变分算法

原创 2016年11月09日 08:56:27 标签：机器学习 / 算法 / 贝叶斯 1620

注：本文中所有公式和思路来自于邹博先生的《机器学习**升级版**》，我只是为了加深记忆和理解写的本文。

本文介绍的变分算法是机器学习中的参数估计算法，跟数学中的变分法是有一些不一样的。在搜索引擎中搜索变分算法一般都是数学中的变分法，机器学习的好像还真的很少。

变分算法我觉得是机器学习算法中比较难的一个了，因为推导有很多，并且有一些想法是不太直观上被轻易接受或者说理解的，今天斗胆说一说。

我们不妨先看看变分的核心公式：

$$\log q_j(x_j) = E_{-q_j}[\log \tilde{p}(x)] + const$$

看起来怪怪的，又是log又是const又是非q的

我们可以先粗略的解释一下这个公式：当我们更新qj的时候，只需要计算和qj有公共边的那些变量即可，也就是j的马尔科夫毯包含的那些点。

看到非q这个东西是不是感觉像Gibbs采样啊，没错，确实是有些相同地方的，那么区别在哪呢：

Gibbs时采用的邻居节点的（相同文档中的词）的主题采样，而变分使用的是相邻结点的期望，很明显这样的话变分的速度是要比采样更快，因为求得是期望嘛，代替了很多次的采样。

我们以前可以用采样的方式改造EM算法，其实就是在E-Step计算条件概率上做点事情：

$$Q(\theta,\bar{\theta}) = \int p(Z \mid X,\bar{\theta}) \ln p(Z,X \mid \theta) dZ$$

$$Q(\theta,\bar{\theta}) \approx \frac{1}{L}\sum_{i=1}^{L} \ln p(Z^{(i)},X \mid \theta)$$

从上边公式可以看得出，假定p(x)是个不好计算的分布，那么我们可以通过采样的方式来近似，这种算法又叫做MC-EM，展开来还有随机EM等，就不细说了

那么现在我们是不是可以用变分的当时改造EM算法呢？

其实是可以的，我们后边介绍如何改造这里先卖一个关子。

他的最新文章
压力测试工具siege的用法
【卷积神经网络-进化史】从LeNet
linux 环境变量设置（临时 + 永久
gtest install
linux(ubuntu)下的caffe编译安装

文章分类
机器学习
linux
deep learning

文章存档
2017年1月
2016年12月
2016年11月

他的热门文章
linux(ubuntu)下的caffe编译安装
1748
EM算法---基于隐变量的参数估计
1674
概率图模型之贝叶斯网络
1657

说变分之前，我先说一个概念**近似估计：假如现在有一个分布是不容易计算的p(x|D)，那么可以选择一个简单的分布q(x)来近似分布p(x|D)。**

这种近似通常都会在精度和速度上折中，那么怎么度量这个近似呢？

变分提出：假定p*(x)是一个(真实)难解的分布，q(x)是一个近似的(容易)的分布-例如多元的高斯分布或者多个简单分布的乘积

如果这个q(x)有n个参数来控制，那么现在的问题就变成了：优化这些参数从而近似(逼近)p*(x)

那么怎么度量p*(x)和q(x)呢，显然我们可以使用KL散度啊：

$$KL(p^* \| q) = \sum_x p^*(x) log \frac{p^*(x)}{q(x)} = E_{p^*(x)}\left(log \frac{p^*(x)}{q(x)} \right)$$

现在我们来看看这个目标函数，既然我们前面假设p*(x)是一个难解的分布，我们现在目标函数Σ右边竟然还有p*(x)，那目标函数岂不是依然难解，自己打脸呀。

如果用逆KL散度呢？

$$KL(q \| p^*) = \sum_x q(x) log \frac{q(x)}{p^*(x)} = E_{q(x)}\left(log \frac{q(x)}{p^*(x)} \right)$$

这样就转换为计算q(x)的期望了，是不是就简单了很多了。

进一步我们再来分析这个难搞的p*(x)：

$$p^*(x) = p(x|D) = \frac{p(x,D)}{p(D)} \triangleq \frac{\tilde{p}(x)}{Z} \Rightarrow \tilde{p}(x) = Z \cdot p^*(x)$$

其中p(D)就是数据发生的似然概率，是一个已知可求的量，可以作为归一化因子，那么接下来整理一下：

$$J(q) = KL(q \| \tilde{p}) = \sum_x q(x) log \frac{q(x)}{\tilde{p}(x)}$$

这样我们就得到了新的目标函数，其实这样做还有一个好处，就是可以得到局部最小值。为啥呢？

两种KL散度的区别

KL(q||p)，又称为I-投影，信息投影(information projection)

$$KL(q \| p) = \sum_x q(x) log \frac{q(x)}{p(x)} = E_{q(x)}\left(log \frac{q(x)}{p(x)} \right)$$

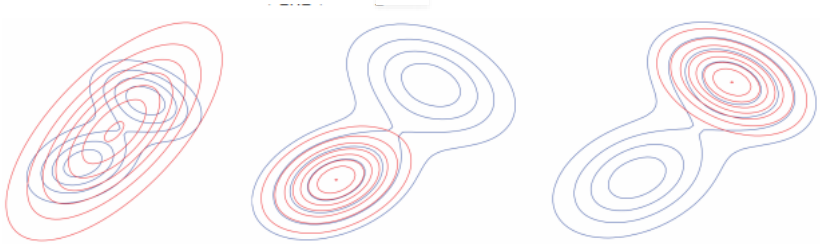
现在仔细看这个公式，p(x)是真实的分布，那么如果p(x)=0是不是KL散度就会无穷大？所以当p(x)=0时候q(x)也必须要为0，也

KL(p||q)，又称为M-投影，矩投影(moment projection)

$$KL(p\|q)=\sum_x p(x)\log\frac{p(x)}{q(x)}=E_{p(x)}\left(\log\frac{p(x)}{q(x)}\right)$$

再仔细看看公式，p(x)是真实的分布，如果p(x)≠0那么q(x)=0的话，KL散度就会无穷大，所以如果p(x)≠0，那么也只能q(x)≠0，也就是该公式对待q(x)是0避免的，所以p(x)往往是被高估的。

下图种蓝色的线是真实的分布，红色的是两种变分的近似效果，其中左边的是KL(p||q)，也就是M投影，可以看得出其实没办法锁住一个峰(也就是没办法求极值)，中间和右边的是KL(q||p)的I投影，因为初始值不同所以锁住的峰是不同的，但是都是可以求出一个极值的，所以我们使用I投影来做目标函数。



其实这个事情还可以说的更清楚，只不过书面上还真是不太容易说的太清楚，也没办法自己边画图边讲，所以尽量说的明白一点。

其实这两种KL散度也是有联系的，q(x)和p(x)也并不是很割裂的，是有关联公式：

首先给出q(x)和p(x)的距离定义：

$$D_\alpha(p\|q)=\frac{2}{1-\alpha^2}\left(1-\int p(x)^{\frac{1+\alpha}{2}}q(x)^{\frac{1-\alpha}{2}}dx\right)$$

再给出KL散度：

$$KL(p\|q)=\int p(x)\log\frac{p(x)}{q(x)}dx=-\int p(x)\log\frac{q(x)}{p(x)}dx$$

变换一下其实可以得出以下结论的：

当α=1的时候退化为KL(p||q)

当α=-1的时候退化为KL(q||p)

当α=0的时候就是Hellinger distance：

$$D_\alpha(p\|q)=\frac{2}{1-\alpha^2}\left(1-\int p(x)^{\frac{1+\alpha}{2}}q(x)^{\frac{1-\alpha}{2}}dx\right)$$
$$\Rightarrow D_H(p\|q)=2\left(1-\int\sqrt{p(x)q(x)}dx\right)=2-2\int\sqrt{p(x)q(x)}dx$$
$$=\int p(x)dx+\int q(x)dx-\int 2\sqrt{p(x)q(x)}dx$$
$$=\int\left(p(x)-2\sqrt{p(x)q(x)}+q(x)\right)dx$$
$$=\int\left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2dx$$

上面嘚吧嘚吧一大坨其实就是说了两种KL散度的区别，那么新的目标函数就是使用I投影的，我们可以做一下简单的变换：

$$J(q)=KL(q\|\tilde{p})=\sum_x q(x)\log\frac{q(x)}{\tilde{p}(x)}$$
$$=\sum_x q(x)\log\frac{q(x)}{Z\cdot p^*(x)}$$
$$=\sum_x q(x)\log\frac{q(x)}{p^*(x)}+\sum_x q(x)\log\frac{1}{Z}$$
$$=\sum_x q(x)\log\frac{q(x)}{p^*(x)}-\log Z$$
$$=KL(q\|p^*(x))-\log Z$$

这个结果就是简单地变换而来，没有任何其他东西加进来，很简单，其中Z是一个上边那个常数p(D)，-logZ=-logP(D)，是负对数似然。我们就要最小化J(q)，能够使q近似p*

又因为KL为非负的，所以J(q)是NLL(负对数似然)的上界：

$$J(q)=KL(q\|p^*)-\log Z\geq-\log Z=-\log p(D)$$

那么进一步变换：

$$L(q)\overset{\Delta}{=}-J(q)=-KL(q\|p^*)+\log Z\leq\log Z=\log p(D)$$

只不过是加了一个负号，变成似然函数的下界，仅当q=p*的时候，可以取等号。

不知道看到这里你是不是会有似曾相识的感觉，EM算法不就用另外一个函数来不断优化逼近原分布的极大值的么！！！

EM算法：计算关于隐变量后验概率的期望得到下界；

变分：计算KL散度，得到下界； $\min_{q_1,\cdots q_D} KL(q\|p)$

相同的思维：不断的迭代，得到更好的下界；

坐标上：不断地上升

目标函数已经出来了，暂时放在这里，接下来要介绍一个很重要的方法：平均场方法。

平均场方法

最流行的的变分方法之一就是平均场近似，在这种方法中，假设后验概率能够近似分解为若干因子的乘积：

$$q(x)=\prod_i q_i(x_i)$$

我们的目标是最优化问题：

$$\min_{q_1,\cdots q_D} KL(q \parallel p)$$

平均场方法使得可以在若干边界分布qi上进行以此优化，说白了就是逐个击破，所以有下边的等式：

$$\log q_j(x_j) = E_{-q_j}[\log \tilde{p}(x)] + const$$

从公式中不难看出，要想求qj的对数似然，那么就可以通过求非q的期望加上一个const来近似。

其中，为正则的后验概率：$\tilde{p}(x) = p(x, L$ 。

这就是平均场方法，接下来就到了本文的重头戏，变分推导！！！！

变分推导

$$L(q_j) \overset{\Delta}{==} -J(q_j) = -\sum_x q(x) \log \frac{q(x)}{\tilde{p}(x)}$$
$$= \sum_x q(x)[\log \tilde{p}(x) - \log q(x)]$$
$$= \sum_x \prod_i q_i(x_i) \left[\log \tilde{p}(x) - \log \prod_i q_i(x_i)\right]$$

$$= \sum_{x_j} \sum_{x_{-j}} q_j(x_j) \prod_{i \neq j} q_i(x_i) \left[\log \tilde{p}(x) - \sum_k \log q_k(x_k)\right]$$
$$= \sum_{x_j} q_j(x_j) \sum_{x_{-j}} \prod_{i \neq j} q_i(x_i) \left[\log \tilde{p}(x) - \left(\log q_j(x_j) + \sum_{k \neq j} \log q_k(x_k)\right)\right]$$
$$= \left(\sum_{x_j} q_j(x_j) \sum_{x_{-j}} \prod_{i \neq j} q_i(x_i) \log \tilde{p}(x)\right) - \left(\sum_{x_j} q_j(x_j) \log q_j(x_j)\right) + const$$
$$= \left(\sum_{x_j} q_j(x_j) \log f_j(x_j)\right) - \left(\sum_{x_j} q_j(x_j) \log q_j(x_j)\right) + const = -KL(q_j \parallel f_j)$$

其实把这段公式放在这里我是觉得挺为难的，因为不容易跟大家一一说明每一步是怎么变换过来的，其实不难，就是不太容易想到，邹博先生给我们讲这块的时候我也是听了两遍并且自己手动写了一遍才理解，所以大家最好也消化一下。

我只说一下最后一步：我们得到最后一部这个公式后，然后令：

$$\log f_j(x_j) \overset{\Delta}{==} \sum_{x_{-j}} \prod_{i \neq j} q_i(x_i) \log \tilde{p}(x) = E_{-q_j}[\log \tilde{p}(x)]$$

然后我们可以从第一部分和第二部分中提取公因式q(xi)，其余项在log的作用下变成f(x)/q(x)，我们加一个负号，就得到最终的结果。

很显然是最终结果-KL(q||f)，那么我们既然是想求J(q)的极小值，那么KL(q||f)中的q(x)就应该等于f(x)，最后可以得出这么一个结论：

$$q_j(x_j) = f_j(x_j) = \frac{1}{Z_j} \exp\left(E_{-q_j}[\log \tilde{p}(x)]\right)$$

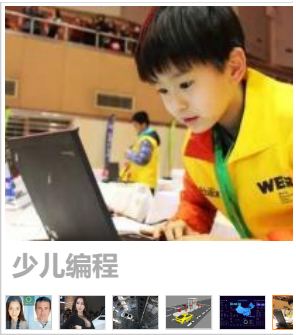
我们忽略掉归一化因子：

$$\log q_j(x_j) = E_{-q_j}[\log \tilde{p}(x)] + const$$

这就是我们的最终结论，就是文章开头给出的那个变分核心公式。

1

到此，变分算法的理论推理介绍完了，我将会在未来的一篇文章中介绍变分法的应用。



变分推理

LIQINGBHU 2016年03月08日 22:23 355

variational inference 泛函极值问题又称变分问题,寻求泛函极值的方法称为变分法「60]。图模型 变分推理是基于数学变分法的近似推理方法,其基本思想是根据凸对偶原理(...

变分推断(variational inference)笔记(1)——概念介绍

ref：http://www.crescentmoon.info/?p=709#more-709 问题描述 变分推断是一类用于贝叶斯估计和机器学习领域中近似计算复杂（intractable...

AiTODD1 2014年11月13日 21:04 7047

变分推断(Variational Inference)-mean field

变分推断的实质就是使用已知简单分布来逼近需要推断的复杂分布，并通过限制近似分布的类型，从而得到一种局部最优，但具有确定解的近似后验分布。...

step_forward_ML 2017年09月24日 16:00 834

变分贝叶斯推断(Variational Bayes Inference)简介

通常在研究贝叶斯模型中，很多情况下我们关注的是如何求解后验概率(Posterior)，不幸的是，在实际模型中我们很难通过简单的贝叶斯理论求得后验概率的公式解，但是这并不影响我们对贝叶斯模型的爱——既然...

aws3217150 2017年02月25日 16:42 6778

PRML读书会第十章 Approximate Inference（近似推断，变分推断，KL散度，平均...

第十章的主要内容是变分推断（Variational Inference），由中科院自动化所戴玮博士（前后分三次讲完。精彩内容有：为什么需要近似推断、变分推断用到的KL散度、根据平均场（Mean Fie...

Nietzsche2015 2015年02月03日 15:51 9201

机器学习中的优化算法、加速训练机制、损失函数、KL散度和交叉熵

1.优化算法为了说明梯度下降法、随机梯度下降法、批量梯度下降法三者区别，我们通过一组数据来拟合 y=θ1∗x1+θ2∗x2 y = \theta_1*x_1 +\theta_2*x_2梯度下降(gr...

Mr_KkTian 2016年11月17日 16:59 869

相对熵（KL距离）的java实现

u012878609 2015年10月26日 10:53 277

相对熵（relative entropy或 Kullback-Leibler divergence,KL距离）的java实现（二） 实验中，我们采用两种方法计算概率。一：以字符为...

非计算机专业研究生自学进BAT的经历！（转）

restaurant123 2015年06月02日 16:20 4001

昨天阿里的师兄带我到部门看了一下，团队的氛围很和谐，主管人也很好，看到我来了也主动跟我打招呼。也跟着团队们听了一个技术讲座，还有可乐零食吃。整个过程就感觉像一个班级再开个会，可以发言，可以开玩笑，很有...