Personalized Machine Learning

Temporal Dynamics and Popularity

Rodrigo Alves

November 27, 2025.

Tossing a Coin

- Tossing a coin is a very popular method worldwide for making binary decisions.
- What is the probability of getting a **head** and a **tail**?
- Let's say I have a fair coin, meaning that $P(\text{head}) = P(\text{tail}) = \frac{1}{2}$.
  - Now, if I were to toss the coin 100 times, and 70 of those times resulted in a **head** what would be the probability of getting a **tail** in the next toss?
- Perhaps you should question whether $P(\text{head}) = P(\text{tail}) = \frac{1}{2}$!

Tossing a Coin

- In the majority of cases, it's challenging to determine if a coin is truly fair.
- What's even more problematic is that it's probable that finding a genuinely fair coin is very difficult.
- Nevertheless, many coins have probabilities such that $P(\text{head}) \approx P(\text{tail}) \approx \frac{1}{2}$.
- Suppose I tell you that I have a fair coin, where $P(\text{head}) = P(\text{tail}) = \frac{1}{2}$.
  - Now, if I were to toss it 100 times, and 70 of those times resulted in **heads**
- Is it likely that $P(\text{head}) = P(\text{tail}) = \frac{1}{2}$?
- Can we estimate the **most likely** values of $P(\text{head})$ and $P(\text{tail})$?

Maximum Likelihood Estimation

To be more formal, let's assume the following:

- Y_i is a random variable representing the i-th toss.
  - $Y_i = 1$ indicates that the i-th toss resulted in a **head**.
  - $Y_i = 0$ indicates that the i-th toss resulted in a **tail**.
- Consequently, we have $P(Y_i = 0) + P(Y_i = 1) = 1$, which implies $P(Y_i = 0) = 1 - P(Y_i = 1)$.
- If we denote $P(Y_i = 1)$ as p, then $P(Y_i = 0) = 1 - p$.
- We observe a sequence of tosses: $Y_1, Y_2, \cdots, Y_{100}$, and among them, 70 are heads.

$$\begin{aligned} P(Y_1, Y_2, \cdots, Y_{100}) &= \prod_i p^{Y_i}(1-p)^{1-Y_i} \\ &= p^{\sum_i Y_i}(1-p)^{100-\sum_i Y_i} \\ &= p^{70}(1-p)^{30} \end{aligned}$$

Maximum Likelihood Estimation

- It's a fact: the event has occurred, with 70 heads and 30 tails.
- Now, we have $P(Y_1, Y_2, \cdots, Y_{100}) = p^{70}(1-p)^{30}$.
- The key question is: which value of p maximizes $P(Y_1, Y_2, \cdots, Y_{100})$?

In other words, can we compute the **most likely** value of p by considering that we observed 70 heads and 30 tails?

- Once again, we encounter an optimization problem:

$$\mathcal{L}(p) = p^{70}(1-p)^{30}$$

Some Logarithm Properties

- $\ln(\boldsymbol{a}\boldsymbol{b}) = \ln(\boldsymbol{a}) + \ln(\boldsymbol{b})$
- $\ln\left(\frac{\boldsymbol{a}}{\boldsymbol{b}}\right) = \ln(\boldsymbol{a}) - \ln(\boldsymbol{b})$
- $\ln(\boldsymbol{a}^{\boldsymbol{b}}) = \boldsymbol{b}\ln(\boldsymbol{a})$

Maximum Likelihood Estimation

- The function $\mathcal{L}(p) = p^{70}(1-p)^{30}$ is a real and continuous function.
- Therefore, we can simply compute its derivative and set it to zero.

$$\begin{aligned} \mathcal{L}(p) &= p^{70}(1-p)^{30} \\ \ln \mathcal{L}(p) &= \ln\left(p^{70}(1-p)^{30}\right) \\ &= \ln p^{70} + \ln(1-p)^{30} \\ &= 70 \ln p + 30 \ln(1-p) \end{aligned}$$

Maximum Likelihood Estimation

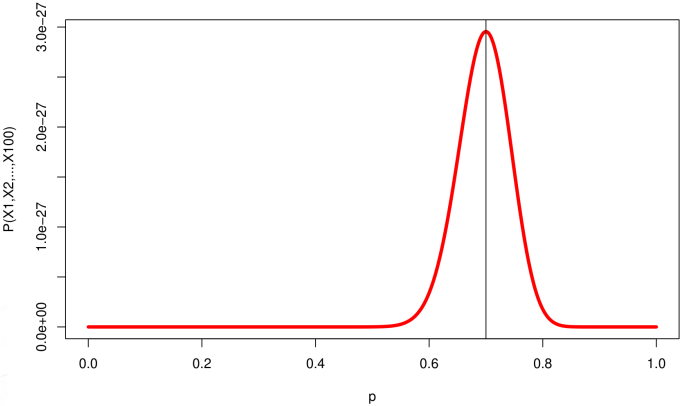
- Computing the derivative:

$$\begin{aligned}\ell(p) &= 70\ln p + 30\ln(1-p)\\ \ell'(p) &= \frac{70}{p} - \frac{30}{1-p}\end{aligned}$$

- By setting the derivative to 0, we get:

$$\begin{aligned}0 &= \frac{70}{p} - \frac{30}{1-p}\\ &= 70(1-p) - 30p = 70 - 70p - 30p\\ 100p &= 70\\ p &= 0.70\end{aligned}$$

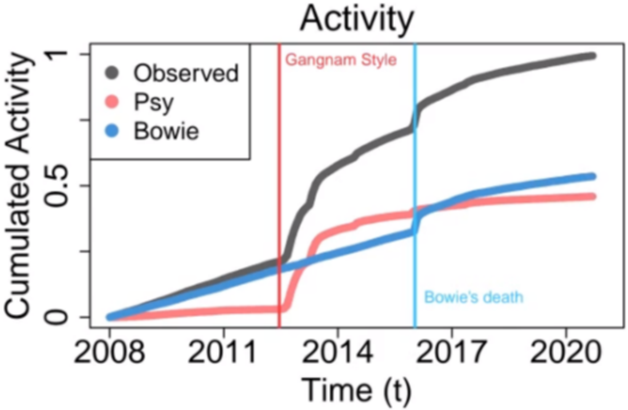
Maximum Likelihood Estimation



Popularity in Recommender Systems

- Popularity is a fundamental concept in recommender systems.
- It refers to the **relative frequency** of items among users.
- Popularity can be measured in various ways, such as the number/rate of views, ratings, or purchases.
- Understanding popularity is essential for building effective recommendation algorithms.

Popularity in Recommender Systems



Popularity in Recommender Systems

- Previous research shows that the popularity of online items varies between:
  - calm (or normal) periods;
  - and huge sequences of events (bursts of events).
- We, thus, model the popularity split into two audiences:
  - **Stable**: responsible for **calm** periods;
  - **Curious**: responsible for the **bursts**.

> "All models are **wrong**. Some are **useful**!"
>
> — George Box

Stable vs Curious

- Curiosity is **hard** to measure. Bursts of events are highly unpredictable:
  - when they will happen.
  - how big the number of events will be.
- Therefore, we are more interested in knowing the **rate of events** of the stable audience.
- And it is important for recommender systems to understand whether an item is in a **stable** or **curious** **mode**.

Point Processes

- Let's model our audiences through point processes.
- **Point Process:** It's a mathematical model for random events occurring over time (or space).
- Events are random and may be discrete or continuous.
- The **intensity function ($\lambda(t)$)** describes the event occurrence rate.
- Events can be dependent or independent.

Intensity Function

Definition

Consider a point process with n observed times $T = \{t_1, t_2, \ldots, t_n\}$. Let $N(t)$ represent the number of events in the point process up to time t. It is defined as:

$$N(t) = \sum_{i=1}^{n} 1_{(t_i \leq t \text{ and } t_i \in T)}$$

The **intensity function** $\lambda(t)$ reflects the **instantaneous event rate** at each point in time within the point process and is defined as:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{\mathbb{E}[N(t + \Delta t) - N(t)]}{\Delta t}$$

Poisson Process

A **Poisson process** is point process, characterized by the following properties:

- **Homogeneity**: The event rate is constant over time, denoted as $\lambda(t) = \lambda_P$.
- **Memorylessness**: The time until the next event follows an exponential distribution, which means it has no memory of the past.

The Poisson process is widely used in various fields to model random events, such as arrivals at a service center, radioactive decay, and more.

Poisson Process: Homogeneity

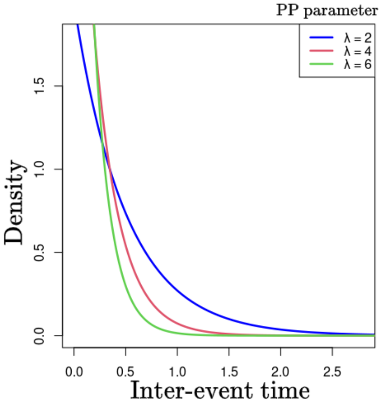
- In a Poisson Process, the rate of incoming events, denoted as $\lambda(t) = \lambda_P$, remains constant.
- This property makes the Poisson Process a suitable model for describing a stable audience.
- Specifically, this means that within each fixed unit of time (e.g., seconds, minutes, days), we expect $\mathbb{E}[N(t+1) - N(t)] = \lambda$.

Note: the constancy of λ does not imply regularity in the sense that events will occur precisely at regular intervals. For example, when $\lambda = 1$, in 10 unit times

- we may not necessarily observe the events at times $\{1, 2, \ldots, 10\}$;
- we only expect to have 10 events in total, but the actual number of events may vary.

Poisson Process: Memorylessness

The time until the next event follows an exponential distribution, which means it has no memory of the past.



Poisson Process: λ

- Now, let's consider the scenario where we have knowledge of the time stamps $T = \{t_1, t_2, \ldots, t_n\} \in (0, \mathcal{T}]$. How can we determine the value of λ?
- Perhaps a more appropriate question is: What is the most likely value for λ?

Likelihood of Point Processes Definition

When considering the history $\mathcal{H}$ of a point process at a specific point t_i, which encompasses all events occurring before t_i, we can define the likelihood function for the parameters of a generic point process as follows:

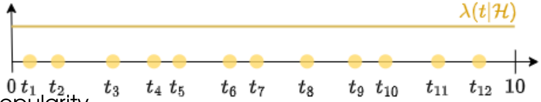
$$\mathcal{L}(\theta) = \prod_{i=1}^{n} \lambda(t_i|\mathcal{H}) e^{-\int_0^{\mathcal{T}} \lambda(t|\mathcal{H})dt}$$

Poisson Process: λ

$$
\begin{aligned}
\mathcal{L}(\theta) &= \prod_{i=1}^{n} \lambda(t_i|\mathcal{H}) e^{-\int_0^{\mathcal{T}} \lambda(t|\mathcal{H})dt} \\
\ell(\theta) &= \log\left(\prod_{i=1}^{n} \lambda(t_i|\mathcal{H}) e^{-\int_0^{\mathcal{T}} \lambda(t|\mathcal{H})dt}\right) \\
&= \sum_{i=1}^{n} \log\lambda(t_i|\mathcal{H}) - \int_0^{\mathcal{T}} \lambda(t|\mathcal{H})dt \\
&= \sum_{i=1}^{n} \log\lambda_P - \int_0^{\mathcal{T}} \lambda_P dt \\
&= n\log\lambda_P - \lambda_P(\mathcal{T} - 0) \\
&= n\log\lambda_P - \lambda_P\mathcal{T}
\end{aligned}
$$

Poisson Process: λ

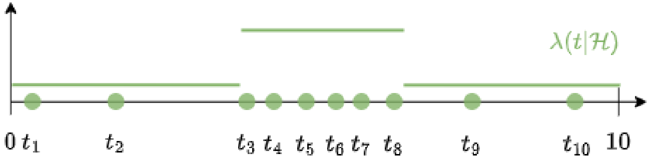
$$\begin{aligned}
\ell(\theta) &= n \log \lambda_P - \lambda_P \mathcal{T} \\
\ell'(\theta) &= n \frac{1}{\lambda_P} - \mathcal{T} \\
0 &= n \frac{1}{\lambda_P} - \mathcal{T} \\
\mathcal{T} &= n \frac{1}{\lambda_P} \\
\frac{\mathcal{T}}{n} &= \frac{1}{\lambda_P} \\
\frac{n}{\mathcal{T}} &= \lambda_P
\end{aligned}$$



$$\lambda_P = \frac{n}{\mathcal{T}} = \frac{12}{10} = 1.2$$

Curious Audiences

- The Poisson Process might not be the best choice for modeling curious audiences.
- As we observed in the examples, bursts of events occur unexpectedly and with varying intensities.
- However, curious behaviors happen continuously.
- During quiet periods, they may not be well-observed because the intensity is lower.
- Therefore, we need to find a process that alternates between calm and bursts of events. An ideal $\lambda_S(t|\mathcal{H})$ would be



Self-Feeding Process

- A point process can be fully defined if we specify its intensity function $\lambda(t|\mathcal{H})$.
- However, obtaining the ideal function as mentioned earlier can be challenging without making some strong assumptions.
- Let's consider a more accessible approximation:

SFP Intensity Function

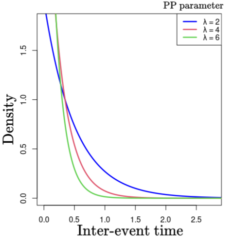
Consider a point process with n observed times $T = \{t_1, t_2, \ldots, t_n\}$ sampled from a Self-Feeding Process (SFP). The intensity function, defined by a parameter μ, is as follows:

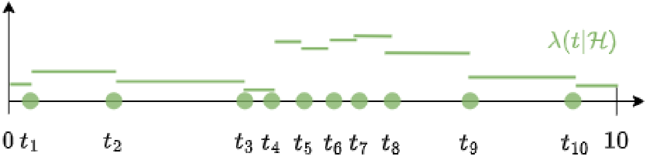
$$\lambda_S(t|\mathcal{H}) = \begin{cases} \mu & \text{if } t \leq t_1, \\ \frac{1}{t_1 + \frac{\mu}{e}} & \text{if } t_1 < t \leq t_2, \\ \frac{1}{\Delta t + \frac{\mu}{e}} & \text{otherwise.} \end{cases}$$

Here, Δt is the difference between the two last events before t.

Self-Feeding Process

$$\lambda_S(t|\mathcal{H}) = \begin{cases} \mu & \text{if } t \leq t_1, \\ \frac{1}{t_1 + \frac{\mu}{e}} & \text{if } t_1 < t \leq t_2, \\ \frac{1}{\Delta t + \frac{\mu}{e}} & \text{otherwise.} \end{cases}$$



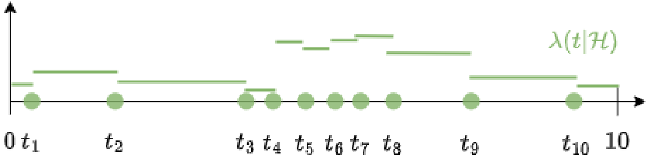


MLE for SFP

- We know that the likelihood for point processes is defined as:

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} \lambda(t_i|\mathcal{H}) e^{-\int_0^{\mathcal{T}} \lambda(t|\mathcal{H})dt}$$

- By observing events, we can compute $\lambda(t|\mathcal{H})$ for any μ:



- Unfortunately, there is no closed-form solution for μ in SFP. However, finding a numerical minimum is straightforward since it's a single parameter.
- We estimate μ using the set $\Delta T = \{t_2 - t_1, t_3 - t_2, \dots, t_n - t_{n-1}\}$:

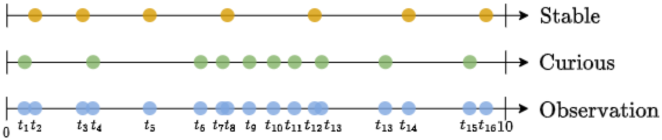
$$\mu \approx \text{Median}(\Delta T)$$

Observation

- We have defined models for both types of audiences: stable (PP) and curious(SFP).
- Given a time series $T = \{t_1, t_2, \ldots, t_n\}$, our goal is to determine the respective parameters of the processes: λ_P and μ.
- We assume that the process generating the stable and curious audiences operate independently.
- If we know which events belong to each process, calculating λ_P and μ is straightforward.
- However, we only observe **a mixture** of both processes.

Observation

- We observe only the joint process $T = \{t_1, t_2, \ldots, t_n\}$.



- Let $z_i \in \{0, 1\}$ be the unobserved labels for the events.
  - $z_i = 0$ if t_i is from the stable (PP). $z_i = 1$ if t_i is from the stable (SFP).
- Therefore, we aim to find the set $Z = \{z_1, z_2, \ldots, z_n\}$, λ_P, and μ that maximizes:

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} \lambda(t_i|\mathcal{H}) e^{-\int_0^{\mathcal{T}} \lambda(t|\mathcal{H})dt}$$

- Randomly choosing labels would be impractical due to the vast number of possibilities.

EM Algorithm

- The Expectation-Maximization (EM) algorithm is a powerful statistical method widely used for estimating parameters in models with hidden or missing data.
- It is an iterative algorithm that alternates between two main steps:
  1. **Expectation (E-step)**: In this step, we calculate the expected values of the missing or **latent variables** using the current parameter estimates.
  2. **Maximization (M-step)**: In this step, we update the **model parameters** to maximize the likelihood based on the expected values obtained in the E-step.
- The EM algorithm continues iterating between the E-step and M-step until convergence, where the parameter estimates no longer change significantly.

In our problem, what are the **latent variables** and what are the **model parameters**?

EM Algorithm

- The labels Z are **latent variables** and $\theta = \{\lambda_P, \mu\}$ **model parameters**.

EM Algoritm - PP + SFP

```
Initialise $Z^{(0)}$, $\lambda_P$ and $\mu$ (e .g, randomly). Set a $N > 0$.
UNTIL $\lambda_P$ and $\mu$ converge DO
E-STEP
    • Update $\forall j$ update $Z^{(j)}$ from previous $Z$s.
    • FOR $j \in \{1, 2, \cdots N\}$ DO
        – $\forall_i$ sample $Z_i^{(j)}$ $\sim$Bernoulli$\left(\frac{\mathcal{L}(\lambda_P,\mu,Z_{-i}^{(j)},z_i=1)}{\mathcal{L}(\lambda_P,\mu,Z_{-i}^{(j)},z_i=0)+\mathcal{L}(\lambda_P,\mu,Z_{-i}^{(j)},z_i=1)}\right)$.
M-STEP
    • Estimate the parameters $\lambda_P$ and $\mu$ as
        $\lambda_P = \frac{\sum_j^N \frac{\sum_j^n (1-z_i^{(j)})}{\mathcal{T}}}{N}$ and $\mu = \frac{\sum_j^N \text{Median}(\Delta T | Z^{(j)})}{N}$
```

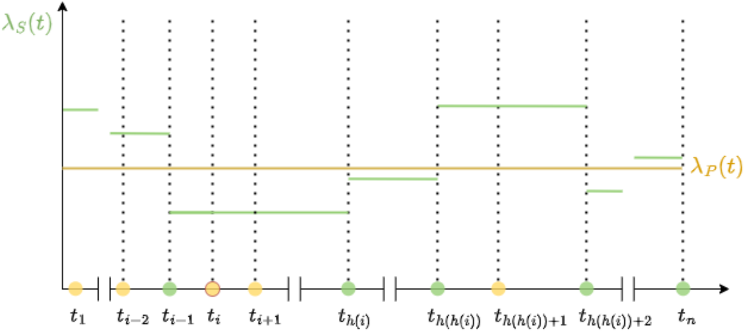
MLE of the Mixture

- Z_{-i} represents all the z's except z_i.
- The likelihood function of the mixture can be written as:

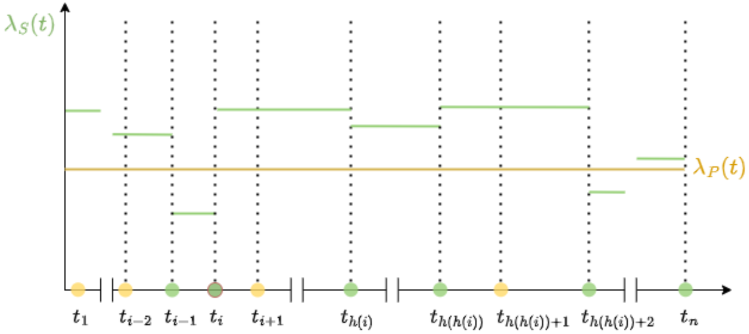
$$\mathcal{L}(\lambda_P, \mu, Z) = \prod_{i=1}^{n} \lambda_S(t_i|\mathcal{H})^{z_i} \lambda_P^{1-z_i} e^{-\int_0^{\mathcal{T}} (\lambda_S(t|\mathcal{H})+\lambda_P)dt}$$

- Consequently, we can compute $\mathcal{L}(\lambda_P, \mu, Z_{-i}^{(j)}, z_i = 1)$ and $\mathcal{L}(\lambda_P, \mu, Z_{-i}^{(j)}, z_i = 0)$. However, this would be computationally expensive.
- Let $h(i)$ denote the next SFP point after t_i.

MLE of the Mixture $z_i = 0$



MLE of the Mixture $z_i = 1$

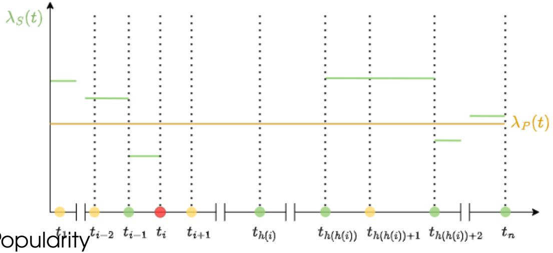


MLE of the Mixture

Note we can factorise $\mathcal{L}(\lambda_P, \mu, Z_{-i}, z_i = z)$ as:

$$\mathcal{L}(\lambda_P, \mu, Z_{-i}, z_i = z) = \prod_{j=i}^{h(h(i))} \lambda_S(t_j|\mathcal{H})^{z_j} \lambda_P^{1-z_j} e^{-\int_{t_i}^{t_{h(h(i))}} (\lambda_S(t|\mathcal{H}) + \lambda_P) dt}$$

$$\times \prod_{\mathbf{j=1}}^{\mathbf{i-1}} \lambda_{\mathbf{S}}(\mathbf{t_j}|\mathcal{H})^{\mathbf{z_j}} \lambda_{\mathbf{P}}^{\mathbf{1-z_j}} \prod_{\mathbf{j=h(h(i))}}^{\mathbf{n}} \lambda_{\mathbf{S}}(\mathbf{t_j}|\mathcal{H})^{\mathbf{z_j}} \lambda_{\mathbf{P}}^{\mathbf{1-z_j}} \mathbf{e}^{-\int_{\mathbf{0}}^{\mathbf{t_i}} (\lambda_{\mathbf{S}}(\mathbf{t}|\mathcal{H}) + \lambda_{\mathbf{P}}) \mathbf{dt} - \int_{\mathbf{t_{h(h(i))}}}^{\mathcal{T}} (\lambda_{\mathbf{S}}(\mathbf{t}|\mathcal{H}) + \lambda_{\mathbf{P}}) \mathbf{dt}}$$



MLE of the Mixture

Let $\mathcal{L}(\bullet)^{[t_i, t_{h(h(i))}]}$ be the likelihood computed in the interval $[t_i, t_{h(h(i))}]$ defined as

$$\mathcal{L}(\bullet)^{(t_i, t_{h(h(i))}]} = \prod_{j=i}^{h(h(i))} \lambda_S(t_j|\mathcal{H})^{z_j} \lambda_P^{1-z_j} e^{-\int_{t_i}^{t_{h(h(i))}} (\lambda_S(t|\mathcal{H}) + \lambda_P) dt}$$

and a constant α the likelihood computed outside of the same interval $[t_i, t_{h(h(i))}]$ as

$$\alpha = \mathcal{L}(\bullet)^{(\mathbf{0},\mathcal{T}]/[\mathbf{t_i},\mathbf{t_{h(h(i))}}]} =$$

$$\prod_{\mathbf{j=1}}^{\mathbf{i-1}} \lambda_{\mathbf{S}}(\mathbf{t_j}|\mathcal{H})^{\mathbf{z_j}} \lambda_{\mathbf{P}}^{\mathbf{1-z_j}} \prod_{\mathbf{j=h(h(i))}}^{\mathbf{n}} \lambda_{\mathbf{S}}(\mathbf{t_j}|\mathcal{H})^{\mathbf{z_j}} \lambda_{\mathbf{P}}^{\mathbf{1-z_j}} \mathbf{e}^{-\int_{\mathbf{0}}^{\mathbf{t_i}} (\lambda_{\mathbf{S}}(\mathbf{t}|\mathcal{H}) + \lambda_{\mathbf{P}}) \mathbf{dt} - \int_{\mathbf{t_{h(h(i))}}}^{\mathcal{T}} (\lambda_{\mathbf{S}}(\mathbf{t}|\mathcal{H}) + \lambda_{\mathbf{P}}) \mathbf{dt}},$$

.

MLE of the Mixture

Call $\mathcal{L}(\lambda_P, \mu, Z_{-i}, z_i = z)$ by $\mathcal{L}(\bullet, z_i = 1)$ and $\lambda_S(t|\mathcal{H}|z_i = z) + \lambda_P$ by $\lambda(t|z)$. Then we have

$$\frac{\mathcal{L}(\bullet, z_i = z)}{\mathcal{L}(\bullet, z_i = 0) + \mathcal{L}(\bullet, z_i = 1)} = \frac{\left[\mathcal{L}(\bullet)^{(t_i, t_{h(h(i))})]}\right]_{z_i=z} \quad \alpha}{\left[\mathcal{L}(\bullet)^{(t_i, t_{h(h(i))})]}\right]_{z_i=0} \quad \alpha + \left[\mathcal{L}(\bullet)^{(t_i, t_{h(h(i))})]}\right]_{z_i=1} \quad \alpha}$$

$$= \frac{\left[\mathcal{L}(\bullet)^{(t_i, t_{h(h(i))})]}\right]_{z_i=z}}{\left[\mathcal{L}(\bullet)^{(t_i, t_{h(h(i))})]}\right]_{z_i=0} + \left[\mathcal{L}(\bullet)^{(t_i, t_{h(h(i))})]}\right]_{z_i=1}}$$

EM Algorithm

- Finally we have all the tools for the EM-Algorithm.

EM Algoritm - PP + SFP

```
Initialise $Z^{(0)}$, $\lambda_P$ and $\mu$ (e .g, randomly). Set a $N > 0$.
UNTIL $\lambda_P$ and $\mu$ converge DO
E-STEP
    • Update $\forall j$ update $Z^{(j)}$ from previous $Z$s.
    • FOR $j \in \{1, 2, \cdots N\}$ DO
        – $\forall_i$ sample $Z_i^{(j)}$ $\sim$Bernoulli$\left(\frac{\mathcal{L}(\lambda_P,\mu,Z_{-i}^{(j)},z_i=1)}{\mathcal{L}(\lambda_P,\mu,Z_{-i}^{(j)},z_i=0)+\mathcal{L}(\lambda_P,\mu,Z_{-i}^{(j)},z_i=1)}\right)$.
M-STEP
    • Estimate the parameters $\lambda_P$ and $\mu$ as
          $\lambda_P = \frac{\sum_j^N \frac{\sum_j^n (1-z_i^{(j)})}{\mathcal{T}}}{N}$ and $\mu = \frac{\sum_j^N \text{Median}(\Delta T | Z^{(j)})}{N}$
```

Obrigado :) - Faculty of Information Technology