{argmax}} \sum_i \int_{z^{(i)}} Q_i(z^{(i)}) \log\left(\frac{P(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}\right) dz^{(i)}}$$

k-均值聚类

我们记$c^{(i)}$ 为数据点i 的簇，μ_j 是簇j 的中心。

❒ **算法** – 在随机初始化簇中心$\mu_1,\mu_2,...,\mu_k \in \mathbb{R}^n$ 后，k-均值算法重复下列步骤直至收敛:

$$\boxed{c^{(i)} = \underset{j}{\arg\min}||x^{(i)}-\mu_j||^2} \quad \text{et} \quad \boxed{\mu_j = \frac{\sum_{i=1}^m 1_{\{c^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{c^{(i)}=j\}}}}$$



❒ **失真函数** – 为了看到算法是否收敛，我们看看如下定义的失真函数:

$$\boxed{J(c,\mu) = \sum_{i=1}^m ||x^{(i)} - \mu_{c^{(i)}}||^2}$$

层次化聚类

❒ **算法** – 结合聚合层次化观点的聚类算法，按照逐次构建嵌套簇的方式进行。

❒ **类型** – 存在不同的层次化聚类算法，解决不同的目标函数优化问题，在下表中总结列出:

内链	均链	全链
最小化簇内距离	最小化簇对平均距离	最小化簇对的最大距离

聚类评测度量

在一个无监督学习设定中，通常难以评测一个模型的性能，因为我们没有像监督学习设定中那样的原始真实的类标。

❒ **Silhouette 系数** – 通过记a 和b 为一个样本和在同一簇中的其他所有点之间的平均距离和一个样本和在下一个最近簇中的所有其他点的平均距离，针对一个样本的Silhouette 系数s 定义如下：

$$\boxed{s = \frac{b-a}{\max(a,b)}}$$

❒ **Calinski-Harabaz 指标** – 通过记k 为簇的数目，B_k 和W_k 分别为簇间和簇内弥散矩阵，定义为：

$$B_k = \sum_{j=1}^{k} n_{c^{(i)}}(\mu_{c^{(i)}} - \mu)(\mu_{c^{(i)}} - \mu)^T, \qquad W_k = \sum_{i=1}^{m}(x^{(i)} - \mu_{c^{(i)}})(x^{(i)} - \mu_{c^{(i)}})^T$$

Calinski-Harabaz 指标$s(k)$ 表示一个聚类模型定义簇的好坏，分数越高，簇就越稠密和良好分隔。其定义如下：

$$\boxed{s(k) = \frac{\mathrm{Tr}(B_k)}{\mathrm{Tr}(W_k)} \times \frac{N-k}{k-1}}$$

主成分分析

这是一种维度降低的技巧，找到投影数据到能够最大化方差的方向。

❒ **特征值，特征向量** – 给定矩阵$A \in \mathbb{R}^{n\times n}$，$\lambda$ 被称为A 的一个特征值当存在一个称为特征向量的向量$z \in \mathbb{R}^n \backslash \{0\}$，使得：

$$\boxed{Az = \lambda z}$$

❒ **谱定理** – 令$A \in \mathbb{R}^{n\times n}$。如果$A$ 是对称阵，那么A 可以被一个实正交矩阵$U \in \mathbb{R}^{n\times n}$ 对角化。通过记$\Lambda = \mathrm{diag}(\lambda_1,...,\lambda_n)$ 我们有：

$$\boxed{\exists \Lambda \text{ 为对角阵}, \quad A = U\Lambda U^T}$$

注：关联于最大的特征值的特征向量被称为矩阵A 的主特征向量。

❒ **算法** – 主成分分析（PCA）过程就是一个降维技巧，通过最大化数据的方差而将数据投影到k 维上：

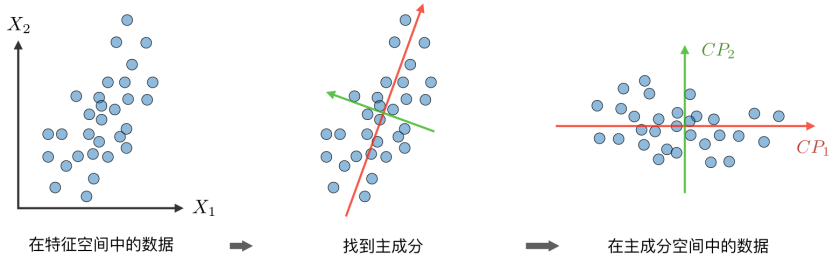
- 步骤1: 规范化数据使其均值为0 方差为1。

$$\boxed{x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{\sigma_j}} \quad \text{où} \quad \boxed{\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}} \quad \text{et} \quad \boxed{\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(x_j^{(i)} - \mu_j)^2}$$

- 步骤2: 计算$\Sigma = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} x^{(i)^T} \in \mathbb{R}^{n\times n}$，其为有实特征值的对称阵

- 步骤3: 计算Σ 的k 个正交的主特征向量$u_1,...,u_k \in \mathbb{R}^n$，即对应$k$ 个最大特征值的正交特征向量。

- 步骤4: 投影数据到$\mathrm{span}_{\mathbb{R}}(u_1,...,u_k)$ 上。这个过程最大化所有k 维空间的方差



独立成分分析

这是旨在找到背后生成源的技术。

❒ **假设** – 我们假设数据x 已经由n-维源向量$s = (s_1,...,s_n)$ 生成出来，其中s_i 是独立的随机变量，通过一个混合和非奇异矩阵A 如下方式产生：

$$\boxed{x = As}$$

目标是要找到去混合矩阵$W = A^{-1}$。

❒ **Bell-Sejnowski ICA 算法** – 该算法找出去混合矩阵W ，通过下列步骤：

- 记概率$x = As = W^{-1}s$ 如下：

$$p(x) = \prod_{i=1}^{n} p_s(w_i^T x) \cdot |W|$$

- 记给定训练数据$\{x^{(i)}, i \in [\![1,m]\!]\}$ 对数似然函数其中g 为sigmoid 函数如下：

$$l(W) = \sum_{i=1}^{m}\left(\sum_{j=1}^{n} \log\left(g'(w_j^T x^{(i)})\right) + \log|W|\right)$$

因此，随机梯度下降学习规则是，对每个训练样本$x^{(i)}$，我们如下更新W：

$$\boxed{W \longleftarrow W + \alpha\left(\begin{pmatrix} 1-2g(w_1^T x^{(i)}) \\ 1-2g(w_2^T x^{(i)}) \\ \vdots \\ 1-2g(w_n^T x^{(i)}) \end{pmatrix} x^{(i)^T} + (W^T)^{-1}\right)}$$