无监督学习[1]

简介

给定训练集$\left\{x^{(1)}, \cdots, x^{(m)}\right\}$，其中$x^{(i)} \in \mathbb{R}^n$一如既往，但是并没有给出对应的标签$y^{(i)}$，我们该如何将这些样本划分成不同的族（cluster）？像这种不给出标签，需要我们利用某种算法去给样本打上标签（聚类）的问题就称为无监督学习（unsupervised learning）。

1 k-means聚类算法

下面给出k-means算法

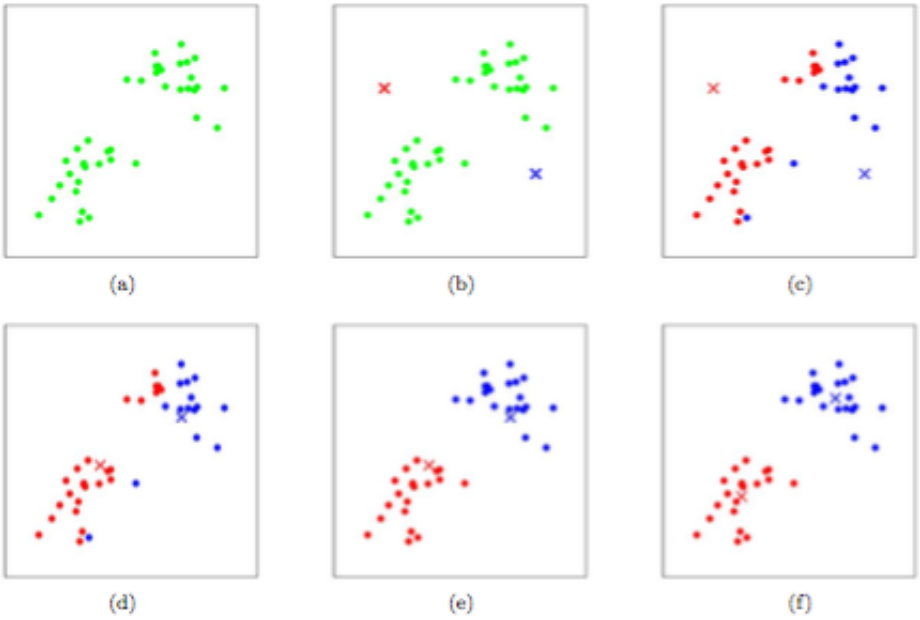
1. 随机初始化聚类质心（cluster contriods）$\mu_1, \cdots, \mu_k \in \mathbb{R}^n$
2. 重复以下过程直至收敛
    - For every i, set

$$c^{(i)} := \arg\min_j \|x^{(i)} - \mu_j\|^2$$

    - For every j, set

$$\mu_j := \frac{\sum_{i=1}^m 1\left\{c^{(i)} = j\right\} x^{(i)}}{\sum_{i=1}^m 1\left\{c^{(i)} = j\right\}}$$

在以上算法中，参数k表示我们要找到的族的数目，聚类质心μ_j表示我们对第j个族的中心的估计。我们可以随机选择k个训练样本，并以它们初始化聚类质心（step 1）。算法的内部循环进行了两步：(1)给每个样本分配一个族，分配的依据是到所有聚类质心距离最短的质心的编号；(2)更新所有的聚类质心，更新的依据是属于质心所在族的所有样本的平均值。重复内部循环直至聚类质心收敛。



上图很好的演示了算法。在图(b)中初始化两个聚类质心，在图(c)中给所有的样本分配族，在图(d)中更新质心的位置，在图(e)中重新分配族，在图(f)中继续更新质心位置，至此已收敛。

k-means算法一定能够保证收敛吗？答案是肯定的。我们定义失真函数（distortion function）为

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

显然J表示的是所有样本到它所属族的质心[2]的距离之和。我们可以观察到，k-means算法的过程其实就是坐标上升算法：首先固定μ并最小化关于c的函数J；接着固定c并最小化关于μ的函数J。重复这个过程，J会不断减小并最终收敛[3]。

事实上，失真函数J是一个非凸函数，因此坐标上升并不能保证J收敛到全局最优解。一般情况下，k-means算法的工作效果不错，我们也有办法避免结果收敛到局部最优解，比如使用不同的初始化聚类质心，最终在这些结果里选择失真函数$J(c,\mu)$最小的。我们也可以用这种方法来选择合理的k值。

2 混合高斯模型

在这里我们希望对未标记数据建立一个模型，样本和类标签的联合分布的概率$p\left(x^{(i)},z^{(i)}\right)=p\left(x^{(i)}|z^{(i)}\right)p\left(z^{(i)}\right)$。其中，$z^{(i)}\sim\text{Multinomial}(\phi)$[4]，且$x^{(i)}|z^{(i)}\sim\mathcal{N}(\mu_j,\Sigma_j)$。令$k$表示$z^{(i)}$所有可能取值的种类数，那么给定$z^{(i)}$，$x^{(i)}$便服从于$k$种依赖于$z^{(i)}$高斯分布之一。我们称这个模型为混合高斯模型（Mixtures of Guassian），称$z^{(i)}$为隐含随机变量。

在我们模型中存在3个参数ϕ,μ,Σ（多项式分布的参数和高斯分布的参数），为了估计它们，构造log-likelihood

$$
\begin{aligned}
l(\phi, \mu, \Sigma) &= \sum_{i=1}^{m} \log p\left(x^{(i)}; \phi, \mu, \Sigma\right) \\
&= \sum_{i=1}^{m} \log \sum_{z^{(i)}=1}^{k} p\left(x^{(i)} | z^{(i)}; \mu, \Sigma\right) p\left(z^{(i)}; \phi\right)
\end{aligned}
$$

令上式l对参数的偏导等于0，我们会发现不可能解出参数的最大似然估计。然而，如果我们可以固定$z^{(i)}$，解参数的最大似然估计就会容易得多，此时的log-likelihood是

$$
l(\phi, \mu, \Sigma) = \sum_{i=1}^{m} \log p\left(x^{(i)} | z^{(i)}; \phi, \mu, \Sigma\right) + \log p\left(z^{(i)}; \phi\right)
$$

得到最大似然估计

$$
\begin{aligned}
\phi_j &= \frac{1}{m} \sum_{i=1}^{m} 1\left\{z^{(i)} = j\right\} \\
\mu_j &= \frac{\sum_{i=1}^{m} 1\left\{z^{(i)} = j\right\} x^{(i)}}{\sum_{i=1}^{m} 1\left\{z^{(i)} = j\right\}} \\
\Sigma_j &= \frac{\sum_{i=1}^{m} 1\left\{z^{(i)} = j\right\} \left(x^{(i)} - \mu_j\right) \left(x^{(i)} - \mu_j\right)^T}{\sum_{i=1}^{m} 1\left\{z^{(i)} = j\right\}}
\end{aligned}
$$

可以观察到，固定$z^{(i)}$后我们几乎会得到一个高斯判别分析模型（Guassian Discriminant Analysis），不同的是高斯判别分析中，$z^{(i)} \sim \text{Bernoulli}(\phi)$，且我们一般令所有高斯分布$x|z$的形状相同，也就是说协方差矩阵$\Sigma$相同。

在此基础上，我们提出最大期望算法（Exception-Maximization, EM）。EM算法是一种迭代模型，包含两个主要步骤。具体到我们的算法中，即E-step——这一步尝试『估计』$z^{(i)}$的值，和M-step——这一步基于我们的估计更新模型的参数。由于在M-step中我们得到了一个确切值的估计而非随机变量，所以最大化的过程变得很容易。算法的步骤如下

- 重复以下过程直至收敛
    - (E-step) For each i,j, set

$$
w_j^{(i)} := p\left(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma\right)
$$

    - (M-step) Update the parameters

$$\phi_j = \frac{1}{m}\sum_{i=1}^{m} w_j^{(i)}$$

$$\mu_j = \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{m} w_j^{(i)} \left(x^{(i)} - \mu_j\right)\left(x^{(i)} - \mu_j\right)^T}{\sum_{i=1}^{m} w_j^{(i)}}$$

在E-step中，我们在给定$x^{(i)}$以及当前参数的设置值的情况下计算出$z^{(i)}$的后验概率，也就是估计$z^{(i)}$的值。应用贝叶斯公式，我们可得

$$p\left(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma\right) = \frac{p\left(x^{(i)} | z^{(i)} = j; \mu, \Sigma\right) p\left(z^{(i)} = j; \phi\right)}{\sum_{l=1}^{k} p\left(x^{(i)} | z^{(i)} = l; \mu, \Sigma\right) p\left(z^{(i)} = l; \phi\right)}$$

在M-step中，更新公式与当$z^{(i)}$取一确切值时参数的似然估计是一致的，只是为了计算上的方便我们将$1\left\{z^{(i)} = j\right\}$替换成$w_j^{(i)}$[5]。更新参数后发现$w_j^{(i)}$不对了，需要重新计算，于是周而复始直至收敛。

可以观察到混合高斯模型和k-means聚类算法很相似，只是在k-means中我们对样本分配标签是所谓『硬分配』，而在混合高斯模型中我们对是『软分配』[6]。与k-means一样，混合高斯模型也容易收敛到局部最优解，所以以不同的初始值多次训练模型是个好主意。

事实上，在混合高斯模型中用到的算法是EM的一个特例，在下一章中我们将介绍更一般的EM。我们可以在包含隐含变量的估计问题上很容易地应用它，而且可以保证结果收敛。

---

1. Written by Jimmy on 2016/03/10. ↩
2. 样本$x^{(i)}$所属族的编号为$c^{(i)}$，因此$\mu_{c^{(i)}}$即样本所属族的质心。 ↩
3. 一般而言，当J收敛时，c, μ也同时收敛。但是在理论上存在一种可能，那就是算法在几种不同的聚类结果上摇摆，即这些结果c, μ不同但是J取值相同。这在现实中几乎不可能出现。 ↩
4. 在多项式分布中，$\phi_j \geq 0, \sum_{j=1}^{k} \phi_j = 1$，其中$\phi_j = p\left(z^{(i)} = j\right)$。当$k = 2$时，我们得到了伯努利二项分布。 ↩
5. M-step的参数更新公式表明在得出它之前，我们是知道$x^{(i)}$所对应的$z^{(i)}$的。因为将最初的最大似然估计做一下等价变形就得到了我们的更新公式，因此我们在E-step求出公式里所需的$w_j^{(i)}$即可，但却隐含着我们在估计$z^{(i)}$的思想。 ↩
6. 『硬分配』是指k-means迭代的第一步会给所有样本分配确定的族，『软分配』是指混合高斯模型在E-step给样本分配族是一个概率形式的。 ↩