MLAPP（Machine Learning A Probabilistic Perspective）读书会第六次活动

第十一章 Mixture models and the EM algorithm

主讲人 SIAT

（新浪微博:@priceton）

QQ 群 177217565

读书会微信公众平台请扫描下面的二维码

SIAT(983755855) 19:59:19
本次读书会主要参考了 NG 的课件，主要内容包括：
k-Means 与 Em 算法；混合高斯模型与 EM 算法；EM 算法基本原理；EM 算法对缺失值的处理。
首先是 K-means 聚类与 EM 算法， 问题是这样：

给定样本集合 $\{x^{(1)}, \ldots, x^{(m)}\}$，我们希望把这么多数据聚成 K 类，其算法就是不断地纠正每个样本的所属的类别，并且不断修正聚类中心：

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence: {

   For every i, set
   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

   For each j, set
   $$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

   }

但是，问题是，我们最后怎么保证这个算法会收敛，就是怎么是的每次迭代更新，所得到的解与真实的解更接近？

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

在这里可以定义 误差函数：

用这个函数来衡量每一次迭代结果的好坏，然后进行判断。
gdanskamir<gdanskamir@gmail.com> 20:08:47
m 表示啥？
CodeMan(976209075) 20:09:06
样本个数吧，赶上直播了
SIAT(983755855) 20:09:07
m 样本个数
对这个误差函数，每次迭代中，固定每个类的质心，根据距离质心的远近来调整每个样本的类属性；另一方面，在本轮循环中， 又需要固定每个类的属性， 而计算类所属的质心。感觉有点像是先有鸡还是先有蛋的问题，从哲学上讲， 这类问题一般是从进化的角度来考虑。但在这里，我们需要用误差函数来衡量，即每次迭代都使得误差能够减少 。
因为 K-Means 方法比较浅显易懂，这部分过的比较快，下面谈下其与 EM 算法的关系。
疯狂的雪(675638960) 20:16:41
为什么这样子不断更新，每一轮可以是误差函数减小？每一轮一定会比上一轮误差小么？
Bavaria lumberjack(354616478) 20:17:07
坐标上升法，保证了一定收敛，NG 的课件或视频，关于收敛讲的很到位。
SIAT(983755855) 20:17:37
这个问题，在等会讲到 EM 算法原理时候会涉及到。因为误差函数是一个非凸函数，在迭代过程中也会陷入到局部最优，但可以随机的多试几次，选择最大即可。
关于 K-means 与 EM 算法，可以这样来看：
我们进行 K 均值聚类，其实就是希望将每个样本标号，找到每个样本所属的隐含类别 yi，在开始阶段，对

每个样本可以任意设定一个类别，当然，在 K 均值里面，我们是把相近的样本点设为一类，其过程可以这样分：
E 步： 暂时确定每个样本所属的类别，即隐含变量；
M 步：更新参数，优化误差函数 J
这部分大家有没有什么问题，没有的话，下面进入第二部分
♂cannon~~(514430052) 20:25:19
不是随机初始化类的中心点么？
SIAT(983755855) 20:25:35
是随机的，但有时候会陷入局部极值，通常多做几次随机就行了。
终点更是起点(331863609) 20:26:16
多次随机迭代看 J 最小的吧。
未来 Robot(887400) 20:26:47
通常会选择某些 xi 作为起始点，以免出现某个类为空的情况
c:\cph(1499321804) 20:27:40
E步： 暂时确定每个样本所属的类别，即隐含变量 那这个是给出了哪个隐含变量的期望呢，感觉不太好类比。
未来 Robot(887400) 20:32:26
那这个是给出了哪个隐含变量的期望呢，感觉不太好类比 --- 这里比较特殊，ng 的讲义上说是一个 hard 指定。是直接选出样本所属的一个类，而不是给定样本在不同分类的概率，相当于某个类概率为 1，其余为 0。
疯狂的雪(675638960) 20:29:20
是不是 M 步中，重新计算每个簇中心，取上一 E 步赋给该 cluster 的样本点的均值，能够是误差最小？然后在 E 步中，调整每个样本的类别属性，选择上一 M 步计算出来的簇中心离它最近的那个，这样就使得在 E 步和 M 步，误差都在降低？
SIAT(983755855) 20:30:08
是这样子。
相比于 K-Means，混合高斯模型只是把原来在 K-Means 中，用高斯模型取代聚类中心，在 K-Means 中，我们一直在做优化聚类中心的坐标。而对应于混合高斯模型，我们需要求高斯分布的参数。

We wish to model the data by specifying a joint distribution $p(x^{(i)}, z^{(i)}) = p(x^{(i)}|z^{(i)})p(z^{(i)})$. Here, $z^{(i)} \sim \text{Multinomial}(\phi)$ (where $\phi_j \geq 0$, $\sum_{j=1}^{k} \phi_j = 1$, and the parameter ϕ_j gives $p(z^{(i)} = j)$,), and $x^{(i)}|z^{(i)} = j \sim \mathcal{N}(\mu_j, \Sigma_j)$. We let k denote the number of values that the $z^{(i)}$'s can take on. Thus, our

这段话意思就是 数据集来自不同的 K 个服从 Gaussian 分布的模型，但我们只有这些数据，和对模型的假设，我们希望找到这多个 Gaussian 分布的参数。
在接下来过程中，我们用似然函数来来优化，所谓似然函数，就是指我得到了一些样本， 而且这些样本是来自比如说 Gaussian 分布，这些数据的出现，是在某种参数下最优可能出现的，比如样本服从均匀分布，但是不知道区间，达到 N 个样本，因此可以估计均匀分布的参数是 这 N 各样的最大值和最小值。所以，同样的道理，我们得到了这些数据，就可以假定，它们是在某种参数情况下最优可能出现的情况。通常，假设样本是独立同分布，对其取对数：

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

这个里面 Z 是隐含变量，表示每个数据的类属性，即来自那个 Gaussian 模型，我们要对这个式子进行优化，但由于隐含变量未知，所以直接求通常会比较麻烦。当然，如果告诉每个数据样本的类属性，我们可以按照单个高斯模型来求解：

$$\ell(\phi,\mu,\Sigma)=\sum_{i-1}^{m}\log p(x^{(i)}|z^{(i)};\mu,\Sigma)+\log p(z^{(i)};\phi).$$

优化这个函数，得到第 j 个分布的解：

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

这是假设知道每个数据所属的类别情况，就像在 K-Means 中，一直数据集中的某些点属于第 J 类，我们直接算其质心，都是可以类比的，只是这里参数稍稍复杂一些。
但现实是，我们并不知道每一个数据样本所属的类别，就像 K-Means 中，我们开始并不清楚每个样本属于哪个类。按照 K-Means 中的思路，开始可以随意假设，一般的思路，我们把每个样本赋予一个概率，表示该样本属于第 j 类的概率：$w_j^{(i)}:=p(z^{(i)}=j|x^{(i)};\phi,\mu,\Sigma)$

现在我们就可以做之前做过的事情，已知每个样本所属的类，计算类的模型参数：

$$\begin{aligned}\phi_j &:= \frac{1}{m}\sum_{i=1}^{m}w_j^{(i)},\\ \mu_j &:= \frac{\sum_{i=1}^{m}w_j^{(i)}x^{(i)}}{\sum_{i=1}^{m}w_j^{(i)}},\\ \Sigma_j &:= \frac{\sum_{i=1}^{m}w_j^{(i)}(x^{(i)}-\mu_j)(x^{(i)}-\mu_j)^T}{\sum_{i=1}^{m}w_j^{(i)}}\end{aligned}$$

在暂时已知模型时，又可以进一步返回迭代去更新每个样本所属的类属性，即我们又回到：

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

当然，实际中，这里的权值最后都需要归一化处理，那是小问题了，上面这个式子用 Bayes 展开：

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

和 K-Means 相比，他们背后都是这样一种思路，现在应该看得比较清楚，关于这个算法为什么会趋于最优（局部最优），下面第三部分会介绍，这部分大家看有没有什么问题？

==========讨论==========

♂cannon~~(514430052) 21:02:13

GMM 可以不用 EM 算法求解么？

SIAT(983755855) 21:03:06

你还有别的什么方法可以一起分享吗？

未来 Robot(887400) 21:05:21

zi 的分布，从哪里体现了多项式分布？

猛虎下山(12784305) 21:05:44

EM 方法和变分方法是什么关系？

HX(458728037) 21:06:12

顶 想知道 EM 变分 和普通 EM 的差别

一叶知秋(63160393) 21:06:22

同问

SIAT(983755855) 21:07:11

这里所指的多项式分布， Z 取不同值的概率，完全表示类的关系

秦淮/sun 人家(76961223) 21:07:31

EM 的 E 得到精确的后验，而 EM 变分只能近似后验

SIAT(983755855) 21:07:51

Z = 1 表示样本来自第一类 Gaussian 模型

未来 Robot(887400) 21:09:09

其实就是取 n 类的概率，多项式分布是指某个类重复取 k 次的分布吧？

SIAT(983755855) 21:09:56

就是取第 n 类的概率

HX(458728037) 21:10:00

为什么变分 EM 只是得到近似的,这位老师讲完后可以稍微普及一下什么变分 EM 哦

SIAT(983755855) 21:11:08

其实变分 EM 算法我不是太了解，等会内容结束后可以一起讨论

HX(458728037) 21:12:32

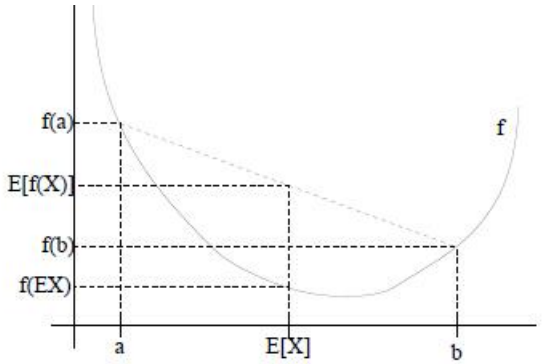
恩

==========讨论结束==========

SIAT(983755855) 21:12:33

下面进入第三部分：为什么 E M会逐渐变得最优（局部最优）。

假设大家知道 Jensen 不等式，关于凸函数有： $$\mathrm{E}[f(X)] \geq f(\mathrm{E}X).$$



之前我们希望求得参数，使得似然函数最大：

$$\begin{aligned}\ell(\theta) &= \sum_{i=1}^{m} \log p(x;\theta) \\ &= \sum_{i=1}^{m} \log \sum_{z} p(x,z;\theta).\end{aligned}$$

在第二部分，我们假设隐含变量服从多项式分布，这里可以进一步放宽，但需要是离散的，如果连续，就涉及到动态系统方面的知识，后续部分会有人讲到，这里只考虑离散情况 。

Q_i be some distribution over the z's ($\sum_z Q_i(z) = 1$,

上面的似然函数，可以这样处理：

$$\begin{aligned}\sum_i \log p(x^{(i)};\theta) &= \sum_i \log \sum_{z^{(i)}} p(x^{(i)},z^{(i)};\theta) \\ &= \sum_i \log \sum_{z^{(i)}} Q_i(z^{(i)}) \frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})} \\ &\geq \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}\end{aligned}$$

这里的不等式就用到了 刚才 Jensen 不等式，log 是一个凹函数

若要满足取等条件：就有 $\frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})} = c$

得到 $Q_i(z^{(i)}) \propto p(x^{(i)},z^{(i)};\theta).$

关于隐含变量的分布，最后得到：

$$\begin{aligned}Q_i(z^{(i)}) &= \frac{p(x^{(i)},z^{(i)};\theta)}{\sum_z p(x^{(i)},z;\theta)} \\ &= \frac{p(x^{(i)},z^{(i)};\theta)}{p(x^{(i)};\theta)} \\ &= p(z^{(i)}|x^{(i)};\theta)\end{aligned}$$

关于隐含变量的一个分布，我们用这个分布，进行下一次循环迭代，EM 算法，最后总结出来就是这样关键两步：

Repeat until convergence {

(E-step) For each i, set

$$Q_i(z^{(i)}) := p(z^{(i)}|x^{(i)};\theta).$$

(M-step) Set

$$\theta := \arg\max_{\theta} \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}.$$

这是 EM 算法的核心部分：我们看一下它怎么收敛的 ：

$$\ell(\theta^{(t)}) = \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)},z^{(i)};\theta^{(t)})}{Q_i^{(t)}(z^{(i)})}.$$

通过最大化右边的式子，得到：

$$\begin{aligned}\ell(\theta^{(t+1)}) &\geq \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t+1)})}{Q_i^{(t)}(z^{(i)})} \\ &\geq \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{Q_i^{(t)}(z^{(i)})} \\ &= \ell(\theta^{(t)})\end{aligned}$$

第一个式子是用到了 Jensen 不等式。

第二个是 我们在参数优化中，取 $\arg\max_\theta \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})},$

所以在每轮迭代后，下一次的似然 $\ell(\theta^{(t+1)})$ 会比上一次增加。

到这里，我们知道了迭代的方向性，就是下一次的似然函数值比之前的要大一些，但这还不能保证其收敛性，只有单调性，还需要一个有界的条件，但实际情况是这个有界的条件并不容易获得， 在算法里面可以把似然函数稳定下来的值作为其最大值（局部最大），第三部分就暂到这里，这里主要是说明其单调性。
疯狂的雪(675638960) 21:36:45
似然函数应该是有界的吧，最大也就是 1，毕竟似然函数也是个概率，概率也不会超过 1 啊。
SIAT(983755855) 21:37:14
实际问题通常都会有界，是这样的， 1 是上界。没有什么问题，就进入第四部分。
HX(458728037) 21:40:21

$$\begin{aligned}\ell(\theta^{(t+1)}) &\geq \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t+1)})}{Q_i^{(t)}(z^{(i)})} \\ &\geq \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{Q_i^{(t)}(z^{(i)})} \\ &= \ell(\theta^{(t)})\end{aligned}$$

通过最大化右边的式子，得到：

其中第二个式子是怎么来的，再解释一下?
SIAT(983755855) 21:40:43
第二个不等式吗
HX(458728037) 21:40:57
是的
SIAT(983755855) 21:42:32

$$\theta := \arg\max_\theta \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}.$$

我们在这步的时候，是选择使得后面式子最大的 theta
HX(458728037) 21:43:39
这是说明在第 t 次迭代的时候找到的最大 theta,那为什么第 t+1 次的一定会大于第 t 次的?
SIAT(983755855) 21:44:02

而最大的 θ 就是作为下一轮的 $\theta^{(t+1)}$，至少不会比它小，如果他们一直相等，就说明找到解了。

HX(458728037) 21:44:56

哦 这样 好的哈 多谢,你继续哈

SIAT(983755855) 21:45:40

接下来第四部分：对缺失数据的处理。

因为拿到的数据，某些属性丢失，所以需要先对数据建模处理

data matrix, due to missing data (usually represented by NaNs). More formally, let $O_{ij} = 1$ if component j of data case i is observed, and let $O_{ij} = 0$ otherwise. Let $\mathbf{X}_v = \{x_{ij} : O_{ij} = 1\}$ be the visible data, and $\mathbf{X}_h = \{x_{ij} : O_{ij} = 0\}$ be the missing or hidden data. Our goal is to compute

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\mathbf{X}_v|\boldsymbol{\theta}, \mathbf{O}) \tag{11.96}$$

数据包括完整 和不完整两部分，我们的目标是：求使得观测数据的似然最大的参数。数据包括观测到的部分和未观测到的部分。

假设样本独立同分布：

$$p(\mathbf{X}_v|\boldsymbol{\theta}, \mathbf{O}) = \prod_{i=1}^{N} p(\mathbf{x}_{iv}|\boldsymbol{\theta})$$

取对数得似然函数：

$$\log p(\mathbf{X}_v|\boldsymbol{\theta}) = \sum_i \log p(\mathbf{x}_{iv}|\boldsymbol{\theta})$$

where

$$p(\mathbf{x}_{iv}|\boldsymbol{\theta}) = \sum_{\mathbf{x}_{ih}} p(\mathbf{x}_{iv}, \mathbf{x}_{ih}|\boldsymbol{\theta})$$

v 表示观测到数据，h 表示确实数据

这个式子展开就得到：
$$\log p(\mathbf{X}_v|\boldsymbol{\theta}) = \sum_i \log \left[\sum_{\mathbf{x}_{ih}} p(\mathbf{x}_{iv}, \mathbf{x}_{ih}|\boldsymbol{\theta}) \right]$$

这种形式和之前的 EM 算法中介绍的非常相似，当然，这里还需要假设其他们是离散的，假设模型仍旧服从多元高斯分布，按照 EM 算法的思路：

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) &= \mathbb{E}\left[\sum_{i=1}^{N} \log \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma}) | \mathcal{D}, \boldsymbol{\theta}^{t-1} \right] \\ &= -\frac{N}{2} \log |2\pi\boldsymbol{\Sigma}| - \frac{1}{2} \sum_i \mathbb{E}\left[(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \right] \\ &= -\frac{N}{2} \log |2\pi\boldsymbol{\Sigma}| - \frac{1}{2} \mathrm{tr}(\boldsymbol{\Sigma}^{-1} \sum_i \mathbb{E}\left[(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \right] \\ &= -\frac{N}{2} \log |\boldsymbol{\Sigma}| - \frac{ND}{2} \log(2\pi) - \frac{1}{2} \mathrm{tr}(\boldsymbol{\Sigma}^{-1} \mathbb{E}\left[\mathbf{S}(\boldsymbol{\mu}) \right]) \end{aligned}$$

where

$$\mathbb{E}\left[\mathbf{S}(\boldsymbol{\mu}) \right] \triangleq \sum_i \left(\mathbb{E}\left[\mathbf{x}_i \mathbf{x}_i^T \right] + \boldsymbol{\mu}\boldsymbol{\mu}^T - 2\boldsymbol{\mu}\mathbb{E}\left[\mathbf{x}_i \right]^T \right)$$

这个式子看起来比较不舒服，对这个式子求导，也就是 $\nabla Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)}) = \mathbf{0}$,

也可以得到对参数估计：

$$\begin{aligned} \boldsymbol{\mu}^t &= \frac{1}{N} \sum_i \mathbb{E}\left[\mathbf{x}_i \right] \\ \boldsymbol{\Sigma}^t &= \frac{1}{N} \sum_i \mathbb{E}\left[\mathbf{x}_i \mathbf{x}_i^T \right] - \boldsymbol{\mu}^t (\boldsymbol{\mu}^t)^T \end{aligned}$$

详细的内容可以参考 Mlapp 这本书部分。这个地方推导比较繁琐，大家下来可以试试，好吧，今天就到这里吧。下来可以再交流，今天非常感谢大家参与。如果关于 EM 算法有不太清楚的地方，或者有新的见解，

随时欢迎交流：个人邮箱: priceton@163.com

==========讨论==========

疯狂的雪(675638960) 22:04:26

是不是对于每个 data case 要计算它的缺失属性取各种值的概率？

SIAT(983755855) 22:06:33

是要估计，不过书上给出的公式:

$$\begin{aligned}\mathbf{x}_{ih}|\mathbf{x}_{iv},\boldsymbol{\theta} &\sim \mathcal{N}(\mathbf{m}_i,\mathbf{V}_i)\\ \mathbf{m}_i &\triangleq \boldsymbol{\mu}_h+\boldsymbol{\Sigma}_{hv}\boldsymbol{\Sigma}_{vv}^{-1}(\mathbf{x}_{iv}-\boldsymbol{\mu}_v)\\ \mathbf{V}_i &\triangleq \boldsymbol{\Sigma}_{hh}-\boldsymbol{\Sigma}_{hv}\boldsymbol{\Sigma}_{vv}^{-1}\boldsymbol{\Sigma}_{vh}\end{aligned}$$

未来 Robot(887400) 22:24:32

$$\ell(\theta^{(t)})=\sum_i\sum_{z^{(i)}}Q_i^{(t)}(z^{(i)})\log\frac{p(x^{(i)},z^{(i)};\theta^{(t)})}{Q_i^{(t)}(z^{(i)})}.$$

我觉得这个是神来之笔，发明这个算法的人，是怎么想到凑出一个 E(fx)来的。应该是对 log 函数特性，Jensen 不等式都很熟悉的人，才会想着往这上面去凑，通常我们求最大化，只会往上限的不等式方面去想，用下限不等式去逼近，第一次见这个方法时，觉得脑洞大开，感觉很不可思议。

疯狂的雪(675638960) 22:30:40

要最大化一个东西，应该就最大化它的下限吧，想最小化一个东西，就找它的上限

未来 Robot(887400) 22:32:18

求一个函数的最大值，显然用 f(x) <= g(x) 更好求

疯狂的雪(675638960) 22:34:07

这样求出来的是 f(x)所达不到的值

未来 Robot(887400) 22:35:36

svm 里面，求解 p*, d*，kkt 条件就是用的这个

疯狂的雪(675638960) 22:36:06

但是如果想最大化 f(x)，找一个它的下限函数 g(x),g(x)取最大值出的 x 若为 x',f(x') >= g(x') >= g(x)

未来 Robot(887400) 22:36:23

在某个值 x*，使 f(x*)取最大值， g(x*)取最小值， f(x*) = g(x*)

疯狂的雪(675638960) 22:37:37

也就是可以保证我们想要最大的东西，至少可以比 g(x')大

未来 Robot(887400) 22:38:38

这种方法不能保证 f(x)全局最大，只是一个可能的下界的最大值，极端一点的话，假如 g(x)为一个常数，是 f(x)的最小值，那么 f(x) >= g(x)成立，这种情况如何求解 f(x)最大值？

疯狂的雪(675638960) 22:41:02

如果 f(x)不是凹或凸的，一般很难找到全局最大或者最小。这样的话，说明找的这个下限函数不恰当。

未来 Robot(887400) 22:53:54

这里的下限逼近法，g(x)不是一个固定函数，而是一个带参数的系列函数。在任何一个 x，都可以重新构造 g(x)，使得 f(x)>=g(x)，而且这里有一个最大的问题，是当 fx=gx 取得等号时， g(x)不能为最大值。但是在大部分的不等式取等号情况下，此时的 fx=gx，为 fx 最小值，gx 的最大值。我觉得找到这么一个下限逼近函数，首先不是很直觉，当然现在我知道有这个方法了，其次 gx 函数不好构造。设想自己写这么一篇论文，怎么才能搞出这么一个方法？

疯狂的雪(675638960) 23:03:22

嗯，同意。EM 中下限是模型参数和隐含变量的函数。每次求得隐含变量的取值概率之后，都可以构造一个下限，使得似然和下限在该点相等，但是下限函数的最大值不在该点，最大化下限就求得了新的模型参数。 上面说的貌似有点不妥，“EM 中下限是模型参数和隐含变量的函数”，传统 EM 中隐含变量取各个值的概率还是模型参数的函数，所以“下限也是模型参数的函数”。
在变分 EM 中，引入变分参数，用隐含变量的变分分布近似隐含变量的后验分布。下限就是变分参数和模型参数的函数。E 步关于变分参数最大化，M 步关于模型参数最大化。不知道我的这点理解对不对，如果有不对的地方，欢迎纠正沟通。
猛虎下山(12784305) 23:41:55
请问有变分 EM 这方面的中文资料吗，文章或者书
未来 Robot(887400) 23:50:59
你们看书时会把每一个公式都推一遍吗？
c:\cph(1499321804) 23:51:35
我感觉推一遍很重要。。。推一遍胜过看 10 遍，个人理解，看任何公式，不推很快就会忘记。