Y_i : # of events

CP - PP

$\{(s_n, i_n)\}$

$Y_i : \{s_i^1, \ldots s_i^{Y_i}\} \Rightarrow$ the set of events happened in $[0, T)$ for i-th entry

$U^1, U^2, \cdots, U^K$

$$p(Y_i) = p(Y_{i_1, \ldots i_K}) \sim \cancel{PP(\lambda_i)}\ PP(Y_i|\lambda_i) = e^{-\lambda_i \cdot T} \lambda_i^{Y_i}$$

$$\lambda_i \triangleq \exp(\langle U_i^1, \ldots, U_i^K \rangle)$$

U^k : $d_k \times r$ matrix

$$U_i^k \sim N(U_i^k | 0, I)$$

$$p(\{Y_i\}_i | \{U^1, \ldots, U^K\}) = \cancel{\prod_{k=1}^{K} \prod_{j=1} N(U_j^k | 0, I) \prod_{i \in \text{Observed}} p(Y_i|\lambda_i)}$$

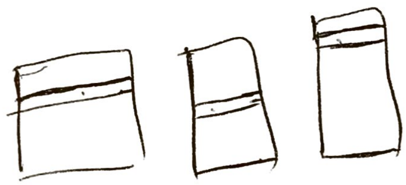
$$= \prod_{k=1}^{K} \prod_{j=1}^{d_K} N(U_j^k | 0, I) \prod_{i \in \text{Observed}} p(Y_i|\lambda_i)$$

$$= \prod_{k=1}^{K} \prod_{j=1}^{d_K} \prod_{i \in \text{Observed}} e^{-\lambda_i T} \lambda_i^{Y_i} \quad \Big(\lambda_i = \exp(\langle U_i^1, \ldots, U_i^K \rangle) \Big)$$

$$= \exp\Big(\sum_{j=1}^{r} \prod_{k=1}^{K} U_{i_k, j}^k\Big) \quad U_{i_k, j}^k$$

k-mode, i_k-th row, j-th column

$$\text{sum}(U_{i_1}^1 \circ U_{i_2}^2 \circ \cdots \circ U_{i_K}^K)$$



take derivative over $U_{i_k}^k$

$$d\lambda_i = \lambda_i\, d(U_{i_k}^k \cdot a^T) \qquad a = (U_{i_1}^1 \circ \cdots \circ U_{i_{k-1}}^{k-1} \circ U_{i_{k+1}}^{k+1} \cdots U_{i_K}^K)$$

↓ row vector

$$= \lambda_i\, a \cdot d(U_{i_k}^k)^T$$

for cal. we can compute $m_i = U_{i_1}^1 \circ \cdots \circ U_{i_K}^K$

$$\frac{d\lambda_i}{dU_{i_k}^k} = \lambda_i \cdot (m_i / U_i^k)$$

→ element-wise dividence

log(joint)

$$= \cancel{\sum_{k=1}^{K} \sum_{j=1}^{d_k}} -$$

$$= \sum_{k=1}^{K} -\frac{1}{2} \cancel{a}\, tr(U^{k^T} U^k)$$

assume observation size is N.

$$+ \sum_{i \in \text{Observed}} (-\lambda_i \cdot T + Y_i \log \lambda_i)$$

the batch size is B_i

$$= \sum_{k=1}^{K} -\frac{1}{2} \|U^k\|_F^2 + \sum_{j} \frac{N}{|B_j|} \sum_{i \in B_j} (-\lambda_i \cdot T + Y_i \log \lambda_i))$$

$j \downarrow$ mini-batch j

for any entry i, $\frac{d\lambda_i}{dU_{i_k}^k} = \lambda_i\left(m_i \,/\, U_i^k\right)$, $\frac{d \log \lambda_i}{dU_{i_k}^k} = m_i \,/\, U_i^k$

then we put gradient into the ∇U^k matrix

then we use SGD to train the model