参数估计

七月算法 邹博

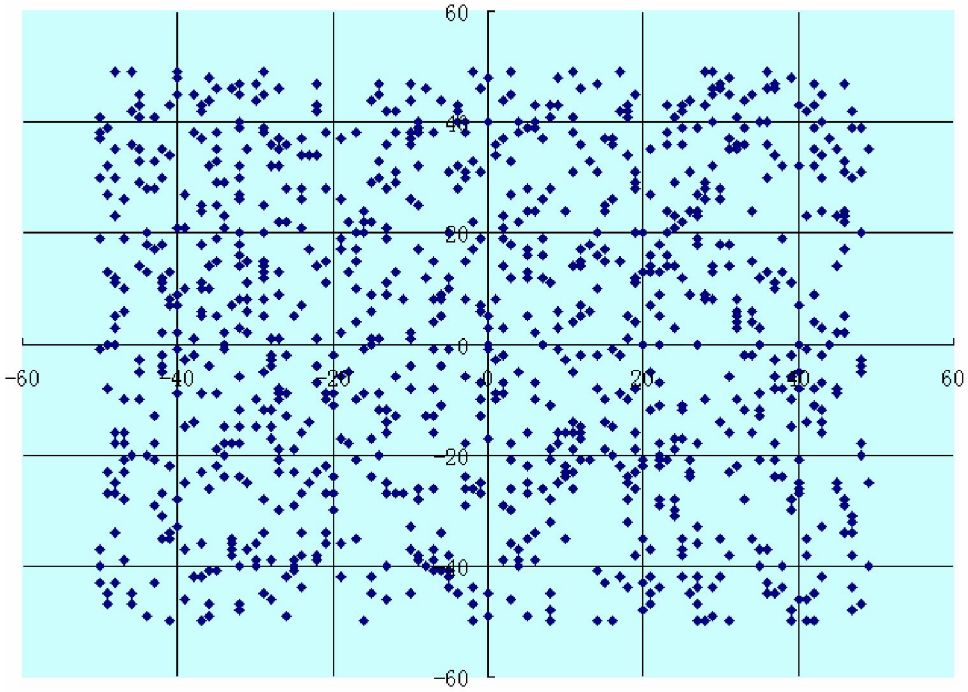
2015年12月6日

给定区域的二维随机数

- 最简单的采样问题：
  - 给定区间$[a_x,b_x]\times[a_y,b_y]$，使得二维随机点(x,y)落在等概率落在区间的某个点上。
- 分析：因为两个维度是独立的，分别生成两个随机数即可。

产生二维随机数代码与效果

```
int rand50()
{
    return rand() % 100 - 50;
}

int _tmain(int argc, _TCHAR* argv[])
{
    ofstream oFile;
    oFile.open(_T("D:\\rand.txt"));
    int x,y;
    for(int i = 0; i < 1000; i++)
    {
        x = rand50();
        y = rand50();
        oFile << x << '\t' << y << '\n';
    }
    oFile.close();
    return 0;
}
```

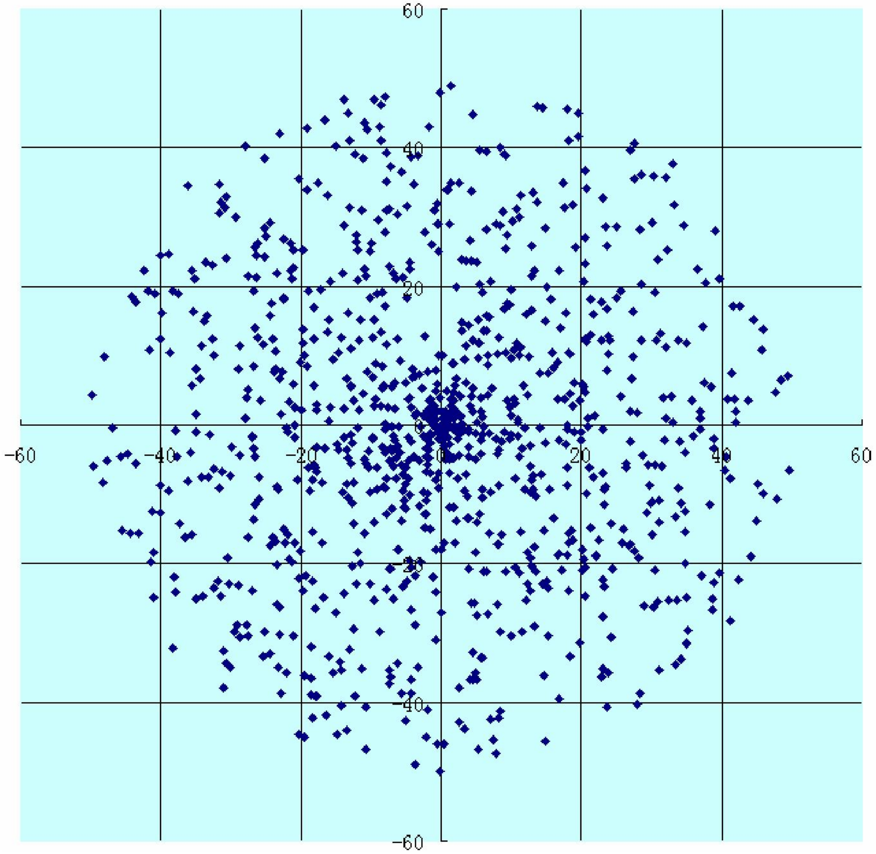


圆内均匀取点

- 给定定点$O(x_0,y_0)$和半径r，使得二维随机点(x,y)等概率落在圆内。
- 分析
  - 直接使用$x=x_0+r*\cos\theta$，$y=y_0+r*\sin\theta$是否可以呢？
  - 具体试验一下。

圆内均匀取点代码与效果

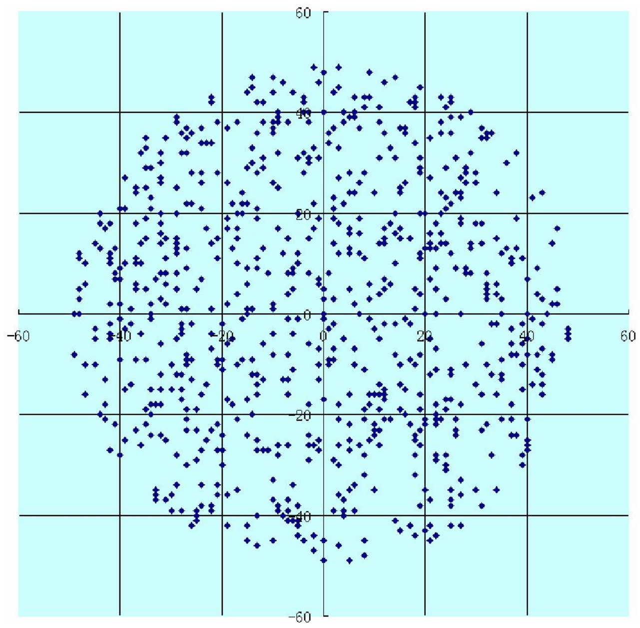
```
int rand50()
{
    return rand() % 100 - 50;
}

int _tmain(int argc, _TCHAR* argv[])
{
    ofstream oFile;
    oFile.open(_T("D:\\rand.txt"));
    double r,theta;
    double x,y;
    for(int i = 0; i < 1000; i++)
    {
        r = rand50();
        theta = rand();
        x = r*cos(theta);
        y = r*sin(theta);
        oFile << x << '\t' << y << '\n';
    }
    oFile.close();
    return 0;
}
```



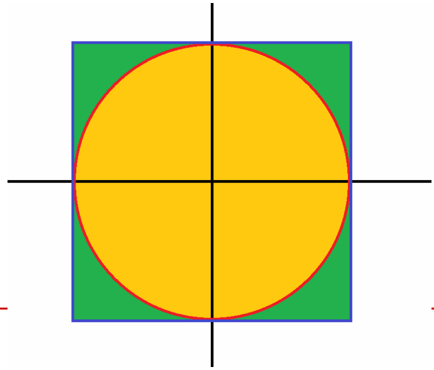
有选择的取点

- 显然上述做法是不对的。但可以使用二维随机点的做法，若落在圆外，则重新生成点。结果如下。

```
int rand50()
{
    return rand() % 100 - 50;
}

int _tmain(int argc, _TCHAR* argv[])
{
    ofstream oFile;
    oFile.open(_T("D:\\rand.txt"));
    int x,y;
    for(int i = 0; i < 1000; i++)
    {
        x = rand50();
        y = rand50();
        if(x*x + y*y < 2500)
            oFile << x << '\t' << y << '\n';
    }
    oFile.close();
    return 0;
}
```



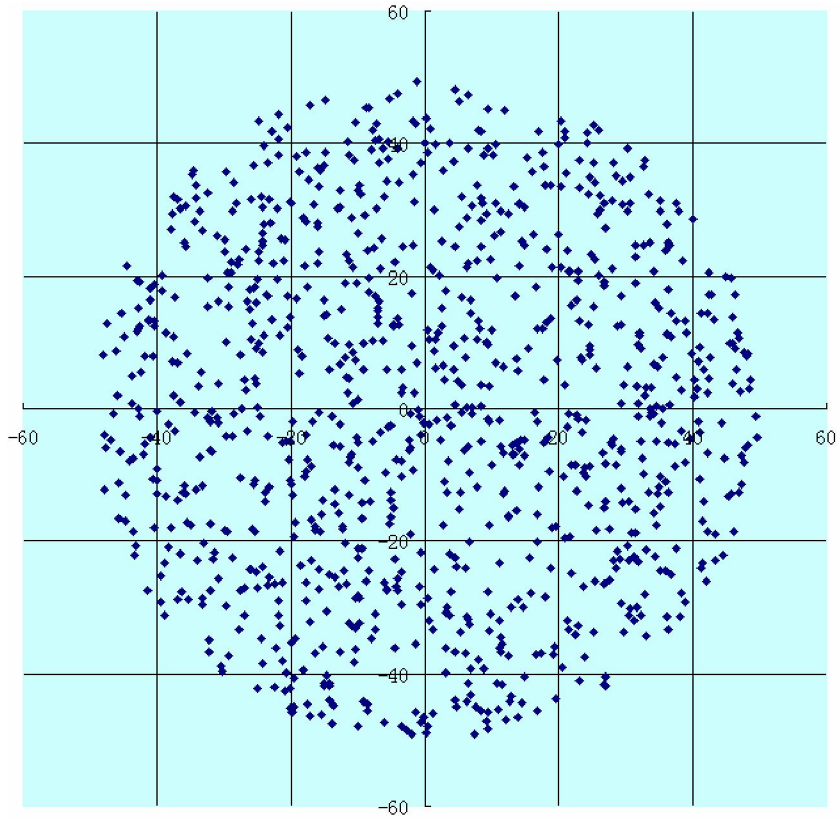
带拒绝的采样分析



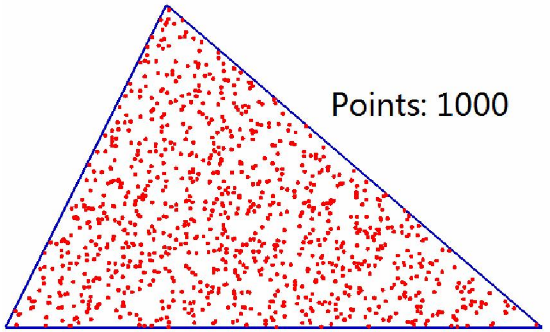
- 在对某区域$f(x,y)\leqslant 0$抽样的过程中，若该区域$f(x,y)\leqslant 0$不容易直接求解，则寻找某容易采样的区域$g(x,y)\leqslant 0$，G为F的上界。当采样$(x_0,y_0)\in G$且落在F内部时，接收该采样；否则拒绝之。
  - 该例中，$f(x,y)\leqslant 0$是圆，$g(x,y)\leqslant 0$是该圆的外包围正方形。
  - 注：区域$f(x,y)\leqslant 0$的可行解集合记做F；区域$g(x,y)\leqslant 0$的可行解集合记做G；显然$F\subseteq G$

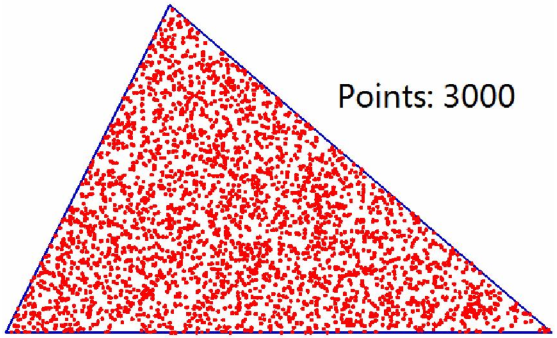
附：产生圆内随机数的其他方法

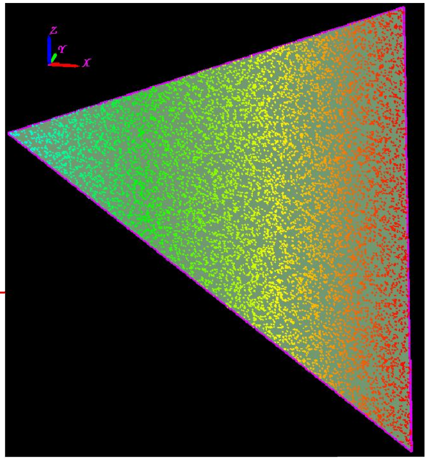
```
double rand2500()
{
    return rand() % 2500;
}

int _tmain(int argc, _TCHAR* argv[])
{
    ofstream oFile;
    oFile.open(_T("D:\\rand.txt"));
    double r,theta;
    double x,y;
    for(int i = 0; i < 1000; i++)
    {
        r = sqrt(rand2500());
        theta = rand();
        x = r*cos(theta);
        y = r*sin(theta);
        oFile << x << '\t' << y << '\n';
    }
    oFile.close();
    return 0;
}
```



附：产生三角形内随机数



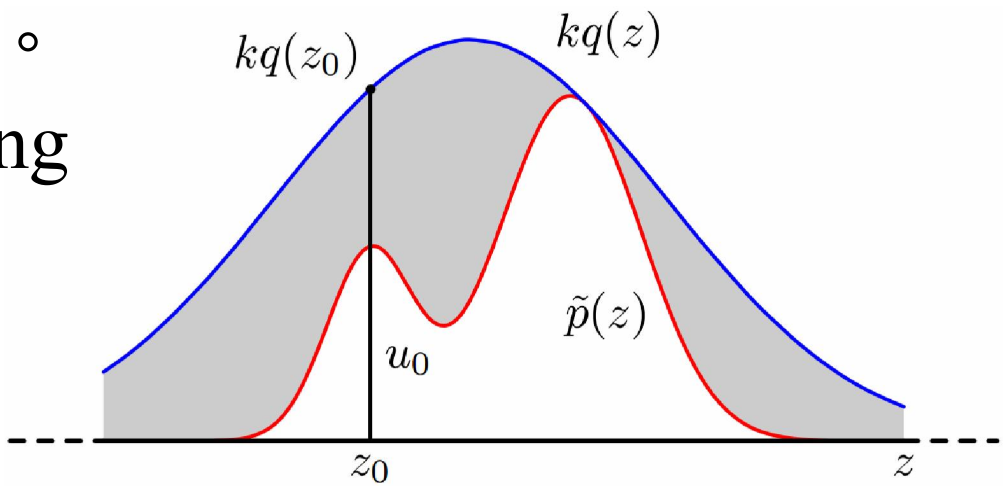




```
void CRandomTriangle::Random2(int nSize)
{
    CalcRotate();
    m_nSize = nSize;
    if(m_pRandomPoint)
        delete[] m_pRandomPoint;
    m_pRandomPoint = new CDelPoint[nSize];
    CDelPoint pt;
    for(int i = 0; i < nSize; i++)
    {
        pt.RandomInRectangle(m_ptExtend, m_ptHeight);
        if(m_tsBig.IsIn(pt))
        {
            pt += m_ptBase;
            m_pRandomPoint[i] = pt;
        }
        else if(m_tsLeft.IsIn(pt))
        {
            CDelPoint::MirrorPoint(pt, m_ptLeft0);
            pt += m_ptBase;
            m_pRandomPoint[i] = pt;
        }
        else if(m_tsRight.IsIn(pt))
        {
            CDelPoint::MirrorPoint(pt, m_ptRight0);
            pt += m_ptBase;
            m_pRandomPoint[i] = pt;
        }
    }
    CDelPoint::Save(m_pRandomPoint, m_nSize, _T("D:\\random.pt"), 0);
}
```

进一步思考：Rejection sampling

- 上述方法能够一定程度的估算圆周率——虽然精度很差。
- 上述抽样问题能否用来解决一般概率分布函数的抽样问题？如：根据均匀分布函数得到正态分布的抽样。
- Rejection sampling



对某概率分布函数进行采样的意义

- 根据抽样结果估算该分布函数的参数，从而完成参数的学习。
  - 前提：系统已经存在，但参数未知；
  - 方法：通过采样的方式，获得一定数量的样本，从而学习该系统的参数。
  - 例：投硬币试验中，进行N次试验，n次朝上，N-n次朝下——可以认为，是进行了N次(独立)抽样。
  - 假定朝上的概率为p，使用对数似然函数作为目标函数：

$$f(n\mid p)=\log\left(p^n(1-p)^{N-n}\right)\xrightarrow{\Delta}h(p)=\log\left(p^n(1-p)^{N-n}\right)$$

$$\frac{\partial h(p)}{\partial p}=\frac{n}{p}-\frac{N-n}{1-p}\xrightarrow{\Delta}0\Rightarrow p=\frac{n}{N}$$

应用Bernoulli版本的大数定理

- 一般的说，上述结论可以直接推广：频率的极限为概率：$p = {}^{n}\!/_{N}$
- 将上述二项分布扩展成多项分布，如K项分布：$p_i = \dfrac{n_i}{N}$
  - 从而得到K项分布的参数：$p = \left(\dfrac{n_1}{N}, \dfrac{n_2}{N} \cdots \dfrac{n_k}{N} \right)$
- 在主体模型LDA中，每个文档的主题分布和每个主题的词分布都是多项分布，如果能够通过采样的方式获得它们的一定数量的样本，即可估算主题分布和词分布的参数，从而完成参数学习！
  - 贝叶斯网络的另一种重要参数学习手段是EM算法，参见GMM、pLSA、HMM的推导过程。

附：Bernoulli版本的大数定理

- 一次试验中事件A发生的概率为p；重复n次独立试验中，事件A发生了n_A次，则p、n、n_A的关系满足：

  对于任意整数ε，

$$\lim_{n\to\infty} P\left\{\left|\frac{n_A}{n}-p\right|<\varepsilon\right\}=1$$

用采样改造EM算法本身

- 在EM算法中，E-Step求出隐变量的条件概率，从而给出期望Q，M-Step将目标函数Q求极大值，期望Q为：

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

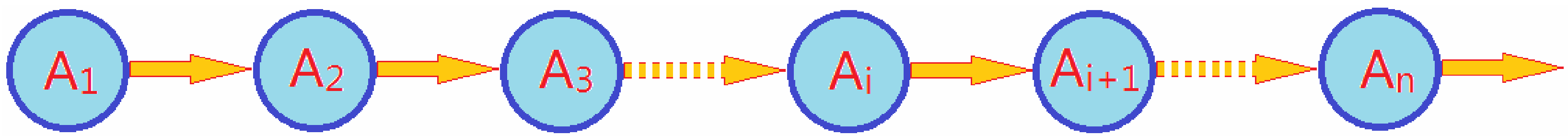
- 显然，这仍然可以使用采样的方式近似得到：

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

- 这种方式的EM算法被称为MC-EM算法(Monte Carlo EM)
  - MC-EM算法仅改变了E的计算，M的求极值本身没有变化。
- 极限情况：若MC-EM算法的期望Q的估计，仅采样一个样本，则称之为随机EM算法(stochastic EM algorithm)。
  - 此外，EM算法的M-Step，可以使用MAP而非MLE的方式，从而目标函数最后多一项lnp(θ)。

重述采样

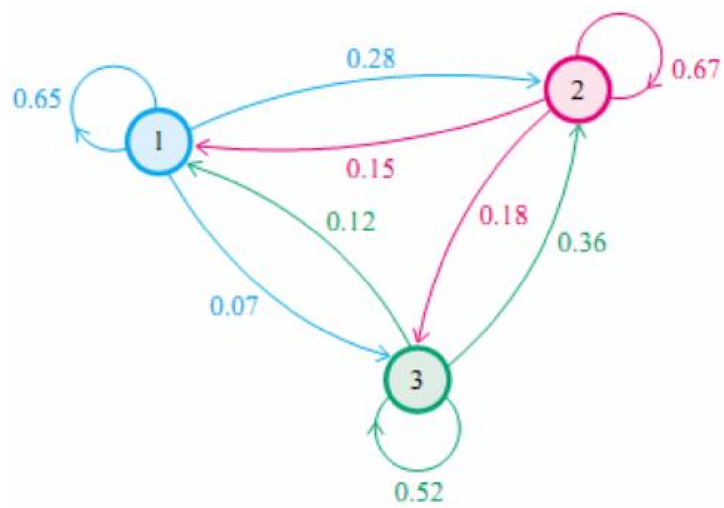
- 采样：给定概率分布p(x)，如何在计算机中生成它的若干样本?
- 方法：马尔科夫链模型
- 考虑某随机过程π，它的状态有n个，用1~n表示。记在当前时刻t时位于i状态，它在t+1时刻位于j状态的概率为P(i,j)=P(j|i)：即状态转移的概率只依赖于前一个状态。



举例

- 假定按照经济状况将人群分成上、中、下三个阶层，用1、2、3表示。假定当前处于某阶层只和上一代有关，即：考察父代为第i阶层，则子代为第j阶层的概率。假定为如下转移概率矩阵：

$$
\begin{array}{cc}
P & \text{子代} \\
\text{父代} & \begin{bmatrix} 0.65 & 0.28 & 0.07 \\ 0.15 & 0.67 & 0.18 \\ 0.12 & 0.36 & 0.52 \end{bmatrix}
\end{array}
$$



概率转移矩阵

- 显然，第n+1代中处于第j个阶层的概率为：

$$\pi(X_{n+1}=j)=\sum_{i=1}^{n}\pi(X_n=i)\cdot P(X_{n+1}=j\mid X_n=i)$$

- 因此，矩阵P即贝叶斯网络中描述的(条件)概率转移矩阵。
  - 第i行元素表示：在上一个状态为i时的分布概率，即：每一行元素的和为1。

初始概率 $\pi = [0.21, 0.68, 0.1]$的迭代结果

第n代	第1阶层	第2阶层	第3阶层
0	0.21	0.68	0.11
1	0.252	0.554	0.194
2	0.27	0.512	0.218
3	0.278	0.497	0.225
4	0.282	0.49	0.226
5	0.285	0.489	0.225
6	0.286	0.489	0.225
7	0.286	0.489	0.225
8	0.286	0.488	0.225
9	0.286	0.489	0.225
10	0.286	0.489	0.225

初始概率 $\pi = [0.75, 0.15, 0.1]$ 的迭代结果

第n代	第1阶层	第2阶层	第3阶层
0	0.75	0.15	0.1
1	0.522	0.347	0.132
2	0.407	0.426	0.167
3	0.349	0.459	0.192
4	0.318	0.475	0.207
5	0.303	0.482	0.215
6	0.295	0.485	0.22
7	0.291	0.487	0.222
8	0.289	0.488	0.225
9	0.286	0.489	0.225
10	0.286	0.489	0.225

马尔科夫随机过程的平稳分布

- 初始概率不同，但经过若干次迭代，π 最终稳定收敛在某个分布上。
- 转移概率矩阵P的性质，而非初始分布的性质。事实上，上述矩阵P的n次幂，每行都是(0.286,0.489,0.225)，n>20
- 如果一个非周期马尔科夫随机过程具有转移概率矩阵P，且它的任意两个状态都是连通的，则 $\lim_{n\to\infty} P_{ij}^n$ 存在，记做 $\lim_{n\to\infty} P_{ij}^n = \pi(j)$。

马尔科夫随机过程的平稳分布

- 事实上，下面两种写法等价：

$$\lim_{n\to\infty} P_{ij}^n = \pi(j) \qquad \lim_{n\to\infty} P^n = \begin{bmatrix} \pi(1) & \pi(2) & \ldots & \pi(n) \\ \pi(1) & \pi(2) & \ldots & \pi(n) \\ \vdots & \vdots & \ddots & \vdots \\ \pi(1) & \pi(2) & \cdots & \pi(n) \end{bmatrix}$$

- 同时，若某概率分布 $\pi P=\pi$，说明
  - 该多项分布 π 是状态转移矩阵P的平稳分布；
  - 线性方程 $xP=x$ 的非负解为 π，而 P^n 唯一，因此 π 是线性方程 $xP=x$ 的唯一非负解。

马尔科夫随机过程与采样

- 上述平稳分布的马尔科夫随机过程对采样带来很大的启发：对于某概率分布π，生成一个能够收敛到概率分布π的马尔科夫状态转移矩阵P，则经过有限次迭代，一定可以得到概率分布π。
- 该方法可使用Monte Carlo模拟来完成，称之为MCMC(Markov Chain Monte Carlo)。

细致平稳条件

- 从稳定分布满足πP=π可以抽象出如下定义：
- 如果非周期马尔科夫过程的转移矩阵P和分布π(x)满足

$$\forall i, j, \pi(i)P(i,j) = \pi(j)P(j,i)$$

- 则π(x)是马尔科夫过程的平稳分布。上式又被称作细致平稳条件(detailed balance condition)。
  - P(i,j)为矩阵P的第i行第j列，其意思为前一个状态为i时，后一个状态为j的概率：即P(j|i)，因此，有时也写成P(i→j)
  - 细致平稳的理解：根据定义，对于任意两个状态i，j，从i转移到j的概率和从j转移到i的概率相等。可直观的理解成每一个状态都是平稳的。

细致平稳条件和平稳分布的关系

- 根据马尔科夫过程的定义：$\pi(j)=\sum_{i=1}^{n}\pi(i)\cdot P(i,j)$
- 根据细致平稳条件：

$$\forall i,j,\pi(i)P(i,j)=\pi(j)P(j,i)$$

- 得：

$$\pi(j)=\sum_{i=1}^{n}\pi(j)\cdot P(j,i)$$

- 从而：

$$\pi=\pi\cdot P$$

设定接受率

- 假定当前马尔科夫过程的转移矩阵为Q，对于给定分布p，一般的说，$p(i)q(i,j) \neq p(j)q(j,i)$
- 通过加入因子α的方式，使得上式满足细致平稳条件 $p(i)q(i,j)\alpha(i,j) = p(j)q(j,i)\alpha(j,i)$
- 满足等式的因子α有很多，根据对称性，可以取：$\alpha(i,j) = p(j)q(j,i)$，$\alpha(j,i) = p(i)q(i,j)$
- 根据接受率α改造转移矩阵Q：

$$p(i)\underline{\underline{q(i,j)\alpha(i,j)}} = p(j)\underline{\underline{q(j,i)\alpha(j,i)}}$$

MCMC：Metropolis-Hastings算法

- 根据需要满足的细致平稳条件

$$p(i)q(i,j)\alpha(i,j)=p(j)q(j,i)\alpha(j,i)$$

- 若令α (j,i)=1，则有：$p(i)q(i,j)\alpha(i,j)=p(j)q(j,i)$
- 从而：

$$\alpha(i,j)=\frac{p(j)q(j,i)}{p(i)q(i,j)}$$

- 将接受率置为恒小于1，从而

$$\alpha(i,j)=\min\left(\frac{p(j)q(j,i)}{p(i)q(i,j)},1\right)$$

Metropolis-Hastings算法

- 初始化马尔科夫过程初始状态$I=i_0$
- 对$t=0,1,2,3\ldots$
  - 第t时刻马尔科夫过程初始状态i_t,采样$q=q(j|i_t)$
  - 从均匀分布中采样$u\in[0,1]$
  - 如果 $$u<\alpha(i,j)=\min\left(\frac{p(j)q(j,i)}{p(i)q(i,j)},1\right)$$

    则接受状态j，即$i_{t+1}=j$

    否则，不接受状态j，即$i_{t+1}=i$

改造MCMC算法

- 分析MCMC：
  - 需要事先给定马尔科夫过程的转移矩阵P；
  - 有一定的拒绝率。
- 若需要采样二维联合分布p(x,y)，固定x，得

$$p(x_1,y_1)\alpha_{x_1}(y_1,y_2)=p(x_1,y_2)\alpha_{x_1}(y_2,y_1)$$

$$\Rightarrow p(x_1)p(y_1\mid x_1)\alpha_{x_1}(y_1,y_2)=p(x_1)p(y_2\mid x_1)\alpha_{x_1}(y_2,y_1)$$

$$\Rightarrow \alpha_{x_1}(y_1,y_2)=p(y_2\mid x_1)\text{，}\alpha_{x_1}(y_2,y_1)=p(y_1\mid x_1)$$

$$\Rightarrow \alpha_{x_1}(y_{cur},y_{other})=p(y_{other}\mid x_1)\text{，}\alpha_{x_1}(y_{other},y_{cur})=p(y_{cur}\mid x_1)$$

  - 若固定y，可得到对偶的结论。

二维Gibbs采样算法

- 由 $\begin{cases}\alpha_{x_1}\left(y_{cur}, y_{other}\right) = p\left(y_{other} \mid x_1\right) \\ \alpha_{y_1}\left(x_{cur}, x_{other}\right) = p\left(x_{other} \mid y_1\right)\end{cases}$
- 很容易得到二维Gibbs采样算法：
  - 随机初始化(X,Y)=(x_0,y_0)
  - 对t=0,1,2…，循环采样：

$$\begin{cases}y_{t+1} = p\left(y \mid x_t\right) \\ x_{t+1} = p\left(x \mid y_{t+1}\right)\end{cases}$$

将二维Gibbs采样推广到高维

- 随机初始化 $(X_1, X_2 \cdots X_n) = \left(x_1^{(0)}, x_2^{(0)} \cdots, x_n^{(0)}\right)$
- 对t=0,1,2...，循环采样：

$$
\begin{cases}
x_1^{(t+1)} = p\left(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_n^{(t)}\right) \\
x_2^{(t+1)} = p\left(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \cdots, x_n^{(t)}\right) \\
\cdots \\
x_i^{(t+1)} = p\left(x_i \mid x_1^{(t)}, \cdots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \cdots, x_n^{(t)}\right) \\
\cdots \\
x_n^{(t+1)} = p\left(x_n \mid x_1^{(t+1)}, x_2^{(t+1)}, \cdots, x_{n-1}^{(t+1)}\right)
\end{cases}
$$

- 很显然，主题模型LDA中采样更新即采取的以上策略。

参数估计总结

- 给定样本$X_1,X_2 \ldots X_n$，求系统参数θ
  - 极大似然估计：Maximum Likelihood Estimate

$$P(\theta \mid X) = \prod_i P(\theta \mid x_i)$$

  - 极大后验概率：Maximum A Posteriori

$$P(\theta \mid X) = \frac{P(X \mid \theta)P(\theta)}{P(X)} \propto P(X \mid \theta)P(\theta) = \prod_i P(x_i \mid \theta)P(\theta)$$

- 若存在隐变量：
  - EM算法——衍生品：随机EM、MAP-EM、IP算法
    - GMM、pLSA、HMM、CRF
  - 采样：MCMC、Gibbs

另一个思路

- 变分推导(variational inference)是一般的确定性的近似推导算法。
- 基本思想：选择一个容易计算的近似分布q(x)，它能够尽可能的接近真正的后验分布p(x|D)。
  - 通过降低约束条件，在精度和速度上折中。
- 问题：如何定义两个分布的相似度？

变分的提法

- 假定p*(x)是真实(难解的)分布，q(x)是某个近似的(容易的)分布——如多元高斯分布或者多个简单分布的乘积。
- 假定q(x)有若干自由参数需要估计，我们需要优化这些未知参数使得q近似于p*。
- 一个显然的损失函数是最小化KL散度

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

变分目标函数分析

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

- 上式关于后验概率p*的期望是不容易计算的，作为替代，将上述KL散度变成“逆KL散度”(reverse KL divergence)

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

- 第二个式子的主要优点是转换为计算关于q的期望(而q是关于未知参数的简单分布)；进一步，由于p(D)是归一化因子

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

- 上式变成：$J(q)=KL(q \| \tilde{p})=\sum_x q(x)\log\frac{q(x)}{\tilde{p}(x)}$

新目标函数的可行性 $J(q)=KL(q\|\tilde{p})$

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

- 由于Z是常数，通过最小化J(q)，能够使得q接近p*。

变分和EM的联系

- 因为KL散度总是非负的，J(p)是NLL的上界
  - negative log likelihood

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

  - 进一步：

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

- 因此，L(q)是似然函数的下界，当q=p*时取等号。
  - 可取等号，说明下界是紧的(tight)
- EM和变分
  - 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；
  - 相同的思维：不断迭代，得到更好的下界。
  - 不断上升。

思考：目标函数的物理含义

- 定义能量 $E(x) = -\log \tilde{p}(x)$
- 目标函数是能量的期望减去系统的熵。J(q)被叫做“变分自由能”或“Helmholtz free energy”。

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

$$= E_q\left(\log \frac{q(x)}{\tilde{p}(x)}\right) = E_q\left(\log q(x) - \log \tilde{p}(x)\right)$$

$$= E_q\left(\log q(x)\right) + E_q\left(-\log \tilde{p}(x)\right)$$

$$\stackrel{\triangle}{=\!=} -H(X) + E_q\left(E(x)\right)$$

思考：似然函数期望与目标函数

- 负似然函数NLL的期望，加上一个惩罚项——近似分布与先验分布的KL距离。

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

$$=E_q\left(\log\frac{q(x)}{p(x,D)}\right)=E_q\left(\log\frac{q(x)}{p(x)p(D\,|\,x)}\right)$$

$$=E_q\left(\log\frac{1}{p(D\,|\,x)}+\log\frac{q(x)}{p(x)}\right)=E_q\left(\log\frac{1}{p(D\,|\,x)}\right)+E_q\left(\log\frac{q(x)}{p(x)}\right)$$

$$=E_q(-\log p(D\,|\,x))+KL(q\,\|\,p)$$

两个KL散度的区别

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

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

  - 如果p(x)=0，q(x)>0，则KL为无穷大。因此，当p(x)=0时必须保证q(x)=0。即：该公式是对待求分布q“0强制”(zero forcing)的。从而，q往往被低估。

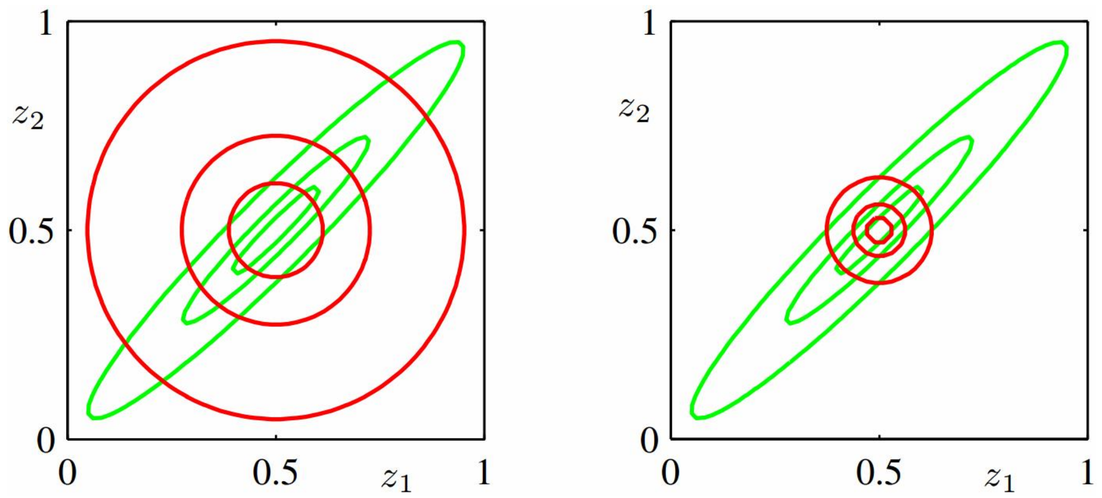
- KL(p||q)，又称为M-投影，矩投影(moment projection)

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

  - 如果p(x)>0，q(x)=0，则KL为无穷大。因此，当p(x)>0时必须保证q(x)>0。即：该公式是对待求分布q“0避免”(zero avoiding)的。从而，q往往被高估。

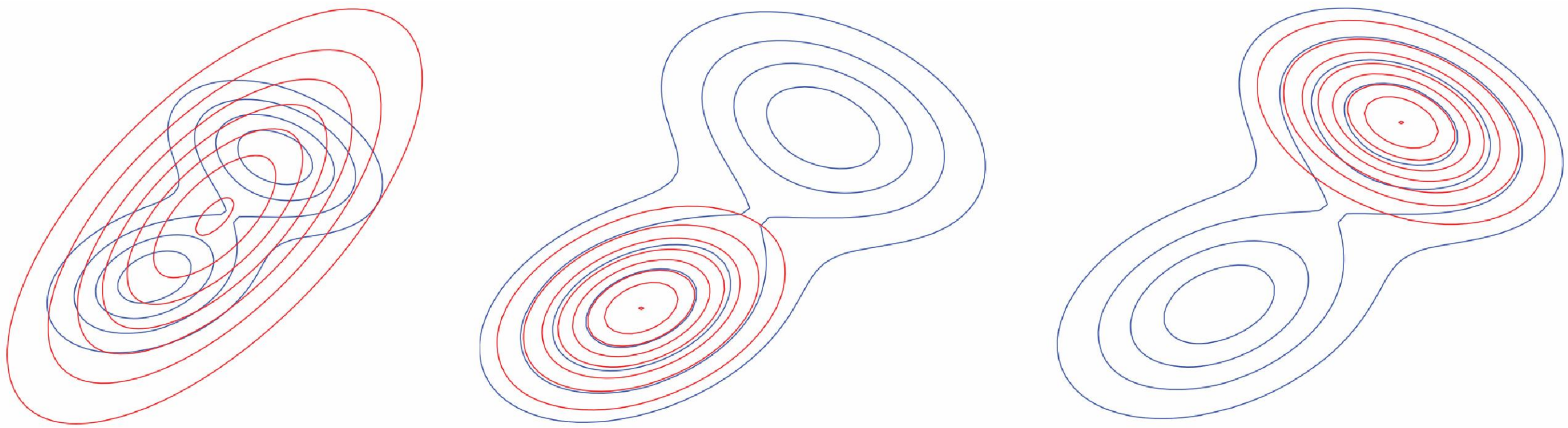
两个KL散度的区别

- 绿色曲线是真实分布p的等高线；红色曲线是使用近似$p(z_1,z_2)=p(z_1)p(z_2)$得到的等高线
  - 左：KL(p||q)：zero avoiding
  - 右：KL(q||p)：zero forcing



两个KL散度的区别

- 蓝色曲线是真实分布p的等高线；红色曲线是单模型近似分布q的等高线。
  - 左：KL(p||q)：q趋向于覆盖p
  - 中、右：KL(q||p)：q能够锁定某一个峰值



两个KL散度之间的联系

- 给定分布p和q的距离定义

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

- p和q的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：

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

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

两个KL散度之间的联系

$$u=\frac{q(x)}{p(x)} \Rightarrow \begin{cases} f(u)=u^{\frac{1-\alpha}{2}} \\ g(u)=\log u \end{cases} \Rightarrow \begin{cases} f'(u)=\frac{1-\alpha}{2}u^{-\frac{1+\alpha}{2}} \\ g'(u)=u^{-1} \end{cases} \Rightarrow \begin{cases} \frac{1-\alpha}{2}=1 \\ -\frac{1+\alpha}{2}=-1 \end{cases} \Rightarrow \begin{cases} \alpha=-1 \\ \alpha=1 \end{cases}$$

- 当 α =1时退化为KL(q||p)
- 当 α =-1时退化为KL(q||p)
- 当 α =0时？

Hellinger distance

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

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

$$=\int p(x)dx+\int q(x)dx-\int 2\sqrt{p(x)q(x)}dx$$

$$=\int\left(p(x)-2\sqrt{p(x)q(x)}+q(x)\right)dx$$

$$=\int\left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2dx$$

- 该距离满足三角不等式，是对称、非负距离

平均场方法(Mean field method)

- 最流行的变分方法之一是平均场近似。在这种方法中，假定后验概率能够近似分解为若干因子的乘积。
  - 思考：无向图中的“最大团” Hammersley-Clifford定理

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

- 我们的目标是解决最优化问题：$\min\limits_{q_1,\cdots q_D} KL(q\,\|\,p)$
- 平均场方法使得可以在若干边界分布qi上进行(依次)优化。事实上，很快将得知，有如下近似等式：

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

- 其中，未正则化的后验概率 $\tilde{p}(x)=p(x,D)$
- 关于除了x_j的所有其他变量的f(x)的期望 $E_{-q_j}[f(x)]$

平均场方法(Mean field method)

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

- 当更新q_j时，仅需要计算与x_j有公共边的那些变量即可——j的Markov毯包含的哪些结点。因为该方法使用相邻结点的期望(均值)，所以称作平均场。
- 思考Gibbs采样和变分：
  - Gibbs采样：使用邻居结点的采样值；
  - 变分：采用相邻结点的均值。
  - 这将使得变分往往比采样算法的更高效：用一个均值代替了大量的采样值。直观上，均值的信息是高密(dense)的，而采样值的信息是稀疏(sparse)的。

变分推导/似然下界L $J(q)=KL(q\|\tilde{p})=\sum_x q(x)log\frac{q(x)}{\tilde{p}(x)}$

$$L(q_j)\stackrel{\Delta}{==}-J(q_j)=-\sum_x q(x)log\frac{q(x)}{\tilde{p}(x)}$$

$$=\sum_x q(x)[\log\tilde{p}(x)-\log q(x)]$$

$$=\sum_x\prod_i q_i(x_i)\left[\log\tilde{p}(x)-\log\prod_i q_i(x_i)\right]$$

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

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

$$=\left(\sum_{x_j}q_j(x_j)\sum_{x_{-j}}\prod_{i\neq j}q_i(x_i)\log\tilde{p}(x)\right)-\left(\sum_{x_j}q_j(x_j)\log q_j(x_j)\right)+const$$

$$=\left(\sum_{x_j}q_j(x_j)\log f_j(x_j)\right)-\left(\sum_{x_j}q_j(x_j)\log q_j(x_j)\right)+const=-KL(q_j\|f_j)$$

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

变分推导最终结论

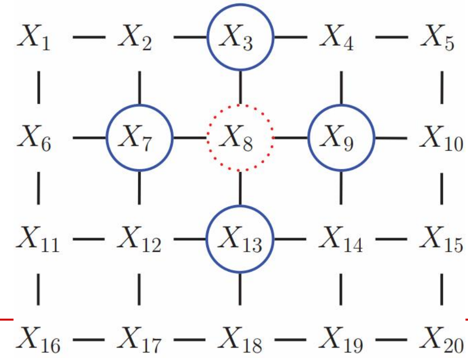
- 下界 $L(q_j)=-KL(q_j \| f_j)$ 取极大，则 $KL(q_j \| f_j)$ 取极小，此刻，要求二者分布相同。
- 由于 $\log f_j(x_j) \overset{\triangle}{=\!=} \sum_{x_{-j}} \prod_{i \neq j} q_i(x_i) \log \tilde{p}(x) = E_{-q_j}[\log \tilde{p}(x)]$
- 所以

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

- 忽略归一化因子

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

Ising model



- Ising模型是统计物理提出的MRF，它最初是用来对磁化行为建模。令$y_s \in \{-1,+1\}$表示原子的自旋，它的旋转角速度方向要么朝上，要么朝下。在某些环境下，表现为铁磁现象(ferro-magnets)：相邻结点的自旋方向趋近于同向；而其他环境中表现为反铁磁现象(anti-ferromagnets)，相邻结点的自旋方向趋近于相反。
- 可以使用MRF建模：连接相邻变量，然后定义团(clique)之间的势函数：

$$\varphi_{st}(y_s, y_t) = \begin{pmatrix} e^{w_{st}} & e^{-w_{st}} \\ e^{-w_{st}} & e^{w_{st}} \end{pmatrix}$$

势函数的系数 $\varphi_{st}(y_s, y_t) = \begin{pmatrix} e^{w_{st}} & e^{-w_{st}} \\ e^{-w_{st}} & e^{w_{st}} \end{pmatrix}$

- w_{st}是结点s和t之间的**耦合强度**(coupling strength)。如果两个结点没有连接，则设置$w_{st}=0$。假定权值矩阵W是对称阵，即$w_{st}=w_{ts}$；进一步假定所有的边有相同的强度，即$w_{st}=J\neq 0$。
- 如果所有的权值都为正(J>0)，则相邻结点的自旋趋向于同向，能够对**铁磁现象**建模：如果权值足够强，则结点的概率分布将只有两种状态：一部分结点是1状态，一部分结点是-1状态，这被称作系统的**基态**(groud states)。
  - 类比：将某状态认为是实际观测的图像，基态认为是去噪后的"干净"的图像。
- 同理，如果J<0可以对**反铁磁现象**建模。

使用变分做图像去噪

- 考虑图像的去噪问题：$x_i \in \{-1,+1\}$ 是隐藏在观测图像背后的干净图像的像素取值。为简洁方便，假定是二值图。x的联合分布假定具有如下先验形式：

$$p(x)=\frac{1}{Z_0}\exp\left(-E_0(x)\right),\qquad \text{其中}, E_0(x)=-\sum_{i=1}^{D}\sum_{j\in nbr_i}W_{ij}x_i x_j$$

- 似然函数

$$p(y\mid x)=\prod_i p(y_i\mid x_i)=\prod_i\left(\exp\left(\ln p(y_i\mid x_i)\right)\right)$$

$$=\exp\sum_i \ln p(y_i\mid x_i)=\exp\sum_i\left(L_i(x_i)\right)$$

后验概率

- 后验概率 $p(x \mid y) = \dfrac{p(y \mid x)p(x)}{p(y)} \propto p(y \mid x)p(x)$

$$\propto \left(\exp \sum_i (L_i(x_i))\right)(\exp(-E_0(x)))$$

$$= \exp\left(-E_0(x) + \sum_i L_i(x_i)\right)$$

$$\Rightarrow p(x \mid y) = \frac{1}{Z}\exp(-E(x))$$

- 其中，$E(x) = E_0(x) - \sum_i L_i(x_i)$

近似概率

- 根据后验概率形式：

$$p(x\mid y)=\frac{1}{Z}\exp(-E(x))=\frac{1}{Z}\exp\left(E_0(x)+\sum_i L_i(x_i)\right)$$

- 得到经验概率的对数：

$$\ln\tilde{p}(x)=\left(E_0(x)+\sum_i L_i(x_i)\right)=\sum_{i=1}^{D}\sum_{j\in nbr_i}W_{ij}x_i x_j+\sum_i L_i(x_i)$$

- 只考虑与i相关的部分：

$$\ln\tilde{p}(x)=x_i\sum_{j\in nbr_i}W_{ij}x_j+L_i(x_i)+const$$

- 从而：

$$q_i(x_i)\propto\exp\left(x_i\sum_{j\in nbr_i}W_{ij}\mu_j+L_i(x_i)\right)$$

根据公式：$q_i(x_i) \propto \exp\left(x_i \sum_{j \in nbr_i} W_{ij}\mu_j + L_i(x_i) \right)$

- 记平均场对结点i的影响为：$m_i = \sum_{j \in nbr_i} W_{ij}\mu_j$
- 进一步，记：$L_i^+ = L_i(+1),\ L_i^- = L_i(-1)$
- 则近似边缘后验概率为：

$$\begin{cases} q_i(x_i = 1) = \dfrac{e^{m_i + L_i^+}}{e^{m_i + L_i^+} + e^{-m_i + L_i^-}} = \dfrac{1}{1 + e^{-2m_i + L_i^- - L_i^+}} = sigm(2a_i) \\ q_i(x_i = -1) = sigm(-2a_i) \end{cases}$$

其中，$a_i = m_i + \dfrac{L_i^+ - L_i^-}{2}$

更新方程

- 结点i新的期望为：

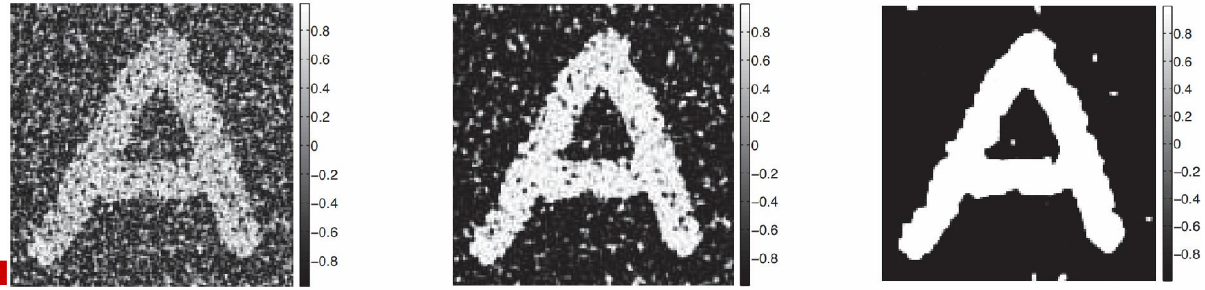
$$\mu_i = E_{q_i}(x_i) = q_i(x_i=+1)\cdot(+1) + q_i(x_i=-1)\cdot(-1)$$

$$= \frac{1}{1+e^{-2a_i}} - \frac{1}{1+e^{2a_i}} = \frac{e^{a_i}}{e^{a_i}+e^{-a_i}} - \frac{e^{-a_i}}{e^{-a_i}+e^{a_i}} = \tanh(a_i)$$

- 因此，更新方程为：

$$\mu_i = \tanh\left(\sum_{j\in nbr_i} W_{ij}\mu_j + \frac{L_i^+ - L_i^-}{2}\right)$$

迭代方程



- 根据上式很容易得到迭代公式：

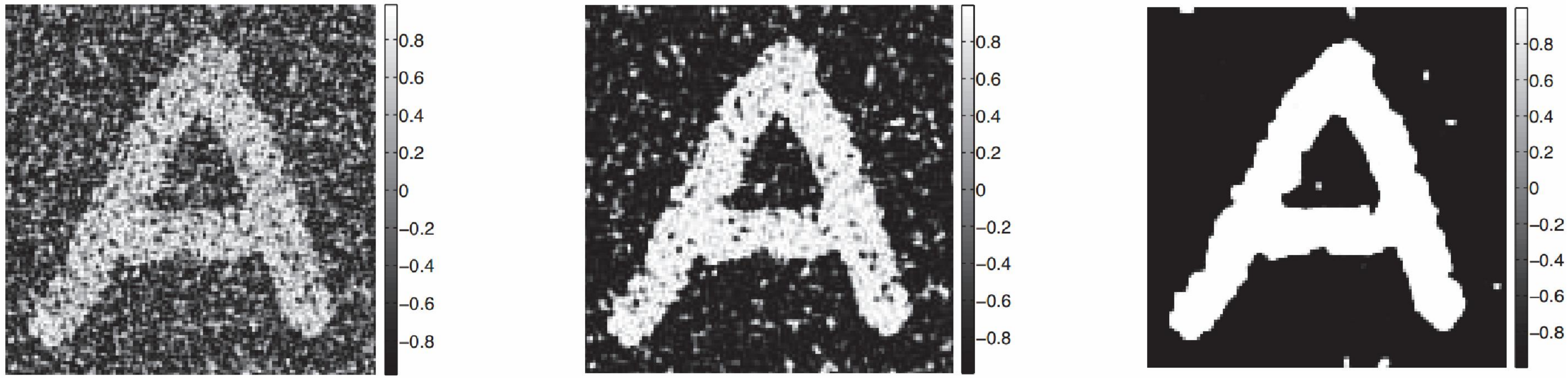
$$\mu_i^t = \tanh\left(\sum_{j \in nbr_i} W_{ij}\mu_i^{t-1} + \frac{L_i^+ - L_i^-}{2} \right)$$

- 实践中，往往需要增加衰减因子，得
  - damped updates：$1 > \lambda > 0$

$$\mu_i^t = (1-\lambda)\mu_i^{t-1} + \lambda \tanh\left(\sum_{j \in nbr_i} W_{ij}\mu_i^{t-1} + \frac{L_i^+ - L_i^-}{2} \right)$$

实际效果

- 2维Ising模型，先验权值都为1，使用衰减因子 $\lambda=0.5$ 并行更新。
  - 左：迭代1次；中：迭代3次；右：迭代15次。



变分贝叶斯(Variational Bayes,VB)

- 上述变分实践是计算给定模型参数，推断隐变量。此外，变分方法也可以推断参数本身。使用平均场方法，将后验概率写成参数各自分布的乘积，即得到变分贝叶斯方法(Variational Bayes, VB)。
- 变分贝叶斯：$p(\theta|D) \approx \prod_k q_k(\theta_k)$

高斯分布的变分贝叶斯Variational Bayes

- 使用变分贝叶斯推断一维高斯分布 $p(\mu,\lambda|D)$ 后验概率的参数。其中，λ为精度(方差的倒数)。为计算方便，使用共轭先验的形式。

$$p(\mu,\lambda)=N\left(\mu|\mu_0,(\kappa_0\lambda)^{-1}\right)\cdot Ga(\lambda|a_0,b_0)$$

- 近似分解得到如下形式：

$$q(\mu,\lambda)=q_\mu(\mu)q_\lambda(\lambda)$$

未正则化的对数后验

$$p(D\mid\mu,\lambda)=\sqrt{\frac{\lambda}{2\pi}}\cdot e^{-\frac{\lambda(x-\mu)^2}{2}}$$

$$p\left(\mu\mid\mu_0,(\kappa_0\lambda)^{-1}\right)=\sqrt{\frac{\kappa_0\lambda}{2\pi}}\cdot e^{-\frac{\kappa_0\lambda(\mu-\mu_0)^2}{2}}$$

目标函数

$$p(\lambda\mid a_0,b_0)=\frac{\beta^{a_0}\lambda^{a_0-1}e^{-b_0\lambda}}{\Gamma(a_0)}$$

$$\begin{aligned}
&\log\tilde{p}(\mu,\lambda)\\
&=\log p(\mu,\lambda,D)\\
&=\log p(D\mid\mu,\lambda)+\log p(\mu\mid\lambda)+\log p(\lambda)\\
&=\log\prod_{i=1}^{N}\sqrt{\frac{\lambda}{2\pi}}\cdot e^{-\frac{\lambda(x_i-\mu)^2}{2}}+\log\sqrt{\frac{\kappa_0\lambda}{2\pi}}\cdot e^{-\frac{\kappa_0\lambda(\mu-\mu_0)^2}{2}}+\log\frac{\beta^{a_0}\lambda^{a_0-1}e^{-b_0\lambda}}{\Gamma(a_0)}\\
&=\frac{N}{2}\log\lambda-\frac{\lambda}{2}\sum_{i=1}^{N}(x_i-\mu)^2+\frac{1}{2}\log(\kappa_0\lambda)-\frac{\kappa_0\lambda}{2}(\mu-\mu_0)^2\\
&+(a_0-1)\log\lambda-b_0\lambda+const
\end{aligned}$$

更新 $q_\mu(\mu)$

- 最优形式的 $q_\mu(\mu)$ 是通过计算关于 λ 的平均值获得的：

$$\log \tilde{p}(\mu,\lambda)=\frac{N}{2}\log\lambda-\frac{\lambda}{2}\sum_{i=1}^{N}(x_i-\mu)^2+\frac{1}{2}\log(\kappa_0\lambda)-\frac{\kappa_0\lambda}{2}(\mu-\mu_0)^2+(a_0-1)\log\lambda-b_0\lambda+const$$

$$\begin{aligned}\log q_\mu(\mu)&=E_{q_\lambda}\left(\log\tilde{p}(\mu,\lambda)\right)\\&=E_{q_\lambda}\left(-\frac{\lambda}{2}\sum_{i=1}^{N}(x_i-\mu)^2-\frac{\kappa_0\lambda}{2}(\mu-\mu_0)^2\right)+const\\&=-\frac{E_{q_\lambda}(\lambda)}{2}\left(\sum_{i=1}^{N}(x_i-\mu)^2+\kappa_0(\mu-\mu_0)^2\right)+const\end{aligned}$$

由$q_\mu(\mu)$得到的参数等式

- 对比标准整体分布的对数式，得到：

$$\log q_\mu(\mu) = -\frac{E_{q_\lambda}(\lambda)}{2}\left(\sum_{i=1}^{N}(x_i-\mu)^2+\kappa_0(\mu-\mu_0)^2\right)+const$$

$$\begin{cases} \mu_N = \dfrac{\kappa_0\mu_0 + N\overline{x}}{\kappa_0 + N} \\ \kappa_N = (\kappa_0 + N)E_{q_\lambda}(\lambda) \end{cases}$$

- 目前尚未知$q_\lambda(\lambda)$，因为无法计算$E_{q_\lambda}(\lambda)$，继续考察$q_\lambda(\lambda)$。

更新 $q_\lambda(\lambda)$

- 最优形式的$q_\lambda(\lambda)$是通过计算关于 μ 的平均值获得的：

$$\log \widetilde{p}(\mu,\lambda)=\frac{N}{2}\log\lambda-\frac{\lambda}{2}\sum_{i=1}^{N}(x_i-\mu)^2+\frac{1}{2}\log(\kappa_0\lambda)-\frac{\kappa_0\lambda}{2}(\mu-\mu_0)^2+(a_0-1)\log\lambda-b_0\lambda+const$$

$$\log q_\lambda(\lambda)=E_{q_\mu}\left(\log\widetilde{p}(\mu,\lambda)\right)$$

$$=E_{q_u}\left(\frac{N}{2}\log\lambda-\frac{\lambda}{2}\sum_{i=1}^{N}(x_i-\mu)^2+\frac{1}{2}\log(\kappa_0\lambda)-\frac{\kappa_0\lambda}{2}(\mu-\mu_0)^2+(a_0-1)\log\lambda-b_0\lambda\right)+const$$

$$=\frac{N}{2}\log\lambda+\frac{1}{2}\log\lambda+(a_0-1)\log\lambda-b_0\lambda-\frac{\lambda}{2}E_{q_u}\left(\kappa_0(\mu-\mu_0)^2+\sum_{i=1}^{N}(x_i-\mu)^2\right)+const$$

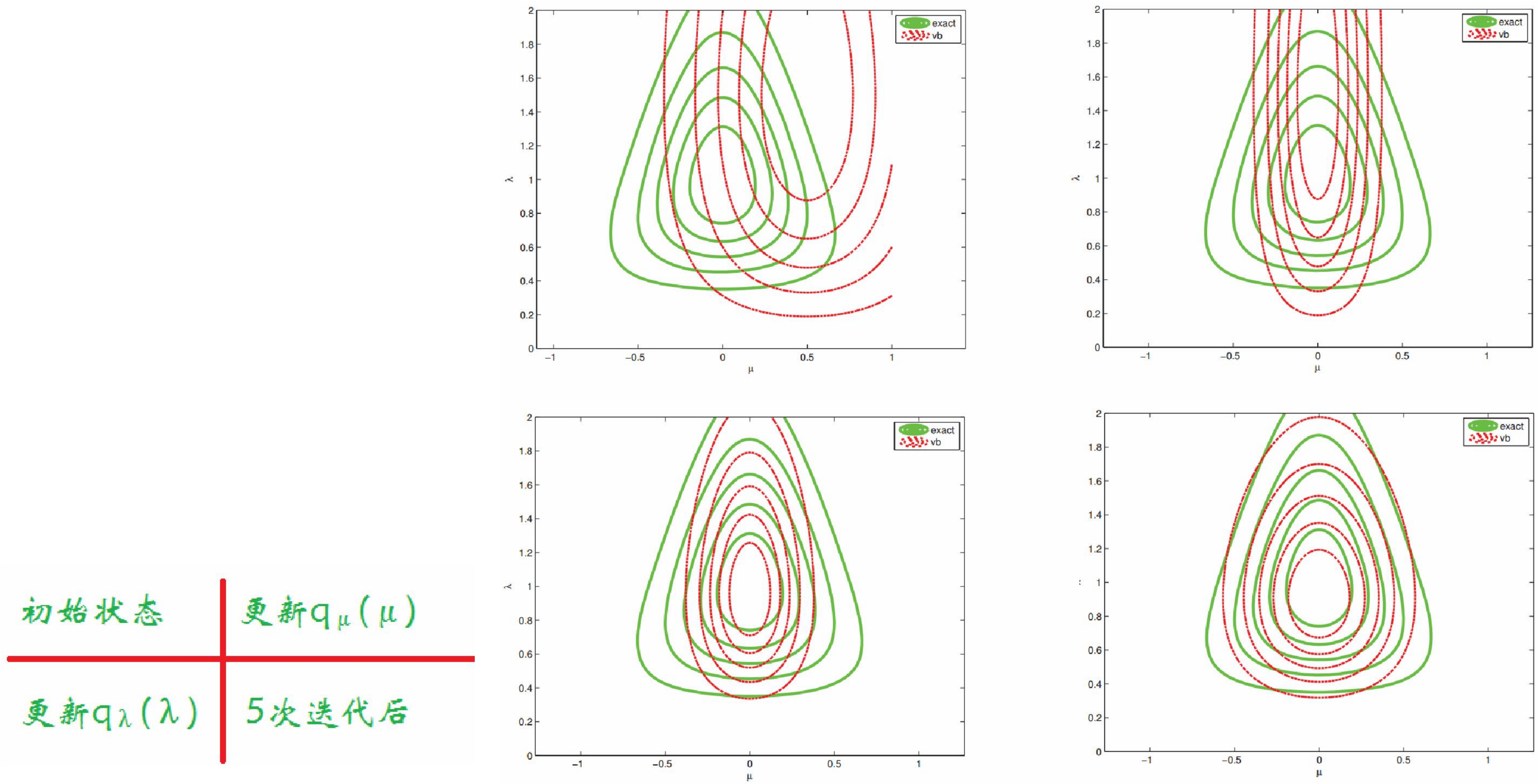
由$q_\lambda(\lambda)$得到的参数等式

- 对比标准整体分布的对数式，得到：

$$\log q_\lambda(\lambda)=\frac{N}{2}\log\lambda+\frac{1}{2}\log\lambda+(a_0-1)\log\lambda-b_0\lambda-\frac{\lambda}{2}E_{q_u}\left(\kappa_0(\mu-\mu_0)^2+\sum_{i=1}^{N}(x_i-\mu)^2\right)+const$$

$$\begin{cases} a_N=a_0+\dfrac{N+1}{2} \\ b_N=b_0+\dfrac{1}{2}E_{q_u}\left(\kappa_0(\mu-\mu_2)^2+\sum_{i=1}^{N}(x_i-\mu)^2\right) \\ =b_0+\kappa_0\left(E\left(\mu^2\right)+\mu_0^2-2E(\mu)\mu_0\right)+\dfrac{1}{2}\sum_{i=1}^{N}\left(x_i^2+E\left(\mu^2\right)-2E(\mu)x_i\right) \end{cases}$$

变分参数估计实例



变分总结

- 变分既能够推断隐变量的分布，也能推断未知参数的分布，是非常有力的参数学习工具。其难点在于公式演算略显复杂，和采样相对：一个容易计算但速度慢，一个不容易计算但运行效率高。
- 平均场方法的变分推导，对离散和连续的隐变量都适用。在平均场方法的框架下，变分推导一次更新一个分布，其本质为坐标上升。可以使用模式搜索(pattern search)、基于参数的扩展(parameter expansion)等方案加速。
- 有时候，假定所有变量都是独立是不符合实际的，可以使用结构化平均场(structured mean field)，将变量分成若干组，每组之间是独立的。
- 变分除了能够和贝叶斯理论相配合得到VB，还能进一步与EM算法结合，得到VBEM，用于带隐变量和未知参数的推断。
  - 如GMM、LDA

参考文献

- Machine Learning:A Probabilistic Perspective, Chapter 21, Kevin P. Murphy, The MIT Press, 2012
- Pattern Recognition and Machine Learning Chapter 10,11, Christopher M. Bishop, Springer-Verlag, 2006

我们在这里

七月算法 http://www.julyedu.com/

- 视频/课程/社区

- 七月题库APP：Android/iOS
  - http://www.julyapp.com/
- 微博
  - @研究者July
  - @七月题库
  - @邹博_机器学习
- 微信公众号
  - julyedu

感谢大家！

恳请大家批评指正！