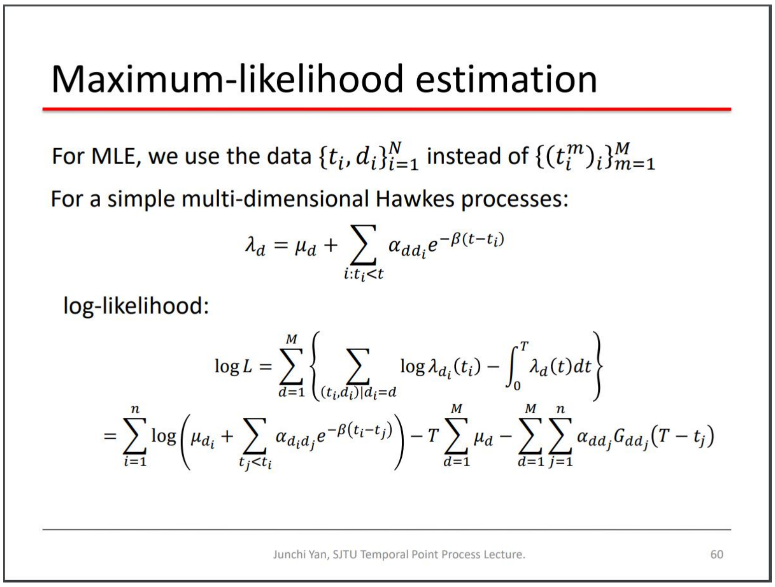
Report of fitting

Student ID: 516030910572
Name: Ding Yaoyao
This is the fitting report of the multi-hawkes point process. In last assignment, the simulation has been implemented. And in this assignment, EM algorithm is used to optimize the likelihood of the model.

EM algorithm

Here is the inductions in Prof. Yan's slides and I just copy them here for simplicity.



Maximum-likelihood estimation

For MLE, we use the data $\{t_i, d_i\}_{i=1}^{N}$ instead of $\{(t_i^m)_i\}_{m=1}^{M}$

For a simple multi-dimensional Hawkes processes:

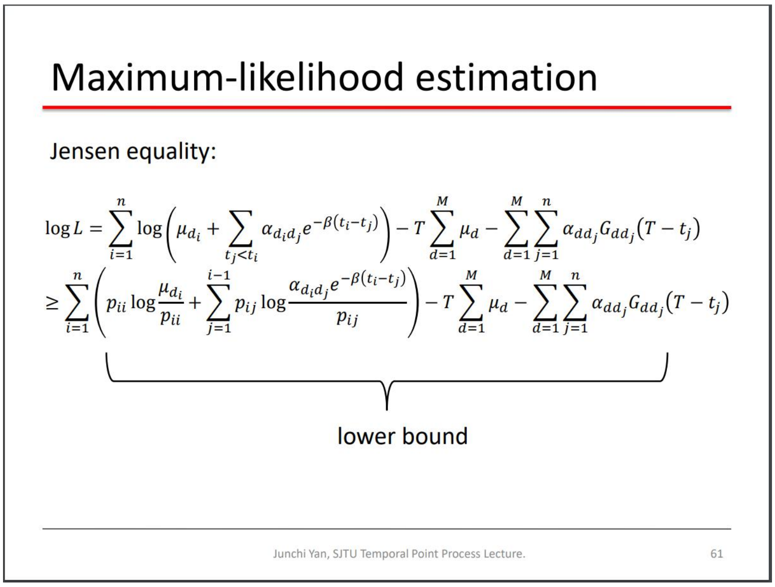
$$\lambda_d = \mu_d + \sum_{i:t_i<t} \alpha_{dd_i} e^{-\beta(t-t_i)}$$

log-likelihood:

$$\log L = \sum_{d=1}^{M} \left\{ \sum_{(t_i,d_i)|d_i=d} \log \lambda_{d_i}(t_i) - \int_0^T \lambda_d(t)dt \right\}$$

$$= \sum_{i=1}^{n} \log\left(\mu_{d_i} + \sum_{t_j<t_i} \alpha_{d_i d_j} e^{-\beta(t_i-t_j)} \right) - T\sum_{d=1}^{M} \mu_d - \sum_{d=1}^{M}\sum_{j=1}^{n} \alpha_{dd_j} G_{dd_j}(T-t_j)$$

Junchi Yan, SJTU Temporal Point Process Lecture. 60



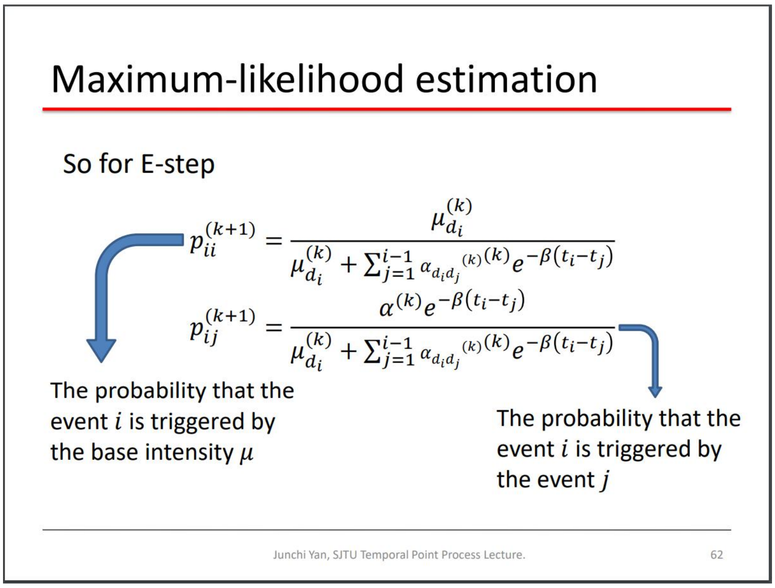
Maximum-likelihood estimation

Jensen equality:

$$\log L = \sum_{i=1}^{n} \log\left(\mu_{d_i} + \sum_{t_j<t_i} \alpha_{d_i d_j} e^{-\beta(t_i-t_j)} \right) - T\sum_{d=1}^{M} \mu_d - \sum_{d=1}^{M}\sum_{j=1}^{n} \alpha_{dd_j} G_{dd_j}(T-t_j)$$

$$\ge \underbrace{\sum_{i=1}^{n} \left(p_{ii} \log\frac{\mu_{d_i}}{p_{ii}} + \sum_{j=1}^{i-1} p_{ij} \log\frac{\alpha_{d_i d_j} e^{-\beta(t_i-t_j)}}{p_{ij}} \right) - T\sum_{d=1}^{M} \mu_d - \sum_{d=1}^{M}\sum_{j=1}^{n} \alpha_{dd_j} G_{dd_j}(T-t_j)}_{\text{lower bound}}$$

Junchi Yan, SJTU Temporal Point Process Lecture. 61



Maximum-likelihood estimation

So for E-step

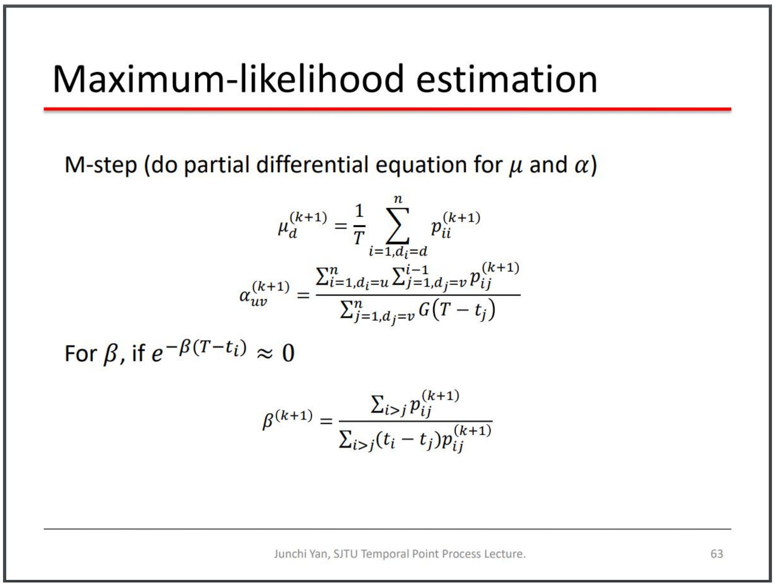
$$p_{ii}^{(k+1)} = \frac{\mu_{d_i}^{(k)}}{\mu_{d_i}^{(k)} + \sum_{j=1}^{i-1} {\alpha_{d_i d_j}}^{(k)(k)} e^{-\beta(t_i - t_j)}}$$

$$p_{ij}^{(k+1)} = \frac{\alpha^{(k)} e^{-\beta(t_i - t_j)}}{\mu_{d_i}^{(k)} + \sum_{j=1}^{i-1} {\alpha_{d_i d_j}}^{(k)(k)} e^{-\beta(t_i - t_j)}}$$

The probability that the event i is triggered by the base intensity μ

The probability that the event i is triggered by the event j

Junchi Yan, SJTU Temporal Point Process Lecture. 62



Maximum-likelihood estimation

M-step (do partial differential equation for μ and α)

$$\mu_d^{(k+1)} = \frac{1}{T} \sum_{i=1, d_i = d}^{n} p_{ii}^{(k+1)}$$

$$\alpha_{uv}^{(k+1)} = \frac{\sum_{i=1, d_i = u}^{n} \sum_{j=1, d_j = v}^{i-1} p_{ij}^{(k+1)}}{\sum_{j=1, d_j = v}^{n} G(T - t_j)}$$

For β, if $e^{-\beta(T - t_i)} \approx 0$

$$\beta^{(k+1)} = \frac{\sum_{i>j} p_{ij}^{(k+1)}}{\sum_{i>j} (t_i - t_j) p_{ij}^{(k+1)}}$$

Junchi Yan, SJTU Temporal Point Process Lecture. 63

Implementation details

Here is the procedures of the implementation:

1. Simulate 10 event sequences, each sequence has 1000 events
2. For each sequence, use EM algorithm to optimize the parameters of U, A, w. We can get 10 sets of parameters. In EM algorithm, I fixed the number of iteration to 20.
3. Average the 10 sets of parameters to get the final result
4. Evaluate the result and output

Usage

```
# cd fitting
# python main.py
```

You can find the fitting result fitted_parameters.json in result folder.

I have run the code and the mean relative error is: 0.168 (I have fixed the random seeds so you are supposed to get the same result after running the code)