第十五课——受限玻尔兹曼机RBM

马尔可夫性

- 马尔可夫，俄罗斯人，物理-数学博士，圣彼得堡科学院院士，彼得堡数学学派的代表人物，以数论和概率论方面的工作著称。

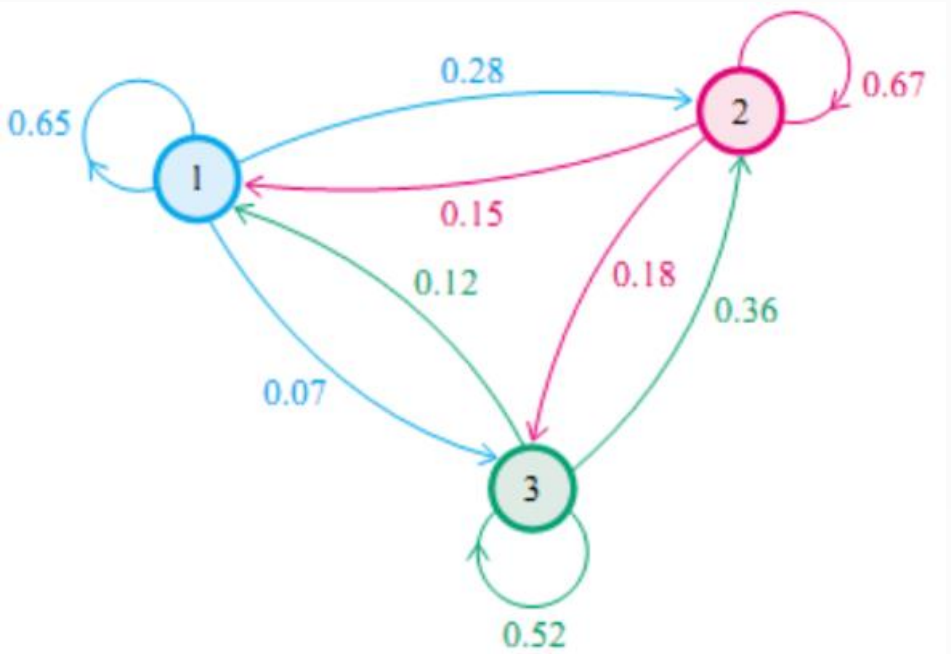
- 马尔可夫性：

当一个随机过程在给定当前的状态及所有过去状态情况下，未来状态的条件概率分布仅依赖于当前状态。

举例

社会学家经常把人按其经济情况分为3类：下层，中层，上层。我们用1,2,3来分别代表这3类。社会学家发现决定一个人收入阶层的最重要因素是其父母的收入阶层。从父代到子代，收入阶层的变化转移概率如下：

		子代		
	State	1	2	3
父代	1	0.65	0.28	0.07
	2	0.15	0.67	0.18
	3	0.12	0.36	0.52



状态转移矩阵

- 将来只依赖于现在不依赖过去的过程，我们称之为马尔可夫过程。时间和状态都是离散的马尔可夫过程称为马尔可夫链。

刚才例子中的状态转移矩阵为：

$$P = \begin{bmatrix} 0.65 & 0.28 & 0.07 \\ 0.15 & 0.67 & 0.18 \\ 0.12 & 0.36 & 0.52 \end{bmatrix}$$

平稳分布

假设初始的概率分布情况为π_0 = [0.210,0.680,0.110]，他们的子女的分布比例是$\pi_1 = \pi_0 P$，他们的孙子代的分布比例是$\pi_2 = \pi_0 P^2$。以此类推，第n代子孙的收入分布比例是$\pi_n = \pi_0 P^n$。

第n代人	下层	中层	上层
0	0.210	0.680	0.110
1	0.252	0.554	0.194
2	0.270	0.512	0.218
3	0.278	0.497	0.225
4	0.282	0.490	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	0.489	0.225
9	0.286	0.489	0.225
10	0.286	0.489	0.225

$$P = \begin{bmatrix} 0.65 & 0.28 & 0.07 \\ 0.15 & 0.67 & 0.18 \\ 0.12 & 0.36 & 0.52 \end{bmatrix}$$

平稳分布

假设初始的概率分布情况为π_0 = [0.750,0.150,0.100]，他们的子女的分布比例是$\pi_1 = \pi_0 P$，他们的孙子代的分布比例是$\pi_2 = \pi_0 P^2$。以此类推，第n代子孙的收入分布比例是$\pi_n = \pi_0 P^n$。

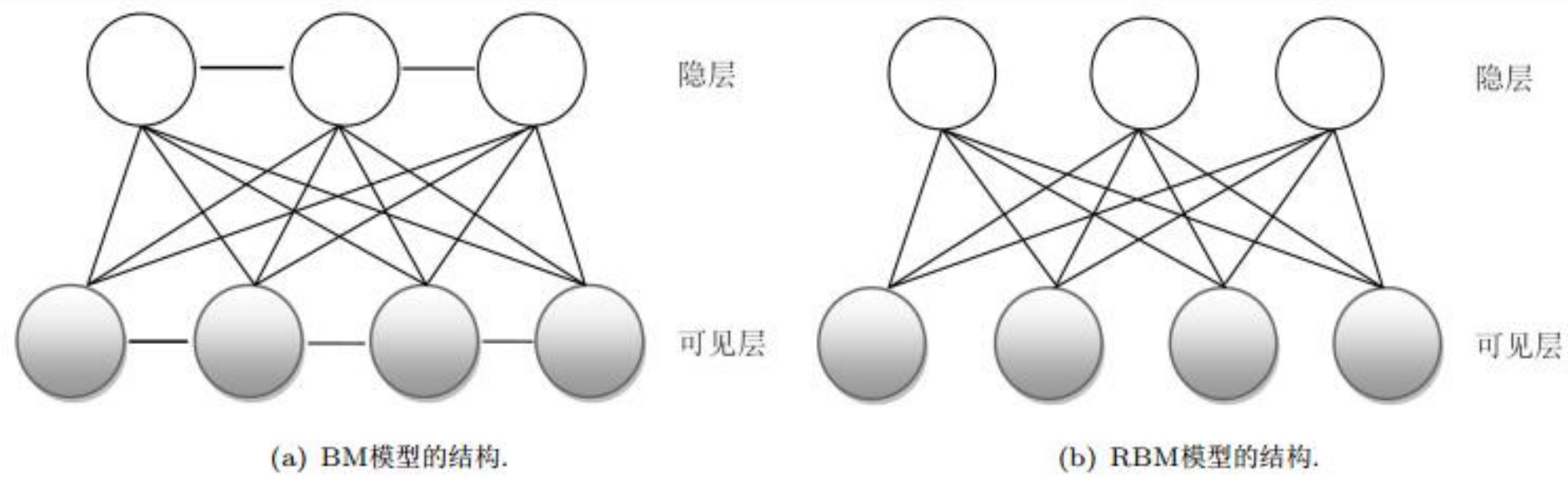
第n代人	下层	中层	上层
0	0.750	0.150	0.100
1	0.522	0.347	0.132
2	0.407	0.426	0.167
3	0.349	0.459	0.192
4	0.318	0.475	0.207
5	0.303	0.482	0.215
6	0.295	0.485	0.220
7	0.291	0.487	0.222
8	0.289	0.488	0.225
9	0.286	0.489	0.225
10	0.286	0.489	0.225

$$P = \begin{bmatrix} 0.65 & 0.28 & 0.07 \\ 0.15 & 0.67 & 0.18 \\ 0.12 & 0.36 & 0.52 \end{bmatrix}$$

应用

- 语言识别，自然语言处理
- 基因预测
- google的网页质量算法-PageRank

受限玻尔兹曼机



(a) BM模型的结构.　　(b) RBM模型的结构.

基于对比散度的RBM快速学习算法

θ ={W_{ij},a_i,b_j}
h隐层
v可见层
W权值
a可见层偏置
b隐层偏置
P概率
T:T个样本
data:P(h|$v^{(t)}$, θ)
model:P(v,h| θ)

对数似然函数:

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

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{t=1}^{T} \left(\left\langle \frac{\partial(-E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta}))}{\partial \theta} \right\rangle_{P(\mathbf{h}|\mathbf{v}^{(t)}, \boldsymbol{\theta})} - \left\langle \frac{\partial(-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta}))}{\partial \theta} \right\rangle_{P(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})} \right)$$

权值和偏置值

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

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

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

基于对比散度的RBM快速学习算法

$\theta = \{W_{ij}, a_i, b_j\}$
h隐层
v可见层
W权值
a可见层偏置
b隐层偏置
P概率
T:T个样本
data:$P(h|v^{(t)}, \theta)$
model:$P(v,h|\ \theta)$

基于对比散度的快速学习算法:

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

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

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

训练RBM

- **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

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

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

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

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

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

data:$P(h|v^{(t)}, \theta)$

微信公众号：深度学习与神经网络

QQ群：616043628

51CTO学院

Thank You !

为梦想增值！