机器读心术之神经网络与深度学习 第6周

法律声明

【声明】本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

http://edu.dataguru.cn

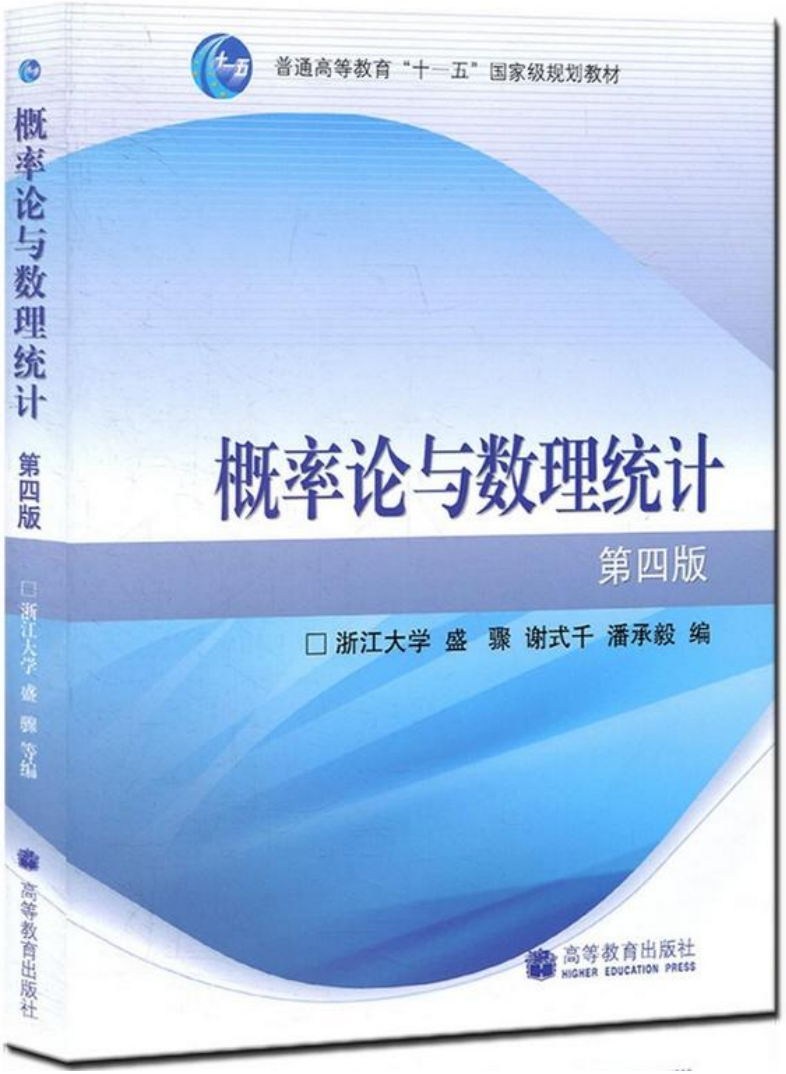
关注炼数成金企业微信

- 提供全面的数据价值资讯，涵盖商业智能与数据分析、大数据、企业信息化、数字化技术等，各种高性价比课程信息，赶紧掏出您的手机关注吧！

主要内容

- 从马尔科夫链说起
- MCMC与Metropolis算法
- Gibbs采样
- 玻尔兹曼机与Gibbs采样的联系
- 受限玻尔兹曼机
- CD学习算法

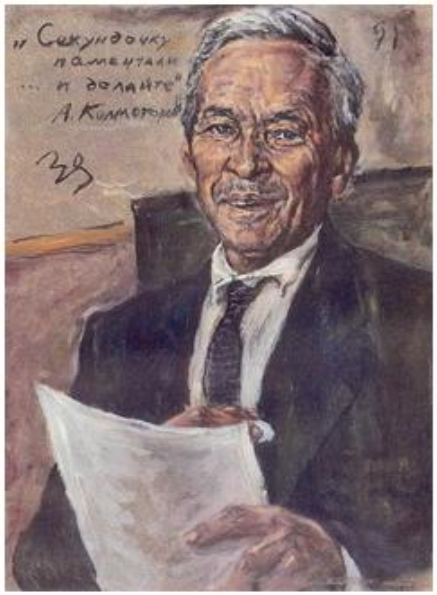
参考书


普通高等教育“十一五”国家级规划教材
概率论与数理统计
第四版
浙江大学 盛 骤 谢式千 潘承毅 编
高等教育出版社
HIGHER EDUCATION PRESS
概率论与数理统计 第四版
浙江大学 盛 骤 等编
高等教育出版社

马尔科夫其人

- 彼得堡数学学派：切比雪夫，马尔科夫和李雅普诺夫
- 学派主要贡献：复兴概率论，把概率论从濒临衰亡的境地挽救出来，恢复其作为一门数学学科的地位，并把它推进到现代化的门槛
- 马尔科夫过程和马尔科夫链

马尔可夫
（A.A.Markov，1856－1922）



柯尔莫哥洛夫
（Andrey N. Kolmogorov，1903－1987）

随机过程

- 浙大书第300页
- 一些实例

设 T 是一无限实数集. 我们把依赖于参数 $t\in T$ 的一族(无限多个)随机变量称为**随机过程**,记为$\{X(t),t\in T\}$,这里对每一个 $t\in T$,$X(t)$是一随机变量. T 叫做**参数集**. 我们常把 t 看作为时间,称 $X(t)$为时刻 t 时过程的**状态**,而$X(t_1)=x$(实数)说成是 $t=t_1$ 时过程处于状态 x. 对于一切 $t\in T$,$X(t)$所有可能取的一切值的全体称为随机过程的**状态空间**.

随机过程还可依时间(参数)是连续或离散进行分类. 当时间集 T 是有限或无限区间时,称$\{X(t),t\in T\}$为**连续参数随机过程**(以下如无特别指明,"随机过程"总是指连续参数而言的). 如果 T 是离散集合,例如 $T=\{0,1,2,\cdots\}$,则称$\{X(t),t\in T\}$为**离散参数随机过程或随机序列**,此时常记成$\{X_n,n=0,1,2,\cdots\}$

随机过程的马尔科夫性与马尔科夫过程

- 浙大书第319页

在物理学中，很多确定性现象遵从如下演变原则：由时刻 t_0 系统或过程所处的状态，可以决定系统或过程在时刻 $t>t_0$ 所处的状态，而无需借助于 t_0 以前系统或过程所处状态的历史资料. 如微分方程初值问题所描绘的物理过程就属于这类确定性现象. 把上述原则延伸到随机现象，即当一物理系统或过程遵循的是某种统计规律时，可仿照上面的原则，引入以下的**马尔可夫性**或**无后效性**：过程（或系统）在时刻 t_0 所处的状态为已知的条件下，过程在时刻 $t>t_0$ 所处状态的条件分布与过程在时刻 t_0 之前所处的状态无关. 通俗地说，就是在已经知道过程“现在”的条件下，其“将来”不依赖于“过去”.

现用分布函数来表述马尔可夫性. 设随机过程 $\{X(t), t\in T\}$ 的状态空间为 I. 如果对时间 t 的任意 n 个数值 $t_1<t_2<\cdots<t_n$，$n\geqslant 3$，$t_i\in T$，在条件 $X(t_i)=x_i$，$x_i\in I$，$i=1,2,\cdots,n-1$ 下，$X(t_n)$ 的条件分布函数恰等于在条件 $X(t_{n-1})=x_{n-1}$ 下 $X(t_n)$ 的条件分布函数，即

$$\begin{aligned}&P\{X(t_n)\leqslant x_n \mid X(t_1)=x_1, X(t_2)=x_2, \cdots, X(t_{n-1})=x_{n-1}\}\\&\quad=P\{X(t_n)\leqslant x_n \mid X(t_{n-1})=x_{n-1}\}, x_n\in \mathbf{R},\end{aligned}\tag{1.1}$$

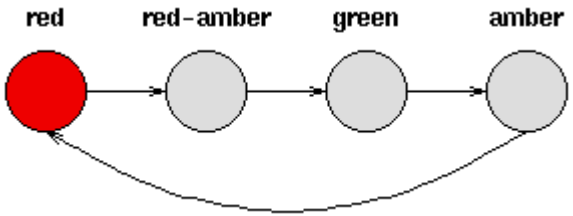
或写成

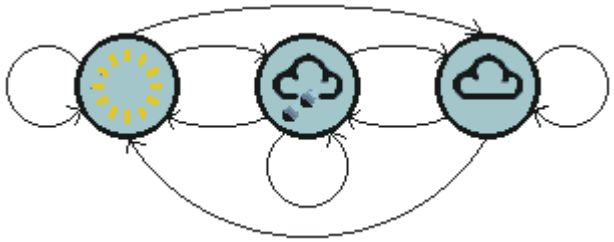
$$F_{t_n\mid t_1\cdots t_{n-1}}(x_n,t_n\mid x_1,x_2,\cdots,x_{n-1};t_1,t_2,\cdots,t_{n-1})=F_{t_n\mid t_{n-1}}(x_n,t_n\mid x_{n-1},t_{n-1}),$$

则称过程 $\{X(t), t\in T\}$ 具有马尔可夫性或无后效性，并称此过程为**马尔可夫过程**.

马尔科夫链

- 时间和状态都是离散的马尔科夫过程称为马尔科夫链
- 转移概率，转移概率矩阵，齐次马氏链，一步转移概率矩阵



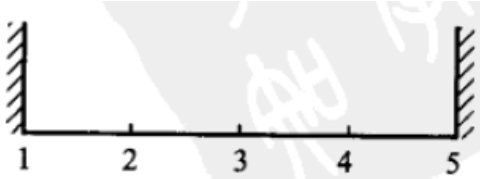


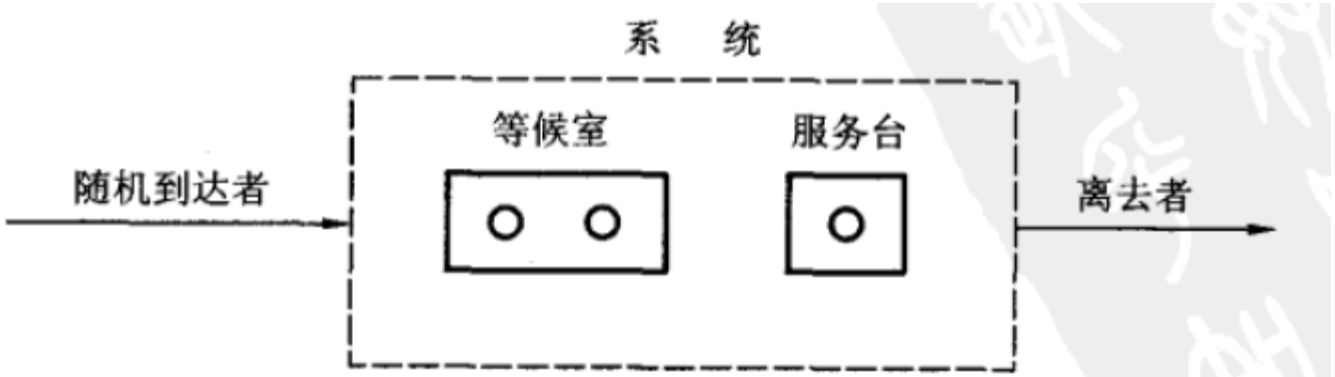
$$
\begin{array}{cc} & Today \\ Yesterday & \begin{array}{c} \\ \text{sun} \\ \text{cloud} \\ \text{rain} \end{array} \end{array}
\begin{array}{c}
\begin{array}{ccc} \text{sun} & \text{cloud} & \text{rain} \end{array} \\
\begin{bmatrix} 0.50 & 0.375 & 0.125 \\ 0.25 & 0.125 & 0.625 \\ 0.25 & 0.375 & 0.375 \end{bmatrix}
\end{array}
$$

随机游动

- 浙大书第321页
- 反射壁
- 吸收壁
- 排队模型

$$P=\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 & 0 \\ 0 & 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}.$$





求多步转移概率

- C-K方程，转化为求一步转移概率矩阵的幂
- 矩阵分解为标准性

C－K 方程也可写成矩阵形式：

$$\boldsymbol{P}(u+v)=\boldsymbol{P}(u)\boldsymbol{P}(v). \qquad (2.1)'$$

利用 C－K 方程我们容易确定 n 步转移概率. 事实上，在(2.1)′式中令 $u=1, v=n-1$，得递推关系：

$$\boldsymbol{P}(n)=\boldsymbol{P}(1)\boldsymbol{P}(n-1)=\boldsymbol{P}\boldsymbol{P}(n-1),$$

从而可得

$$\boldsymbol{P}(n)=\boldsymbol{P}^n. \qquad (2.3)$$

就是说，对齐次马氏链而言，n 步转移概率矩阵是一步转移概率矩阵的 n 次方.

进而可知，齐次马氏链的有限维分布可由初始分布与一步转移概率完全确定.

遍历性

- 浙大书第328页

一般，设齐次马氏链的状态空间为 I，若对于所有 $a_i, a_j \in I$，转移概率 $P_{ij}(n)$ 存在极限

$$\lim_{n\to\infty} P_{ij}(n) = \pi_j \quad (\text{不依赖于 } i)$$

或

$$\boldsymbol{P}(n) = \boldsymbol{P}^n \xrightarrow[(n\to+\infty)]{} \begin{bmatrix} \pi_1 & \pi_2 & \cdots & \pi_j & \cdots \\ \pi_1 & \pi_2 & \cdots & \pi_j & \cdots \\ \vdots & \vdots & & \vdots & \\ \pi_1 & \pi_2 & \cdots & \pi_j & \cdots \\ \vdots & \vdots & & \vdots & \end{bmatrix}.$$

则称此链具有**遍历性**. 又若 $\sum_j \pi_j = 1$，则同时称 $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots)$ 为链的**极限分布**.

遍历性

定理 设齐次马氏链$\{X_n, n\geqslant 1\}$的状态空间为$I=\{a_1, a_2, \cdots, a_N\}$，$\boldsymbol{P}$是它的一步转移概率矩阵，如果存在正整数$m$，使对任意的$a_i, a_j \in I$，都有

$$P_{ij}(m)>0, \quad i,j=1,2,\cdots,N, \tag{3.1}$$

则此链具有遍历性，且有极限分布$\boldsymbol{\pi}=(\pi_1, \pi_2, \cdots, \pi_N)$，它是方程组

$$\boldsymbol{\pi}=\boldsymbol{\pi P} \text{ 或即 } \pi_j=\sum_{i=1}^{N}\pi_i p_{ij}, j=1,2,\cdots,N \tag{3.2}$$

的满足条件

$$\pi_j>0, \sum_{j=1}^{N}\pi_j=1 \tag{3.3}$$

的唯一解.

应用

- 排队论等统计模型建模
- 语音识别
- 基因预测
- 搜索引擎鉴别网页质量——PR值

Google创始人拉里-佩奇和谢尔盖-布林

- 1998年，互联网的狂热达到了颠峰，网络正处于“信息爆炸”状态，唯一的问题是怎样去查找信息。此刻，两名不为人所知的年轻的计算机专业研究生，在斯坦福大学的宿舍里经常一待就是一通宵，他们是拉里·佩奇和谢尔盖·布林。他们想出了在互联网上寻找信息的方法，并决定放弃学业，将想法商业化。1998年9月，布林从老师大卫·切瑞顿和一位斯坦福校友（Sun的共同创始人Andy Bechtolsheim）那里顺利地拿到了第一笔投资：10万美元。依靠这10万美元，在朋友的一个车库里，布林和佩奇开始了谷歌的征程。

投资人大卫·切瑞顿

- http://www.dataguru.cn/article-5221-1.html

Page Rank与马尔科夫过程

- 这是Google最核心的算法，用于给每个网页价值评分，是Google“在垃圾中找黄金”的关键算法，这个算法成就了今天的Google

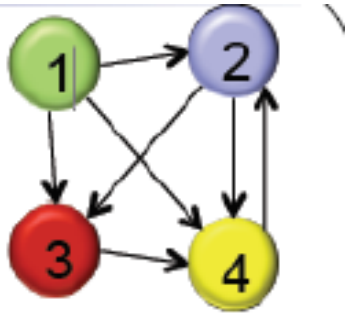
PageRank vector **q** is defined as $q = Gq$

where $G = \alpha S + (1-\alpha)\frac{1}{n}U$

- S is the destination-by-source stochastic matrix,
- U is all one matrix.
- n is the number of nodes
- α is the weight between 0 and 1 (e.g., 0.85)

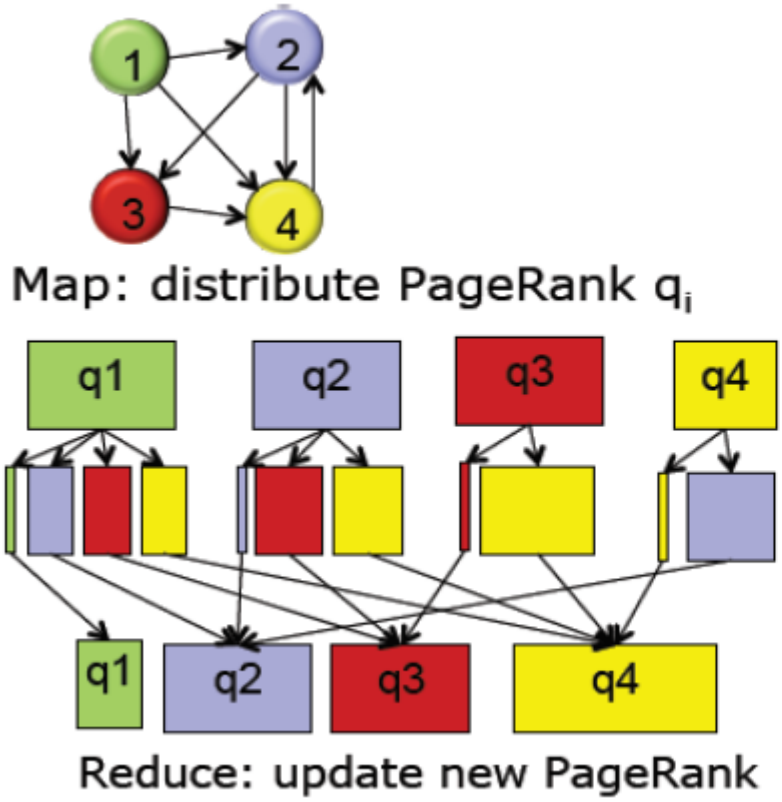
Algorithm: Iterative powering for finding the first eigen-vector

$$q^{next} = Gq^{cur}$$



$$G = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 1/2 & 0 & 0 \\ 1/3 & 1/2 & 1 & 0 \end{bmatrix}$$

核心困难：大型矩阵计算



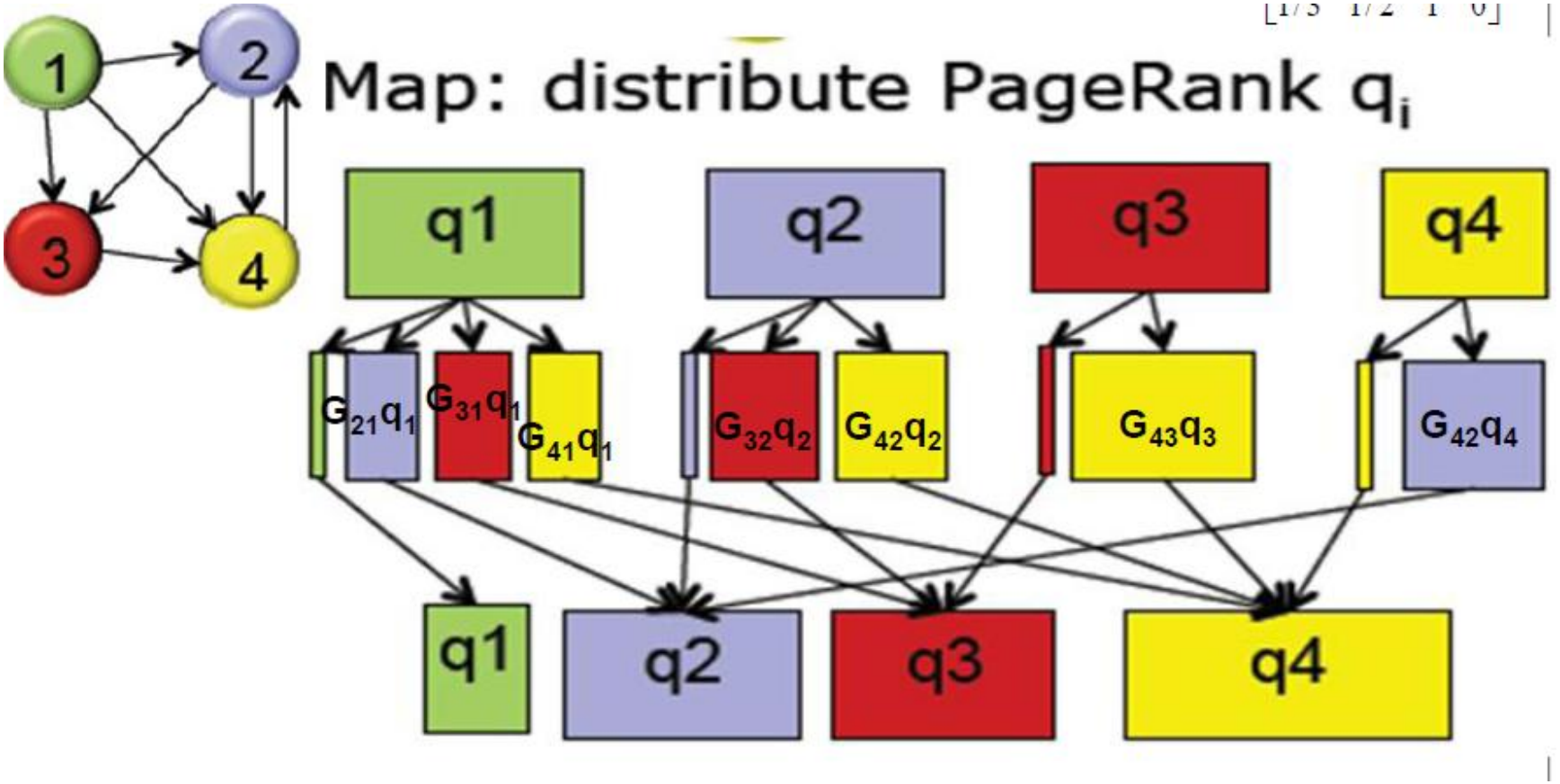
PageRank Map()

- Input:
  - key = page x,
  - value = (q_x, links[$y_1 \ldots y_m$])
- Output:
  - key = page x,
  - value = $partial_x$

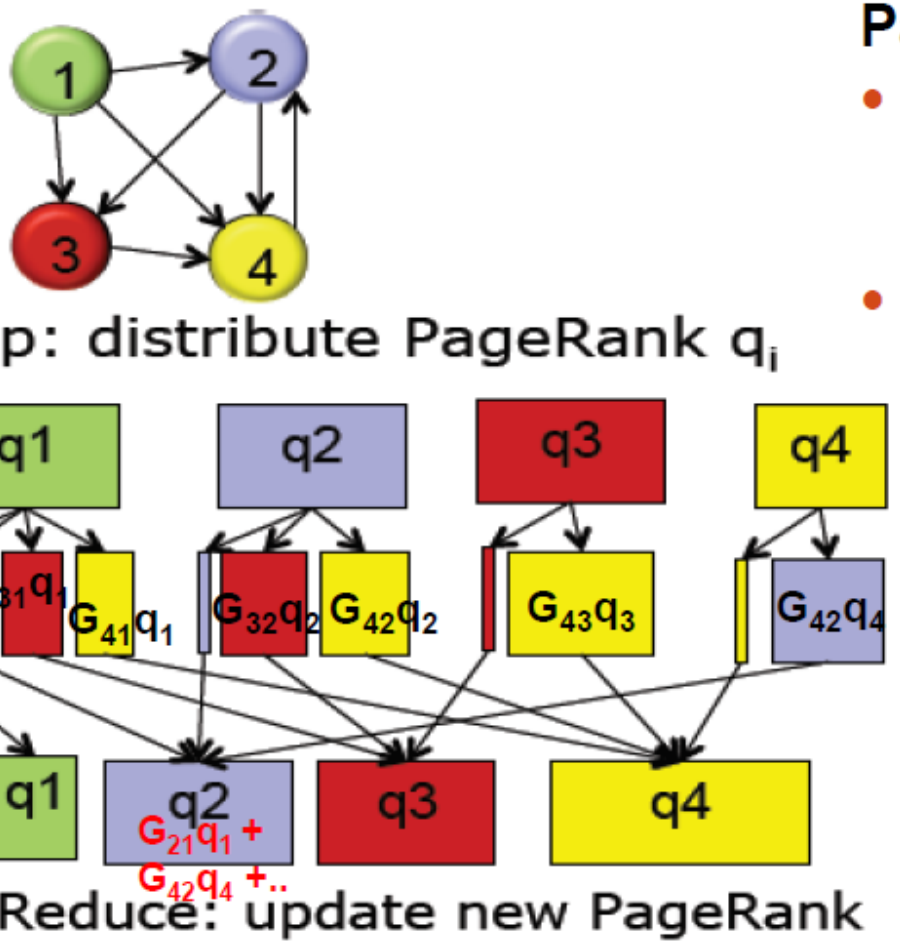
  1. Emit(x, 0) //guarantee all pages will be emitte
  2. For each outgoing link y_i:

     Emit(y_i, $G_{ix}q_x$)

使用Map-Reduce计算PR值



计算PR值



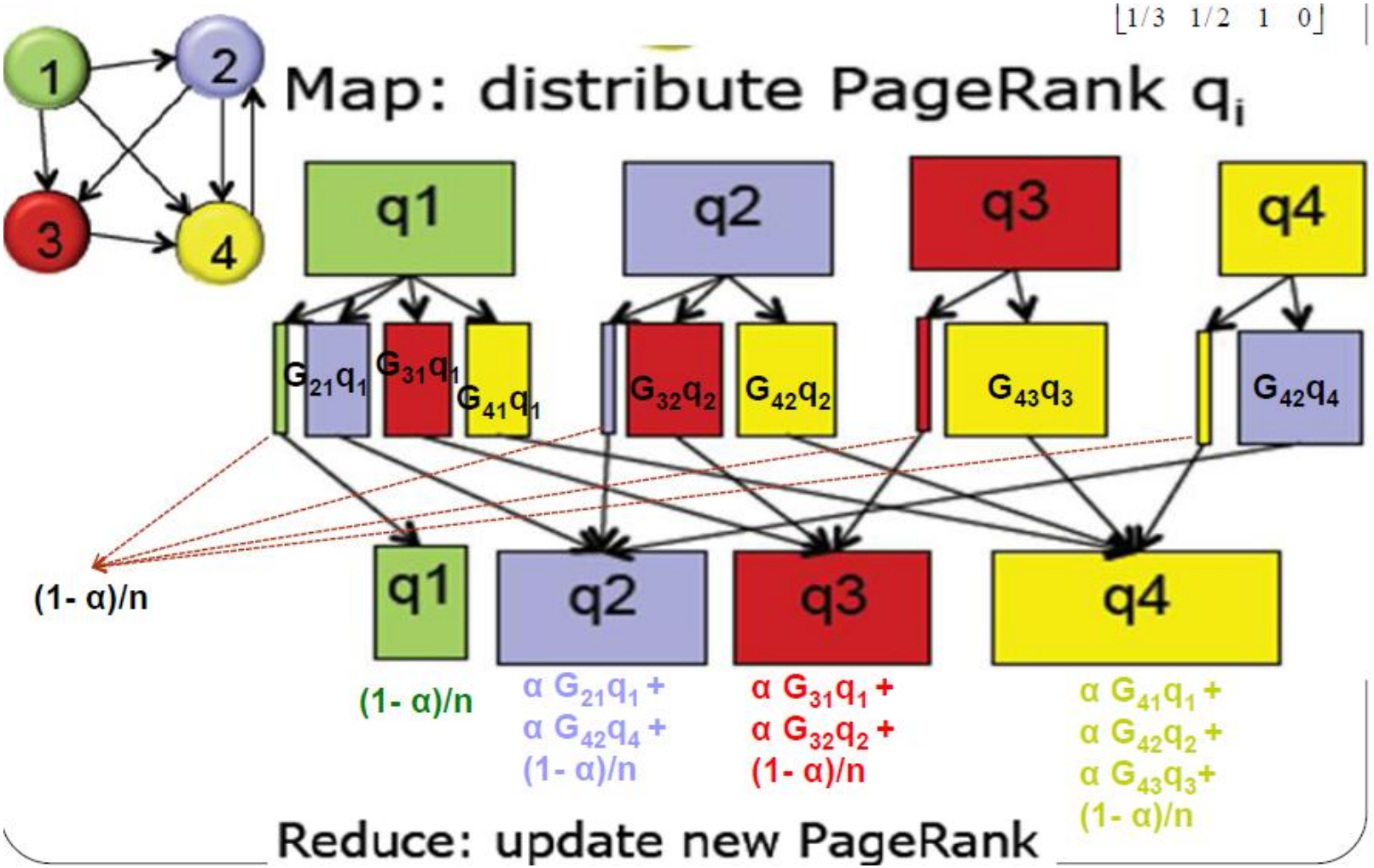
PageRank Reduce()

- Input:
  - key = page x,
  - value = the list of [$partial_x$]
- Output:
  - key = page x,
  - value = PageRank q_x

1. $q_x = 0$
2. For each partial value d in the list:
   q_x += d
3. $q_x = \alpha\, q_x + (1-\alpha)/n$
4. Emit(x, q_x)

$$q^{next} = Gq = \alpha Sq + (1-\alpha)\frac{1}{n}Uq$$

计算PR值


[1/3 1/2 1 0]
1
2
3
4
Map: distribute PageRank q_i
q1
q2
q3
q4
$G_{21}q_1$
$G_{31}q_1$
$G_{41}q_1$
$G_{32}q_2$
$G_{42}q_2$
$G_{43}q_3$
$G_{42}q_4$
(1- α)/n
q1
q2
q3
q4
(1- α)/n
α $G_{21}q_1$ +
α $G_{42}q_4$ +
(1- α)/n
α $G_{31}q_1$ +
α $G_{32}q_2$ +
(1- α)/n
α $G_{41}q_1$ +
α $G_{42}q_2$ +
α $G_{43}q_3$+
(1- α)/n
Reduce: update new PageRank

Gibbs抽样

- 《LDA数学八卦》第25页
- [0,1]内均匀随机数产生
- 正态分布随机数产生：Box-Muller变换
- 高维分布随机数产生比较困难，解决：Gibbs抽样
- Gibbs抽样基于MCMC（马尔科夫链蒙特卡罗方法）
- 抽样三阶段：initialization，burn-in，sampling
- 为了得到近似独立的采样，也可以在采样阶段设置每隔L次迭代采样一次

Gibbs抽样算法

Algorithm 7 二维Gibbs Sampling 算法

1: 随机初始化$X_0 = x_0 Y_0 = y_0$

2: 对$t = 0, 1, 2, \cdots$ 循环采样

1. $y_{t+1} \sim p(y|x_t)$
2. $x_{t+1} \sim p(x|y_{t+1})$

Algorithm 8 n维Gibbs Sampling 算法

1: 随机初始化$\{x_i : i = 1, \cdots, n\}$

2: 对$t = 0, 1, 2, \cdots$ 循环采样

1. $x_1^{(t+1)} \sim p(x_1|x_2^{(t)}, x_3^{(t)}, \cdots, x_n^{(t)})$
2. $x_2^{(t+1)} \sim p(x_2|x_1^{(t+1)}, x_3^{(t)}, \cdots, x_n^{(t)})$
3. $\cdots$
4. $x_j^{(t+1)} \sim p(x_j|x_1^{(t+1)}, \cdots, x_{j-1}^{(t+1)}, x_{j+1}^{(t)}, \cdots, x_n^{(t)})$
5. $\cdots$
6. $x_n^{(t+1)} \sim p(x_n|x_1^{(t+1)}, x_2^t, \cdots, x_{n-1}^{(t+1)})$

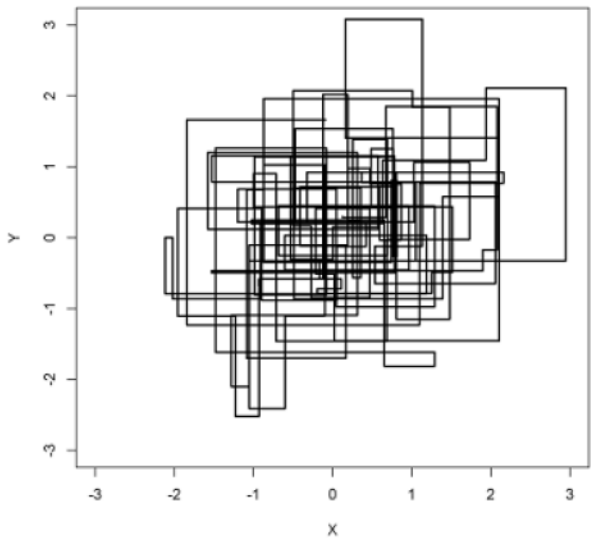
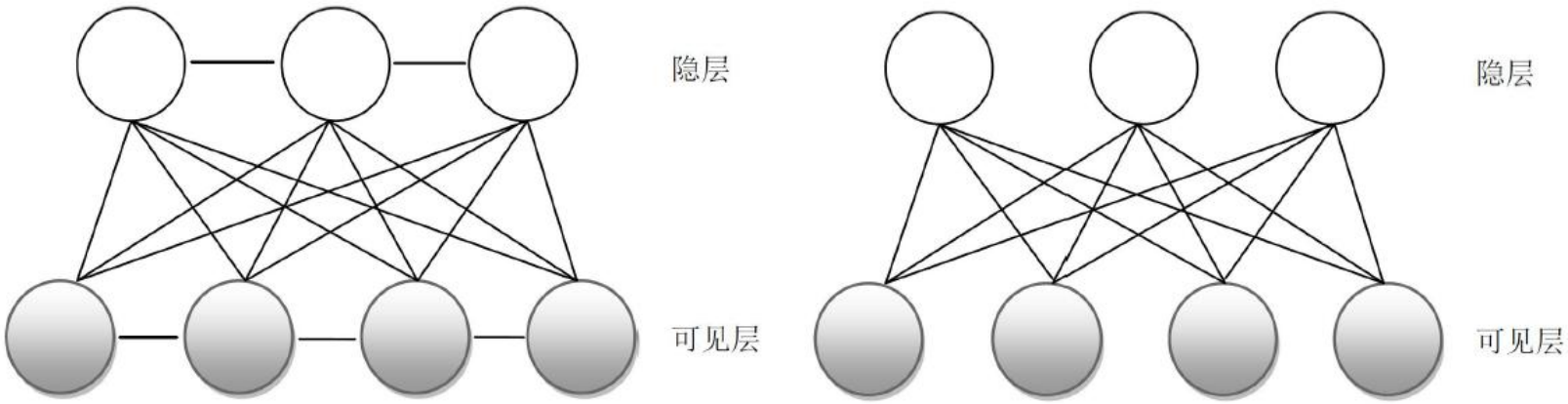


Figure 21: 二维Gibbs Sampling 算法中的马氏链转移

玻尔兹曼机的弱点

- 计算时间漫长，特别是无约束自由迭代的负向阶段
- 对抽样噪音敏感
- 很难找到合适的应用案例
- 流行软件不支持

受限玻尔兹曼机RBM

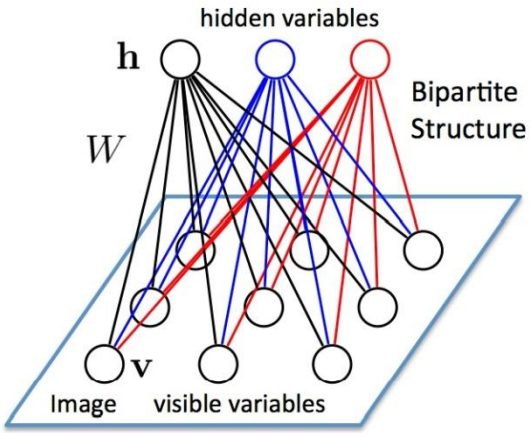


(a) BM模型的结构.　　(b) RBM模型的结构.

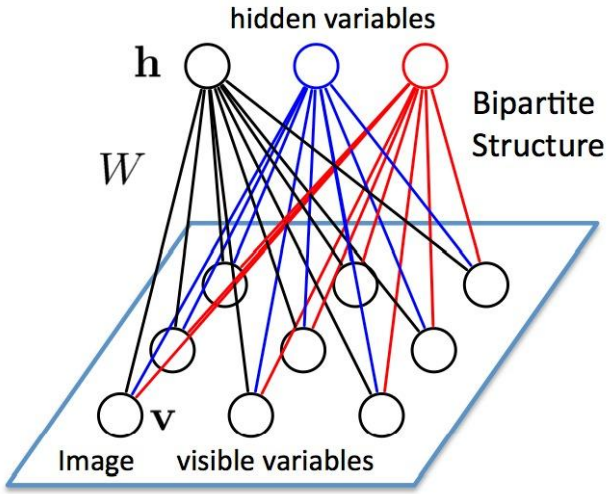
图 1: BM和RBM模型的结构比较.

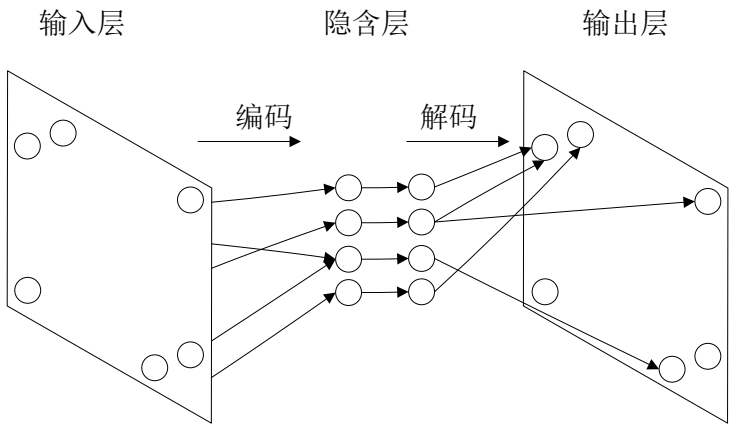
RBM

- 受限玻尔兹曼机(Restricted Boltzmann Machine,简称RBM)是由Hinton和Sejnowski于1986年提出的一种生成式随机神经网络(generative stochastic neural network)，该网络由一些可见单元(visible unit，对应可见变量，亦即数据样本)和一些隐藏单元(hidden unit，对应隐藏变量)构成，可见变量和隐藏变量都是二元变量，亦即其状态取{0,1}。整个网络是一个二部图，只有可见单元和隐藏单元之间才会存在边，可见单元之间以及隐藏单元之间都不会有边连接，如下图所示



对比与思考


hidden variables
h
W
Bipartite
Structure
v
Image
visible variables


输入层
隐含层
输出层
编码
解码

用途

- 分类、回归、降维、高维时间序列建模、图像特征提取、协同过滤
- 作为深度网络的一个基本部件，通过堆叠RBM成为深层，复杂的学习网络

能量函数

- RBM是一种基于能量(Energy-based)的模型，其可见变量v和隐藏变量h的联合配置(joint configuration)的能量为：

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{ij} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j$$

其中θ是RBM的参数{W, a, b}, W为可见单元和隐藏单元之间的边的权重，b和a分别为可见单元和隐藏单元的偏置(bias)。

联合概率

我们可以得到$(\mathbf{v}, \mathbf{h})$的联合概率分布,

$$P(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta}) = \frac{e^{-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})}}{Z(\boldsymbol{\theta})}, \quad Z(\boldsymbol{\theta}) = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})}$$

边际分布（似然函数）

对于一个实际问题, 我们最关心的是由RBM所定义的关于观测数据$\mathbf{v}$的分布$P(\mathbf{v}|\boldsymbol{\theta})$, 即联合概率分布$P(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})$的边际分布, 也称为似然函数(likelihood function),

$$P(\mathbf{v}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})}. \tag{3}$$

为了确定该分布, 需要计算归一化因子$Z(\boldsymbol{\theta})$, 这需要2^{n+m}次计算。因此, 即使通过训练可以得到模型的参数W_{ij}, a_i和b_j，我们仍旧无法有效地计算由这些参数所确定的分布。

最大似然法

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{t=1}^{T} \log P(\mathbf{v}^{(t)}|\boldsymbol{\theta}).$$

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &= \sum_{t=1}^{T} \log P(\mathbf{v}^{(t)}|\boldsymbol{\theta}) = \sum_{t=1}^{T} \log \sum_{\mathbf{h}} P(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta}) \\
&= \sum_{t=1}^{T} \log \frac{\sum_{\mathbf{h}} \exp[-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta})]}{\sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp[-E(\mathbf{v}, \mathbf{h})|\boldsymbol{\theta}]} \\
&= \sum_{t=1}^{T} \left(\log \sum_{\mathbf{h}} \exp[-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta})] - \log \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp[-E(\mathbf{v}, \mathbf{h})|\boldsymbol{\theta}] \right),
\end{aligned}$$

随机梯度上升法

令θ表示$\boldsymbol{\theta}$中的某一个参数, 则对数似然函数关于θ的梯度为

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \theta} &= \sum_{t=1}^{T} \frac{\partial}{\partial \theta} \left(\log \sum_{\mathbf{h}} \exp[-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta})] - \log \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp[-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})] \right) \\
&= \sum_{t=1}^{T} \left(\sum_{\mathbf{h}} \frac{\exp[-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta})]}{\sum_{\mathbf{h}} \exp[-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta})]} \times \frac{\partial(-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta}))}{\partial \theta} \right. \\
&\quad \left. - \sum_{\mathbf{v}} \sum_{\mathbf{h}} \frac{\exp[-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})]}{\sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp[-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})]} \times \frac{\partial(-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta}))}{\partial \theta} \right) \\
&= \sum_{t=1}^{T} \left(\left\langle \frac{\partial(-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta}))}{\partial \theta} \right\rangle_{P(\mathbf{h}|\mathbf{v}^{(t)}, \boldsymbol{\theta})} - \left\langle \frac{\partial(-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta}))}{\partial \theta} \right\rangle_{P(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})} \right)
\end{aligned} \tag{8}
$$

其中, $\langle \cdot \rangle_P$ 表示求关于分布 P 的数学期望。$P(\mathbf{h}|\mathbf{v}^{(t)}, \boldsymbol{\theta})$ 表示在可见单元限定为已知的训练样本 $\mathbf{v}^{(t)}$ 时, 隐层的概率分布, 故式(8)中的前一项比较容易计算。$P(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})$ 表示可见单元与隐单元的联合分布, 由于归一化因子 $Z(\boldsymbol{\theta})$ 的存在, 该分布很难获取, 导致我们无法直接计算式(8)中的第二项, 只能通过一些采样方法(如 Gibbs 采样)获取其近似值。值得指出的是, 在最大化似然函数的过程中, 为了加快计算速度, 上述偏导数在每一迭代步中的计算一般只基于部分而非所有的训练样本进行, 关于这部分内容我们将在后面讨论 RBM 的参数设置时详细阐述。

偏导数

$$\frac{\partial \log P(\mathbf{v}|\boldsymbol{\theta})}{\partial W_{ij}} = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}},$$
$$\frac{\partial \log P(\mathbf{v}|\boldsymbol{\theta})}{\partial a_i} = \langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{model}},$$
$$\frac{\partial \log P(\mathbf{v}|\boldsymbol{\theta})}{\partial b_j} = \langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{model}}.$$

CD算法

- Gibbs采样：从边际分布通过采样推测联合分布，需要较多采样
- Hinton提出基于对比散度的快速学习算法

2002年, Hinton [7]提出了RBM的一个快速学习算法, 即对比散度(Contrastive Divergence, CD)。与吉布斯采样不同, Hinton指出当使用训练数据初始化$\mathbf{v}_0$时, 我们仅需要使用k(通常$k=1$)步吉布斯采样便可以得到足够好的近似。在CD算法一开始, 可见单元的状态被设置成一个训练样本, 并利用式(4)计算所有隐层单元的二值状态。在所有隐层单元的状态确定之后, 根据式(5)来确定第i个可见单元v_i取值为1的概率, 进而产生可见层的一个重构(reconstruction)。这样, 在使用随机梯度上升法最大化对数似然函数在训练数据上的值时, 各参数的更新准则为

$$\Delta W_{ij} = \epsilon(\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{recon}}),$$
$$\Delta a_i = \epsilon(\langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{recon}}),$$
$$\Delta b_j = \epsilon(\langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{recon}}),$$

这里, ϵ是学习率(learning rate), $\langle \cdot \rangle_{\text{recon}}$表示一步重构后模型定义的分布。

伪代码

- 输入: 一个训练样本$\mathbf{x}_0$; 隐层单元个数m; 学习率ϵ; 最大训练周期T.
- 输出: 连接权重矩阵W、可见层的偏置向量$\mathbf{a}$、隐层的偏置向量$\mathbf{b}$.
- 训练阶段:

  初始化: 令可见层单元的初始状态$\mathbf{v}_1 = \mathbf{x}_0$; W、$\mathbf{a}$和$\mathbf{b}$为随机的较小数值。

  For $t = 1, 2, \cdots, T$

  For $j = 1, 2, \cdots, m$(对所有隐单元)

  计算$P(\mathbf{h}_{1j} = 1|\mathbf{v}_1)$, 即$P(\mathbf{h}_{1j} = 1|\mathbf{v}_1) = \sigma(b_j + \sum_i v_{1i}W_{ij})$;

  从条件分布$P(\mathbf{h}_{1j}|\mathbf{v}_1)$中抽取$\mathbf{h}_{1j} \in \{0, 1\}$.

  EndFor

  For $i = 1, 2, \cdots, n$(对所有可见单元)

  计算$P(\mathbf{v}_{2i} = 1|\mathbf{h}_1)$, 即$P(\mathbf{v}_{2i} = 1|\mathbf{h}_1) = \sigma(a_i + \sum_j W_{ij}h_{1j})$;

  从条件分布$P(\mathbf{v}_{2i}|\mathbf{h}_1)$中抽取$\mathbf{v}_{2i} \in \{0, 1\}$.

  EndFor

伪代码

For $j = 1, 2, \cdots, m$(对所有隐单元)

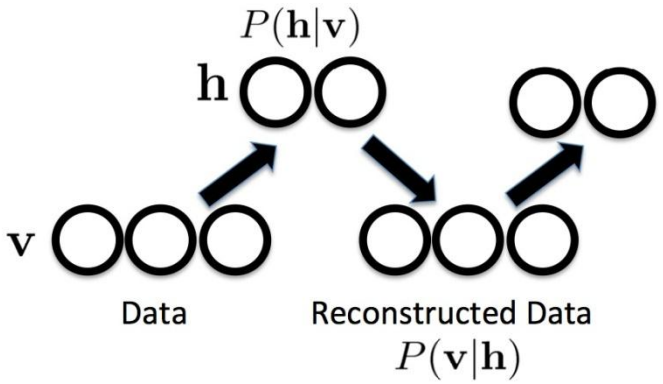
计算$P(\mathbf{h}_{2j} = 1|\mathbf{v}_2)$, 即$P(\mathbf{h}_{2j} = 1|\mathbf{v}_2) = \sigma(b_j + \sum_i v_{2i} W_{ij})$;

EndFor

按下式更新各个参数

- $W \leftarrow W + \epsilon(P(\mathbf{h}_{1.} = 1|\mathbf{v}_1)\mathbf{v}_1^T - P(\mathbf{h}_{2.} = 1|\mathbf{v}_2)\mathbf{v}_2^T)$;
- $\mathbf{a} \leftarrow \mathbf{a} + \epsilon(\mathbf{v}_1 - \mathbf{v}_2)$;
- $\mathbf{b} \leftarrow \mathbf{b} + \epsilon(P(\mathbf{h}_{1.} = 1|\mathbf{v}_1) - P(\mathbf{h}_{2.} = 1)|\mathbf{v}_2)$;

EndFor



炼数成金逆向收费式网络课程

- Dataguru（炼数成金）是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情，请看我们的培训网站 http://edu.dataguru.cn

Thanks

FAQ时间