电子科技大学
University of Electronic Science and Technology of China

Poisson Distribution Poisson Process
Poisson Factorization

Ruizhi Wu

Data Mining Lab, Big Data Research Center, UESTC
Email：nxznwurz@163.com

Abstract

This slide will briefly introduce poisson distribution , poisson process and poisson factorization by using some realistic example and simple math methods. Then we will introduce event model by poisson point process and recommend system by poisson factorization.

数据挖掘实验室
Data Mining Lab

Content

Poisson Distribution

- Two examples
- Poisson distribution

Poisson Process

- Poisson Process
- Event model

Poisson Factorization

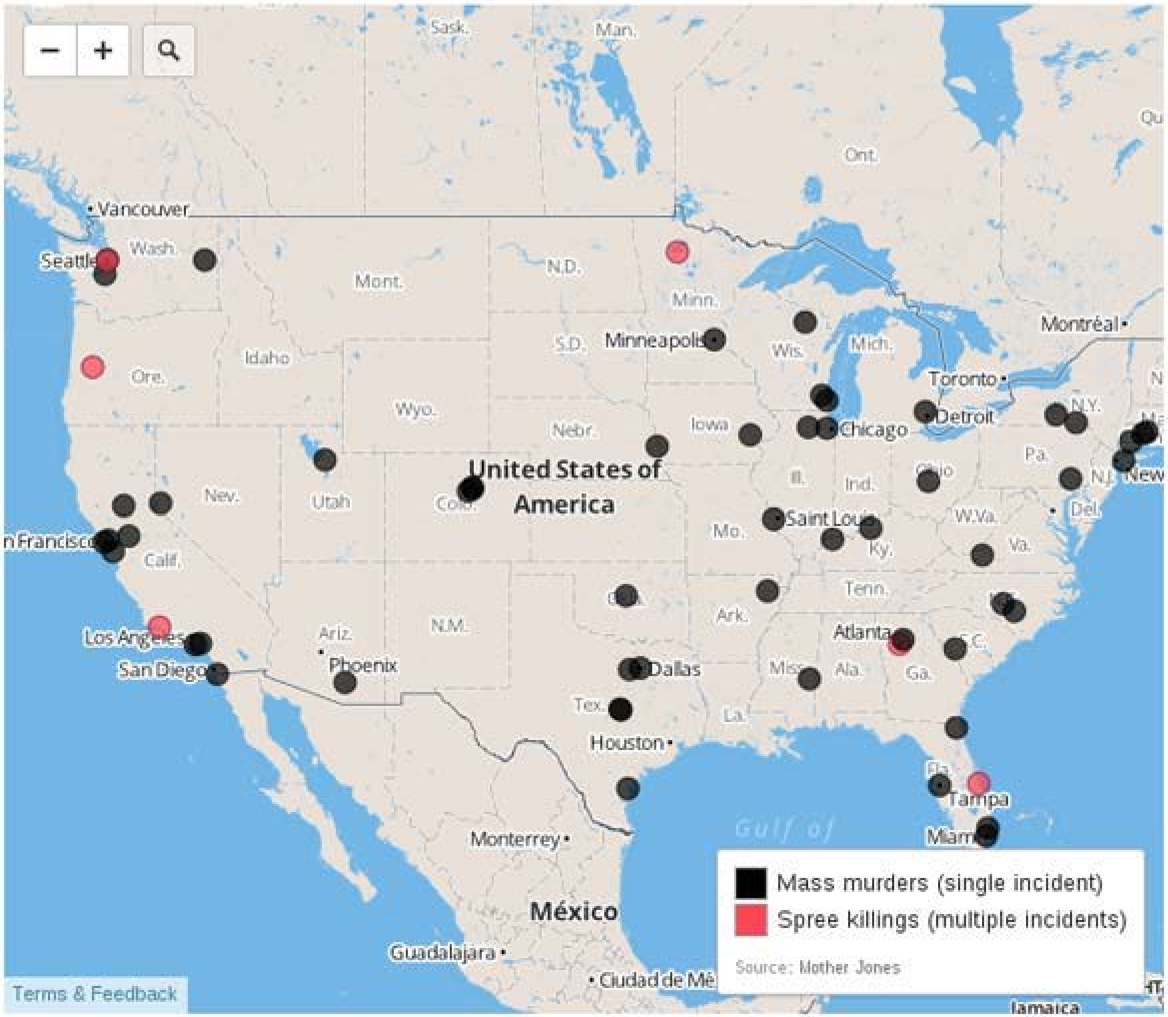
- Basic Model
- Extension Model

Poisson Distribution---Two examples

已知某家小杂货店，平均每周售出2个水果罐头。请问该店水果罐头的最佳库存量是多少？（暂时不考虑水果罐头的保质期）

Poisson Distribution---Two examples

数据挖掘实验室
Data Mining Lab



1982年至2012年，美国共发生62起（大规模）枪击案。其中，2012年发生了7起，是次数最多的一年。2012年有这么多枪击案，这是巧合，还是表明美国治安恶化了？

Poisson Distribution---Two examples

数据挖掘实验室
Data Mining Lab

以上的两个例子中，我们可以做出如下的近似假设：

- 发生枪击案和顾客购买水果罐头都是小概率事件；
- 发生枪击案或者购买水果罐头的顾客，各个事件之间是独立的，不会互相影响；
- 发生枪击案件或顾客购买水果罐头的概率是相对稳定的。

在统计学中，满足上述三个条件的事件，服从泊松分布。

$$P(X = k) = \frac{e^{-\lambda}\lambda^{k}}{k!}$$

数据挖掘实验室
Data Mining Lab

Siméon Denis Poisson 1781~1840

- *Life is good for only two things:*

  to study mathematics and to teach it.

Poisson Distribution

数据挖掘实验室
Data Mining Lab

泊松分布：

$$P(X=k)=\frac{e^{-\lambda}\lambda^{k}}{k!}$$

P：随机变量X发生k次的概率；
X：随机变量；
k：X的取值（0，1，2，3...）；
λ：是单位时间(或单位面积)内随机事件的平均发生率。

泊松分布的期望和方差均为 λ。

泊松分布适合于描述单位时间（或空间）内随机事件发生的次数。如某一服务设施在一定时间内到达的人数，电话交换机接到呼叫的次数，汽车站台的候客人数，由于泊松分布天然的特性，适合对人类行为进行建模。

Poisson Distribution

$$P(X=k)=\frac{e^{-\lambda}\lambda^{k}}{k!}$$

罐头的例子中，$\lambda = 2$

每周销出的罐头数量（k）	概率（P）	累积概率
X = 0	0.135	0.135
X = 1	0.271	0.406
X = 2	0.271	0.677
X = 3	0.180	0.857
X = 4	0.090	0.947
X = 5	0.036	0.983
X ≥ 6	0.017	1.000

如果存货4个罐头，95%的概率不会缺货（平均每19周发生一次）；如果存货5个罐头，98%的概率不会缺货（平均59周发生一次）。

- 发生枪击案和顾客购买水果罐头都是小概率事件；
- 发生枪击案或者购买水果罐头的顾客，各个事件之间是独立的，不会互相影响；
- 发生枪击案件或顾客购买水果罐头的概率是相对稳定的。

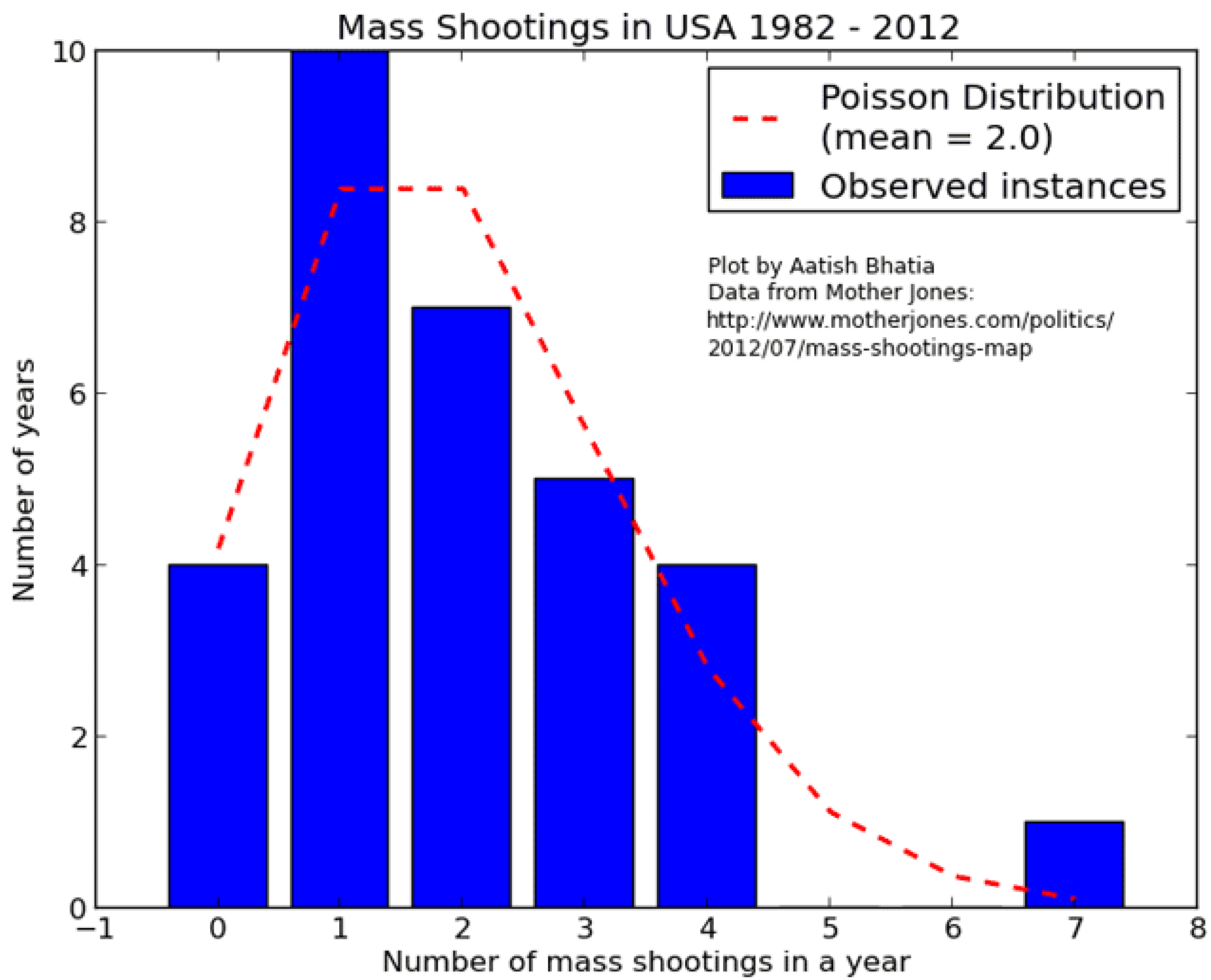
判断美国治安是否发生恶化，关键是判断枪击案的发生概率是否稳定，如果稳定，则治安没有恶化，如果枪击案的概率增大，则治安恶化严重。（既假设3是否成立。）

根据资料，1982--2012年枪击案的分布情况如下表所示：

一年中发生枪击案的数量	年数
0	4
1	10
2	7
3	5
4	4
5	0
6	0
7	1

平均每年发生2起枪击案，
所以 $\lambda = 2$ 。

Poisson Distribution



Poisson Distribution

$$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!}$$

λ = 2

一年中发生枪击案的数量	观察值	泊松分布期望值
0	4	4.2
1	10	8.39
2	7	8.39
3	5	5.59
4	4	2.8
5	0	1.12
6	0	0.37
7	1	0.11

卡方统计量 = Σ [(观察值 - 期望值) ^ 2 / 期望值]

结论： 卡方统计量等于9.82。查表后得到，置信水平0.90、自由度7的卡方分布临界值为12.017。因此，卡方统计量小于临界值，这表明枪击案的观察值与期望值之间没有显著差异。可以接受"发生枪击案的概率是稳定的"假设，也就是说，从统计学上无法得到美国治安正在恶化的结论。

- **Brief summary**

Poisson Distribution

泊松分布的概率函数为：

$$P(\mathrm{X}=\mathrm{k})=\frac{e^{-\lambda}\lambda^{k}}{k!},k=0,1,2...$$

特征函数：

$$\psi(t)=\exp(\lambda(e^{it}-1))$$

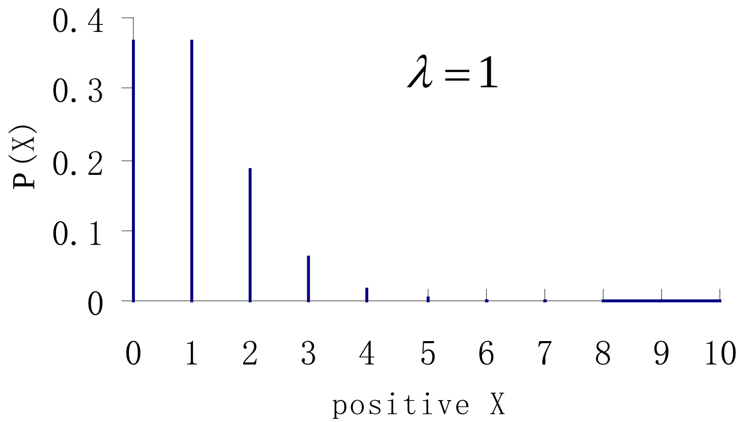
（在概率论中，任何随机变量的特征函数完全定义了它的概率分布。）

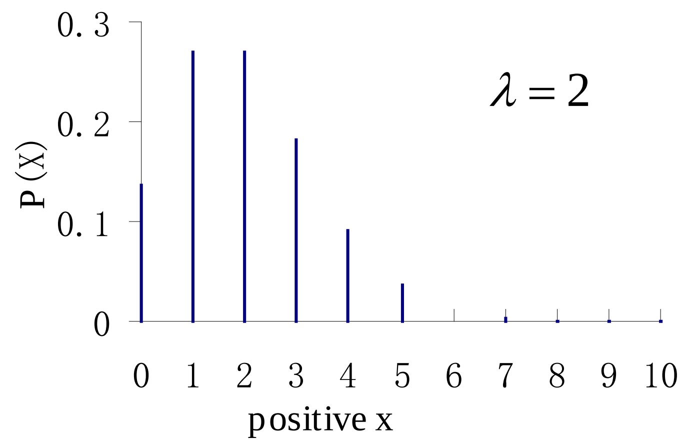
期望与方差：

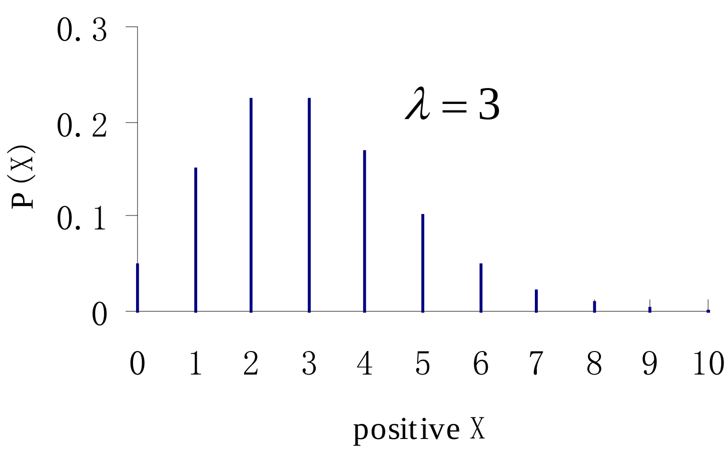
$$\lambda$$

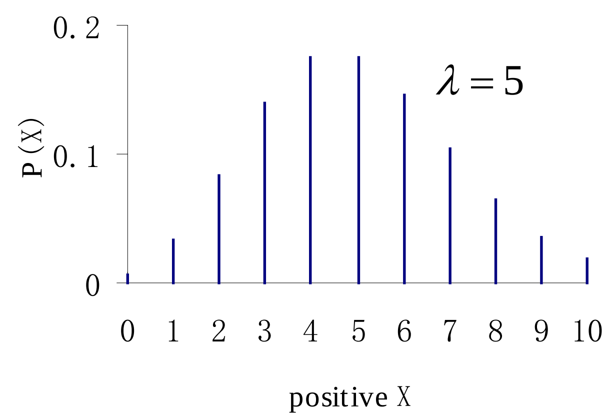
Poisson Distribution

不同λ:









泊松分布与二项分布：

当二项分布的n很大而p很小时，泊松分布可作为二项分布的近似，其中λ为np。事实上，泊松分布正是由二项分布推导而来的。具体推到请向度娘打听，此处不多说~

假设：

- 事件发生的概率是相对稳定的。
- 各个事件之间是独立的，不会互相影响；

人类很多行为可以视作在一定单位时间内的一个事件，同样类型的事件发生的次数整体可以通过泊松过程进行刻画，因为在随机过程中，泊松过程能够很好的刻画计数过程。

$$P(\mathrm{X}=\mathrm{k})=\frac{e^{-\lambda}\lambda^{k}}{k!},k=0,1,2\ldots$$

通过泊松过程刻画事件主要是刻画单位时间或单位空间内，时间发生的强度或次数。分析事件各种因素与发生次数 λ 的关系，构建表达式。

$$\lambda$$

上面构建的表达式通常称作Intensity function。

数据挖掘实验室
Data Mining Lab

假设：

- $\boxed{\text{事件发生的概率是相对稳定的。}}$
- 各个事件之间是独立的，不会互相影响；

假设1确定了λ是一个常量。在早期的研究中，认为人类活动是均匀的。2005年Nature一篇文章直指人类活动是非均匀的。

$$\lambda \begin{cases} time \\ location \\ item \\ \ldots \\ preference \\ social \end{cases}$$

- 时间维度是恒定且无限延长的

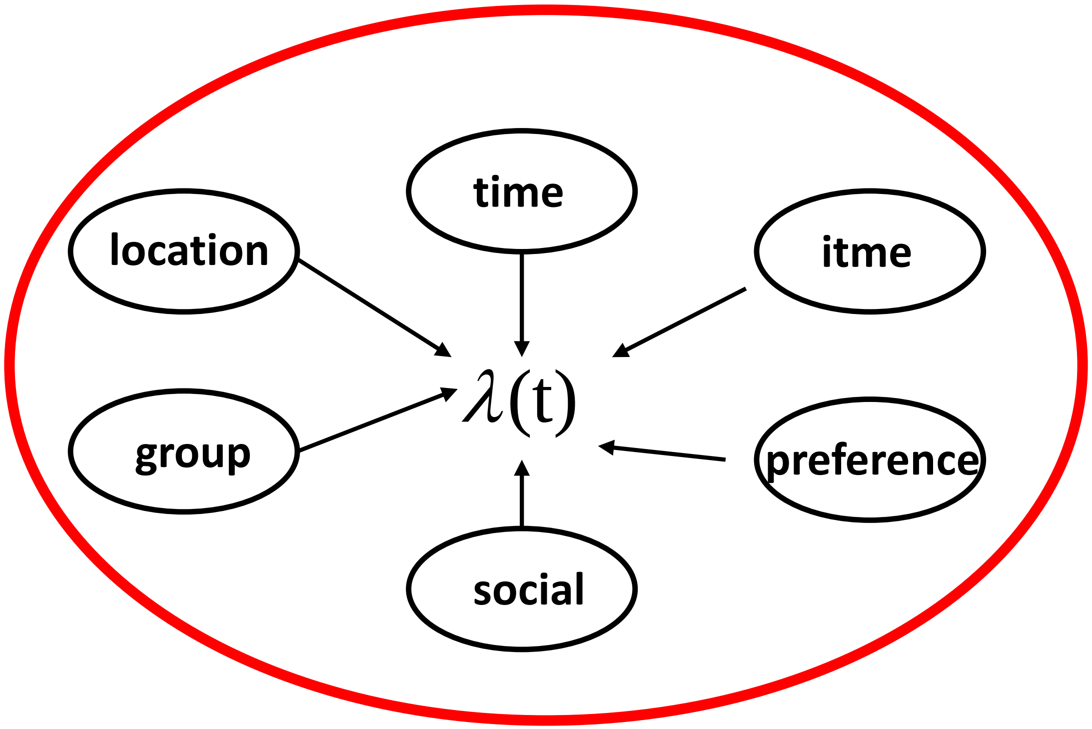
Intensity function：$\lambda(t)$

- 构建单位时间内的概率密度。

假设：

- 事件发生的概率是相对稳定的。
- $\boxed{\text{各个事件之间是独立的，不会互相影响；}}$
- 人类活动既然随着时间、偏好、社交以及其他影响，事件之间必然会产生影响，所以构建事件之间的影响也是建模需要考虑的因素。

Model

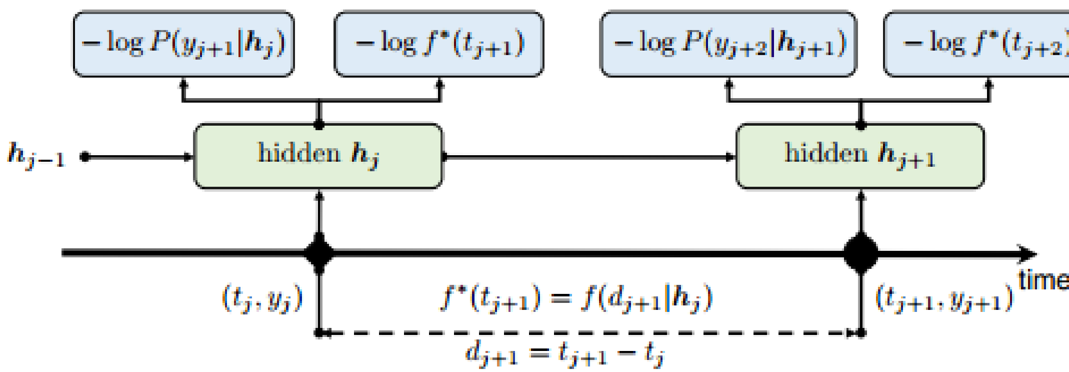


Recurrent Influence

预测：

$$f(t) = \lambda(t)exp(-\int_0^t \lambda(\tau)\,d\tau)$$

Poisson Process



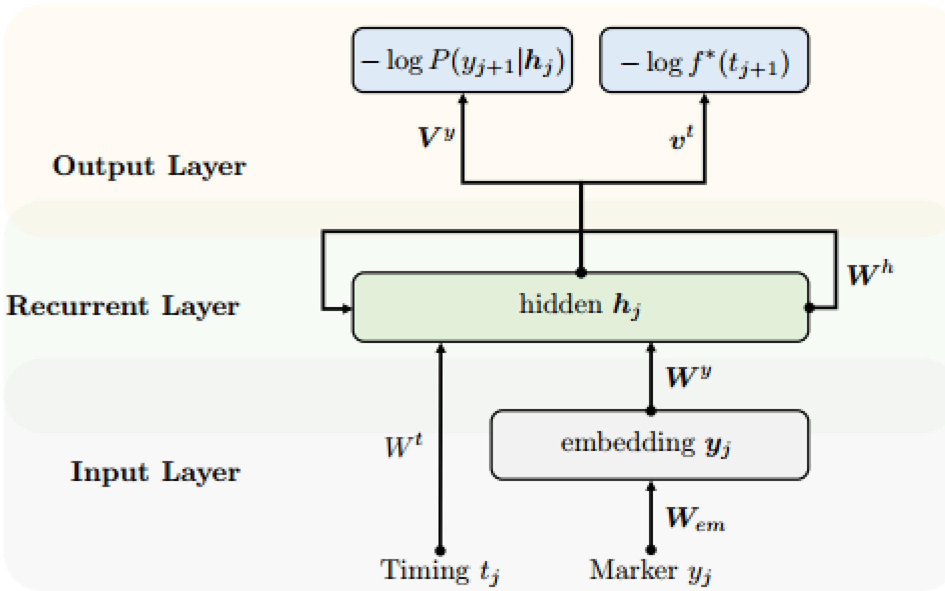
$$\lambda^*(t) = \exp\Big(\underbrace{\boldsymbol{v}^{t\top} \cdot \boldsymbol{h}_j}_{\substack{\text{past}\\\text{influence}}} + \underbrace{w^t(t - t_j)}_{\substack{\text{current}\\\text{influence}}} + \underbrace{b^t}_{\substack{\text{base}\\\text{intensity}}}\Big)$$

过去的影响，偏好，时间等与发生频率是一个黑箱模型。直接在RNN中进行学习，过去的影响用这种循环神经网络进行学习。

Recurrent Marked Temporal Point Processes: Embedding Event History to Vector[C]. 2016. KDD

Poisson Process

数据挖掘实验室
Data Mining Lab



$$h_j = \max\{W^y y_j + W^t t_j + W^h h_{j-1} + b_h, 0\}$$

$$P(y_{j+1} = k|\boldsymbol{h}_j) = \frac{\exp\left(\boldsymbol{V}^y_{k,:}\boldsymbol{h}_j + b^y_k\right)}{\sum_{k=1}^{K}\exp\left(\boldsymbol{V}^y_{k,:}\boldsymbol{h}_j + b^y_k\right)},$$

Recurrent Marked Temporal Point Processes: Embedding Event History to Vector[C]. 2016. KDD

Poisson Process

$$r_{up} \sim \textbf{Poisson}(\theta_u^\top \beta_p)$$

$$\lambda_{up}(t) = \theta_u^\top \beta_p + \sum_{e \in H(t)} \kappa_{up}(e,t)$$

$$\kappa_{up}(e,t) = e^{-\omega(t-t_e)} \mathcal{I}(p_e = p, u_e = u),$$

认为用户(user)，商品(product)是线性关系，单独加一个时间核函数，对过去事件的影响进行模拟。

Recurrent Poisson Factorization for Temporal Recommendation[C]. KDD. 2017.

Poisson Process

$$\lambda_u^*(t,l,c) = \lambda^*(t,c)\, f^*(l \mid t,c)$$

L代表location， t为时间， c为location的类别，下表u代表user， Intensity function 采用条件分布的方法进行构建，认为三元组(t,l,c)为联合分布的关系。

$$\lambda_u(t,c) = \mu_{uc} + \boxed{\sum_{i=1}^{|\mathcal{D}_{uc\cdot}(t)|} \beta_u \exp\left[-\frac{(t-t_i-k_i\tau)^2}{2\sigma^2}\right] \exp(-k_i)}$$ 周期性

$$f_u(\ell|c,t) = \underbrace{\sum_{l=1}^{L} \frac{w_{ucl}}{\eta_{uc}+w_{uc\cdot}} \delta_{\phi_l}(\ell)}_{\text{exploitation}} + \underbrace{\frac{\eta_{uc}}{\eta_{uc}+w_{uc\cdot}} G_0(\ell)}_{\text{exploration}}$$

Spatio-Temporal Modeling of Check-ins in Location-Based Social Networks[J]. 2016.

- **Brief summary**

Poisson Process

Motivation： 分析事件发生次数的影响因素，构建基于时间的Intensity function。建立非均匀、有记忆性的泊松过程，学术名次Marked Point Process。

Model：

- 分析事件发生次数的影响因素；
- 时间记忆性，既过去事件发生对未来时间产生的影响。

方法：

- 各种因素的黑箱模型，采用循环神经网络产生记忆效应；
- 采用分解的方法，单独加核函数构成时间上影响；
- 采用联合分布的方法，在分布中加周期函数，构成记忆效应。

SVD是一种矩阵分解方法，从本质上说是一种特殊的PMF。

评分矩阵: $$R_{U\times I} = \mathrm{U}_{U\times K} V_{K\times I}$$

$$\widehat{r_{ui}} = U_u^T V_i$$

损失: $$e_{ui} = r_{ui} - \widehat{r_{ui}}$$

$$SSE = \frac{1}{2}\sum_{u,i} e_{ui}^2 + \frac{1}{2}|u_u|^2 + \frac{1}{2}|v_i|^2$$

$$= \boxed{\frac{1}{2}\sum_{u,i}\left(r_{ui} - \sum_{k=1}^{K} u_{uk}v_{ki}\right)^2} + \boxed{\frac{1}{2}\lambda\sum_{u,i}\sum_{k=1}^{K} u_{uk}^2 + \frac{1}{2}\lambda\sum_{u,i}\sum_{k=1}^{K} v_{ki}^2}$$

数据挖掘实验室
Data Mining Lab

用贝叶斯的观点看，***R*是观测到的值，*U,V*描述了系统的内部特征**，是需要估计的。

- 对一个近似矩阵进行分解$\hat{\mathrm{R}} = U^T V$
- 要求近似矩阵$\hat{R}$在观测到的评分部分和观测矩阵R尽量相似
- 为了防止过拟合，需要对*U,V*做某种形式的约束

PMF Model

- 观测噪声（观测评分矩阵R和近似评分矩阵$\hat{R}$之差）为高斯分布
- 用户属性U和电影属性V均为高斯分布

评分矩阵的条件概率的先验也服从高斯分布：

$$p(R|U,V,\delta^2) = \prod_{u=1}^{M}\prod_{i=1}^{N}[N(R_{u,i}|\mathrm{U}_u^T V_i,\delta_R^2)]^{I_{u,i}^R}$$

其中$\mathrm{I}_{u,i}^R$是指示函数，当用户u对物品v_i有评分时，为1，否则为0。

假设商品V与用户U的特征矩阵都服从高斯分布：

$$p(U|\sigma_u^2) = \prod_{u=1}^{M} N(U_u|0,\sigma_u^2 I) \qquad p\left(V\middle|\sigma_i^2\right) = \prod_{i=1}^{N} N(V_i|0,\sigma_i^2 I)$$

利用贝叶斯推导，可得到用户和物品的隐式特征的后验概率：

$$P(U,V\middle|R,\delta_R^2,\delta_u^2,\delta_v^2) = p(R|U,V,\delta^2) \times p(U|\sigma_u^2) \times p\left(V\middle|\sigma_i^2\right)$$

$$= \prod_{u=1}^{M}\prod_{i=1}^{N}[N(R_{u,i}|\mathrm{U}_u^T V_i,\delta_R^2)]^{I_{u,i}^R} \times \prod_{u=1}^{M} N(U_u|0,\sigma_u^2 I) \times \prod_{i=1}^{N} N(V_i|0,\sigma_i^2 I)$$

对上式取对数求极大，就可以得到在已知参数$\delta_{\mathrm{R}}^2,\delta_V^2,\delta_U^2$，和评分矩阵的前提下得到最有可能的U和V的隐式特征矩阵。

最大化后验概率：

$$\ln p(U,V|R) = \ln p(R|U,V) + \ln p(U) + \ln p(V)$$

高斯分布和其对数形式：

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp(-\frac{(x-\mu)^2}{2\delta^2}) \qquad \ln p(x) = -\ln(\sqrt{2\pi\sigma}) - \frac{(x-\mu)^2}{2\delta^2}$$

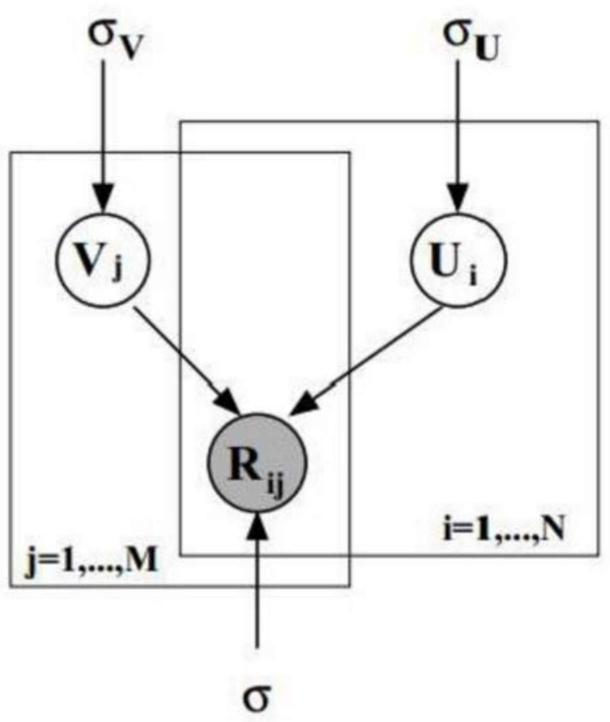
等同于优化：

$$E(U,V) = \frac{(R-U^TV)^2}{2\sigma^2} + \frac{U^TU^2}{2\sigma_U^2} + \frac{V^TV^2}{2\sigma_V^2}$$

最终优化目标：

$$E(U,V) = \frac{1}{2}\sum_{ij} I_{ij}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2}\sum_i U_i^T U_i^2 + \frac{\lambda_V}{2}\sum_j V_j^T V_j^2$$

Graph Model:



数据挖掘实验室
Data Mining Lab

Motivation:

- 不能保证推荐出的评分为非负的，这是由于假设高斯分布的先天缺陷(NMF已经解决)；
- 只对评分项进行拟合，把非评分项忽略，但PF会将用户没有评分的项也作为观测数据，隐式地把矩阵中的零作为用户资源的一个限制；
- PF只在用户观测矩阵中的已经观看过的数据中循环。这样可以利用用户产品矩阵的稀疏性，可以运用在大型的数据集上。

Hierarchical Poisson Factorization

The hierarchical Poisson factorization graph model:

1. For each user u:
   (a) Sample activity $\xi_u \sim \text{Gamma}(a', a'/b')$.
   (b) For each component k, sample preference

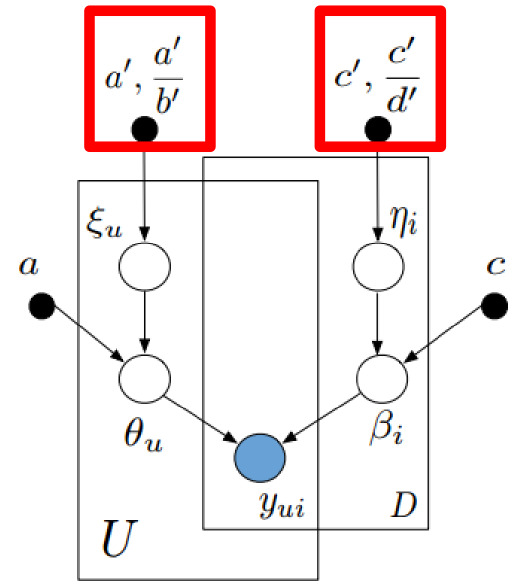
$$\theta_{uk} \sim \text{Gamma}(a, \xi_u).$$

2. For each item i:
   (a) Sample popularity $\eta_i \sim \text{Gamma}(c', c'/d')$.
   (b) For each component k, sample attribute

$$\beta_{ik} \sim \text{Gamma}(c, \eta_i).$$

3. For each user u and item i, sample rating

$$y_{ui} \sim \text{Poisson}(\theta_u^\top \beta_i).$$



Hierarchical Poisson Factorization

$X_{i,j} \sim Poisson\left(\theta_i^T \phi_j\right)$

泊松分布和其对数形式：

$$p(X_{i,j} \mid \theta_i^T, \phi_j) = (\theta_i^T \phi_j)^{X_{i,j}} \frac{\exp\{-\theta_i^T \phi_j\}}{X_{i,j}!}$$

$$\log p(y \mid \theta, \beta) = \sum_{\{y_{ui}>0\}} y_{ui} \log(\theta_u^\top \beta_i) - \log y_{ui}! - \left(\sum_u \theta_u\right)^\top \left(\sum_i \beta_i\right).$$

避免了其他方法所需要的复杂的子采样、近似、或随机优化的需要。

$$\text{score}_{ui} = \mathrm{E}[\theta_u^\top \beta_i \mid y]$$

求后验！！！

模型中的隐变量是θ_{uk}、β_{ik}、z_{ui}。各个参数独立。

$$q(\beta, \theta, \xi, \eta, z) = \prod_{i,k} q(\beta_{ik} \mid \lambda_{ik}) \prod_{u,k} q(\theta_{uk} \mid \gamma_{uk}) \prod_{u} q(\xi_u \mid \kappa_u) \prod_{i} q(\eta_i \mid \tau_i) \prod_{u,i} q(z_{ui} \mid \phi_{ui}).$$

数据挖掘实验室 Data Mining Lab

求后验！！！

For all users and items, initialize the user parameters γ_u, κ_u^{rte} and item parameters λ_i, τ_i^{rte} to the prior with a small random offset. Set the user activity and item popularity shape parameters:

$$\kappa_u^{\text{shp}} = a' + Ka; \quad \tau_i^{\text{shp}} = c' + Kc$$

Repeat until convergence:

1. For each user/item such that $y_{ui} > 0$, update the multinomial:

$$\phi_{ui} \propto \exp\{\Psi(\gamma_{uk}^{\text{shp}}) - \log \gamma_{uk}^{\text{rte}} + \Psi(\lambda_{ik}^{\text{shp}}) - \log \lambda_{ik}^{\text{rte}}\}.$$

2. For each user, update the user weight and activity parameters:

$$\gamma_{uk}^{\text{shp}} = a + \textstyle\sum_i y_{ui}\phi_{uik}$$

$$\gamma_{uk}^{\text{rte}} = \frac{\kappa_u^{\text{shp}}}{\kappa_u^{\text{rte}}} + \textstyle\sum_i \lambda_{ik}^{\text{shp}} / \lambda_{ik}^{\text{rte}}$$

$$\kappa_u^{\text{rte}} = \frac{a'}{b'} + \sum_k \frac{\gamma_{uk}^{\text{shp}}}{\gamma_{uk}^{\text{rte}}}$$

3. For each item, update the item weight and popularity parameters:

$$\lambda_{ik}^{\text{shp}} = c + \textstyle\sum_u y_{ui}\phi_{uik}$$

$$\lambda_{ik}^{\text{rte}} = \frac{\tau_i^{\text{shp}}}{\tau_i^{\text{rte}}} + \textstyle\sum_u \gamma_{uk}^{\text{shp}} / \gamma_{uk}^{\text{rte}}$$

$$\tau_i^{\text{rte}} = \frac{c'}{d'} + \sum_k \frac{\lambda_{ik}^{\text{shp}}}{\lambda_{ik}^{\text{rte}}}$$

Dynamic Poisson Factorization

1. Draw user global factors: $\bar{u}_{nk} \sim \mathcal{N}(\mu_{\bar{u}}, \sigma^2_{\bar{u}})$
2. Draw item global factors: $\bar{v}_{mk} \sim \mathcal{N}(\mu_{\bar{v}}, \sigma^2_{\bar{v}})$
3. For each time step: $t = 1 \ldots T$

   Draw user and item correction factors:

   if $t = 1$

   $$u_{nk,1} \sim \mathcal{N}(\mu_u, \sigma^2_u)$$
   $$v_{mk,1} \sim \mathcal{N}(\mu_v, \sigma^2_v)$$

   else

   $$u_{nk,t} | u_{nk,t-1} \sim \mathcal{N}(u_{nk,t-1}, \sigma^2_u)$$
   $$v_{mk,t} | v_{mk,t-1} \sim \mathcal{N}(v_{mk,t-1}, \sigma^2_v)$$

   Draw a click:

   $$y_{nm,t} \sim \text{Poisson}(\textstyle\sum_{k=1}^{K} e^{(u_{nk,t}+\bar{u}_{nk})} e^{(v_{mk,t}+\bar{v}_{mk})})$$

Poisson Trust Factorization

We now describe the generative process behind PTF. First, user preferences come from a Gamma distribution:

$$\theta_u \sim \text{Gamma}(a, b) \tag{1}$$

(Recall that the Gamma is an exponential family distribution on positive scalars.) Item characteristics also come from a Gamma:

$$\beta_i \sim \text{Gamma}(c, d) \tag{2}$$

For each set of connected users, the trust variable is generated in a similar way:

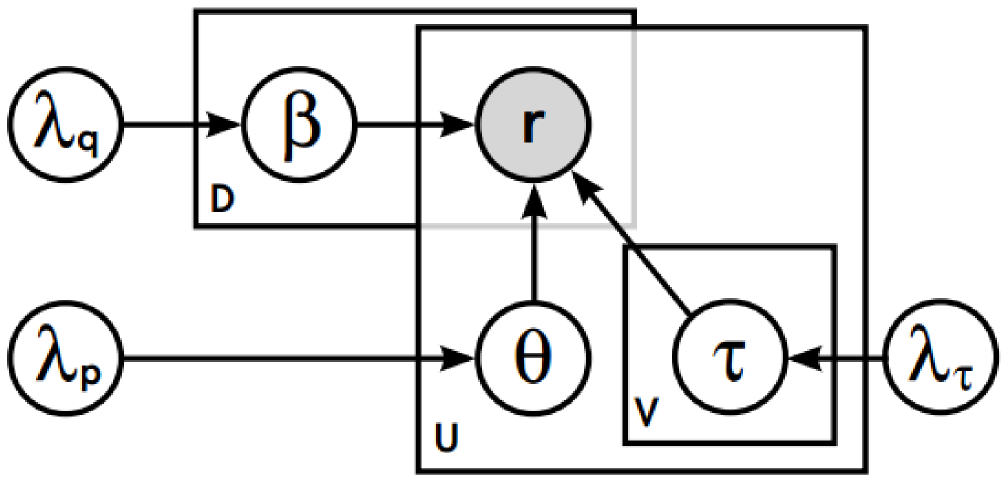
$$\tau_{uv} \sim \text{Gamma}(f, g) \tag{3}$$

Note this is a directed relationship. User u might trust user v, but not vice-versa.

Finally, ratings come from a combination of base matrix factorization and a trust component:

$$r_{ui} \sim \text{Poisson}\left(\theta_u^\top \beta_i + \sum_{v \in N(u,i)} \tau_{uv} r_{vi}\right) \tag{4}$$

Poisson Trust Factorization



- **Brief summary**

Poisson Factorization

Basic Model

- SVD
- Probabilistic Matrix Factorization
- Hierarchical Poisson Factorization

SVD PMF HPF

PMF是从概率生成过程的角度来解释user和item的latent factor，SVD是从优化目标出发，如何确定user和item的latent factor可以使得loss最少。两者解释的方式不一样, 但PMF的MLE 与 优化带正则项的svd的loss function是等价的。

- **Brief summary**

数据挖掘实验室
Data Mining Lab

SVD PMF HPF

HPF是为了解决PMF非负、只对观测数据拟合、和无法适应大规模数据的缺点提出的方法。

Extension Model

- Dynamic Poisson Factorization
- Poisson Trust Factorization

- **Summary**

Poisson distribution： 基本的泊松分布和两个简单的小例子。

Poisson process： 分析事件发生次数的影响因素，构建基于时间的Intensity function。建立非均匀、有记忆性的泊松过程，学术名次Marked Point Process。

Poisson factorization： 从SVD、PMF、HPF，最后拓展了两个模型。

• References

- Hosseini S A, Alizadeh K, Khodadadi A, et al. Recurrent Poisson Factorization for Temporal Recommendation[J]. 2017.
- Du N, Dai H, Trivedi R, et al. Recurrent Marked Temporal Point Processes: Embedding Event History to Vector[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016:1555-1564.
- Zarezade A, Jafarzadeh S, Rabiee H R. Spatio-Temporal Modeling of Check-ins in Location-Based Social Networks[J]. 2016.
- Probability and Statistics (4th Edition) by Morris H. DeGroot.
- 随机过程及应用
- http://maider.blog.sohu.com/304621504.html
- Salakhutdinov R, Mnih A. Probabilistic Matrix Factorization[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2007:1257-1264.
- **Gopalan P. Bayesian Nonparametric Poisson Factorization for Recommendation Systems[J]. 2014:275-283.**
- Gopalan P, Hofman J M, Blei D M. Scalable recommendation with hierarchical Poisson factorization[C]// Conference on Uncertainty in Artificial Intelligence. AUAI Press, 2015:326-335.
- Charlin L, Ranganath R, Mcinerney J, et al. Dynamic Poisson Factorization[C]// ACM Conference on Recommender Systems. ACM, 2015:155-162.

数据挖掘实验室
Data Mining Lab

Q & A