A leader/follower based recommendation system for bonds

IdR BNPP - Initiative de recherche BNPP

Guillaume Lecué

CNRS, CREST, ENSAE

21 March 2015 – Paris Dauphine

Introduction : Aim and Database

Aim : Construct a system that recommends bonds to clients

Every day :

1) **List of recommendations** :

   Ticker *FGBUH* to Client *468*

2) **trader (dealer, sale)** : checks availability and quantity of Ticker *FGBUH* and call client *468* :

   sell 500.000 euros of FGBUH to 468

3) **Client 468** : OK !

(Note : we may recommend a sector instead of bonds).
Aims :

1) Anticipate on clients requests
2) Make proposals to clients who may not have thought about it
3) Gain customers loyalty
4) Simplify trader's work
5) Optimize inventories

Database

Request For Quote (RFQ) from 2013/1/11 to 2014/10/31.
79071 rows, 57 columns.

'Customer'	468	'TradeDate'	20140203
'Sales'	12	'RFQTime'	70902.0
'Trader'	18	'RFQClosedTime'	70914.0
'Trader1'	2	'ClientTier'	'TIER 3'
'SalesTeam'	12	'NbDealers'	3.0
'CustomerTrader'	4135	'NotionalEUR'	147830.586148
'VoiceElec'	'ELEC'	'NotionalUSD'	200000.0
'BuySell'	'Sell'	'NotionalCCY'	200000.0
'Currency'	'USD'	'StatusDetails'	'CustRejectedQuote'
'RFQId'	'00000001_20140203_08'	'TradeStatus'	'TimeOut'
'Venue'	'EBNP'	'BNPPStreamedMid'	100.6
'RFQOrderType'	'Inquiry'	'BNPPStreamedQuantityCcy'	1000000.0
'InitAuditEventTimestamp'	1391411342155	'BNPPStreamedQuote'	100.35
'FinalAuditEventTimestamp'	1391411355188	'RFQCompositePrice'	100.42500305200001

Database

'BNPPAnsweredQuote'	100.35	'Seniority'	'Senior'
'CoverPrice'	0.0	'bbgName'	'FGBUH 3 1/4 01/14/19'
'Bid2Mid'	0.25	'Isin'	'XS0992167865'
'AutoNegStatus'	'Trader'	'CDSIssuer'	'FIRST GULF'
'Region'	'EUROPE'	'CDSSector'	'FINANCIAL'
'Activity'	'LOCAL MARKET'	'CDSSeniority'	'SEN'
'RiskCaptain'	'LOCAL MARKET'	'SalesLocation'	'SIN'
'Owner'	'EUROPE LM TT'	'TraderLocation'	'LON'
'IsFRN'	0.0	'JagPositionCCY'	1000000000.0
'Ticker'	'FGBUH'	'StarPositionCCY'	1000000000.0
'Maturity'	4.0	'Benchmark'	nan
'Sector'	'Bank'	'ANPrice'	nan
'Rating'	'A'	'PV01'	nan
'IssuerNationality'	'AE'	'StpPortfolio'	nan
		'Bid2MidComp'	0.27499771

Some difficulties

1. there is no fixed catalog of bonds (=> we may have to recommend clusters of bonds – or sectors – as a first stage)
2. taste of clients depend on the time of the year
3. there is a Leaders/Followers idea (L. Carlier, BNPP)

Easy solution without the Leaders/followers algorithm we would just learn a RC on the last month RFQ's (avantages : no modelisation issue)

Today talk is about a prospective Leaders/Followers based RC

A three step Leaders/Followers recommendation system

A three step Leaders/Followers recommendation system

1) Representations of the clients and the bonds into a vectors space like $\mathbb{R}^d$ – called the **Latent space** – according to the clients taste.
2) Cluster the clients and the bonds in the Latent space and identify the affinity between clusters of clients and bonds.
3) For each cluster of clients (with identical taste) identify leaders and their followers using a multidimensional Hawkes model for the RFQ of bonds that are liked by this cluster of clients.

Construction of a latent space via NMF

General idea behind recommendation systems

Problem : recommend a list of *bonds* to *user* (Bob)

Idea :

1. Bob likes *bond1*
2. Alice likes *bond1* and *bond2* (Alice and Bob have the same taste for bonds)
3. Recommend *bond2* to Bob

Proximity : Bob and Alice are close when they share similar taste. So it may be a good idea to recommend things that are liked by Alice to Bob

Construction of the users/items matrix

users/items matrix :

Item/ User	10	11	13	14	15	16	17	18	19
1	2	3	4	5					
2		6			6	1	10		
3		4	1		7			2	
4		8	1						5
5		2							
6							5	1	

1. User : '468'
2. Item : 'FGBUH' or 'Sector'
3. Rate : 'NotionalEUR' or some grades (based on quantile or proportion of money spent on such a bond)

Matrix completion algorithm

Item/ User	10	11	13	14	15	16	17	18	19
1	2	3	4	5					
2		6			6	1	10		
3		4	1		7			2	
4		8	1						5
5		2							
6							5	1	

⇓ Matrix Completion ⇓

Item/ User	10	11	13	14	15	16	17	18	19
1	2	3	4	5	2	4	2	3	1
2	4	6	3	4	6	1	10	9	3
3	7	4	1	3	7	7	2	2	9
4	6	8	1	4	6	10	2	3	5
5	1	2	7	2	9	2	9	9	2
6	7	8	2	10	2	3	5	1	3

Outputs :

1. Recommend *item* '16' to *user* '4'
2. Proximity of *users* '3' and '4' (same cluster of clients)

Matrix completion via NMF (Non-Negative Matrix Factorization)

Idea : We construct a space $\mathbb{R}^d$ where *users* and *items* are represented by vectors :

$$(p_i)_{1 \leq i \leq p} \text{ and } (q_j)_{1 \leq j \leq q}$$

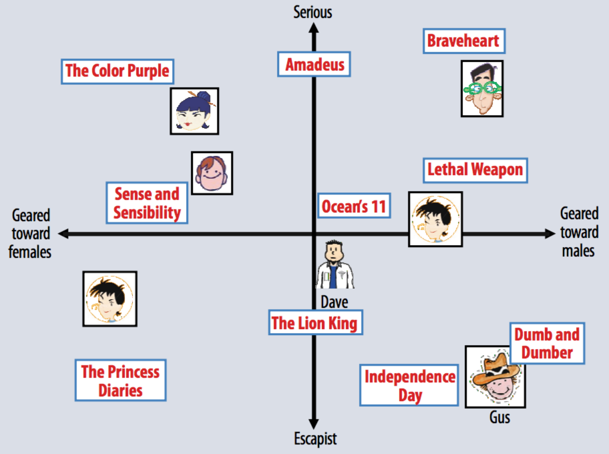
1. p_i : representation of user i in the latent space $\mathbb{R}^d$
2. q_j : representation of item j in $\mathbb{R}^d$

such that

$$\langle p_i, q_j \rangle_{\mathbb{R}^d}$$

measures the affinity between *item* j and *user* i

Latent space for the Netflix prize



Matrix Factorization

Problem : find the vectors $(p_i)_{1\leq i\leq p}$ and $(q_j)_{1\leq j\leq q}$
Data : $\{(user_i, item_j, grade_{ij}) : (i,j)\in\Omega\}$ where

$$\Omega \subset \{1,\ldots,p\}\times\{1,\ldots,q\}$$

Idea : Construct $(p_i)_{1\leq i\leq p}$ and $(q_j)_{1\leq j\leq q}$ such that

$$\langle p_i, q_j\rangle \approx grade_{ij} \quad \forall (i,j)\in\Omega$$

All other values $\langle p_i, q_j\rangle$ for $(i,j)\notin\Omega$ are used to complete the *users/items* matrix
Objective function : minimize

$$(p_i)_{1\leq i\leq p},(q_j)_{1\leq j\leq q} \longrightarrow \sum_{(i,j)\in\Omega}\left(grade_{ij}-\langle p_i, q_j\rangle\right)^2$$

Matrix Factorization

Problem : find the vectors $(p_i)_{1\leq i\leq p}$ and $(q_j)_{1\leq j\leq q}$
Data : $\{(user_i, item_j, grade_{ij}) : (i,j) \in \Omega\}$ where

$$\Omega \subset \{1,\ldots,p\} \times \{1,\ldots,q\}$$

Idea : Construct $(p_i)_{1\leq i\leq p}$ and $(q_j)_{1\leq j\leq q}$ such that

$$\langle p_i, q_j \rangle \approx grade_{ij} \quad \forall (i,j) \in \Omega$$

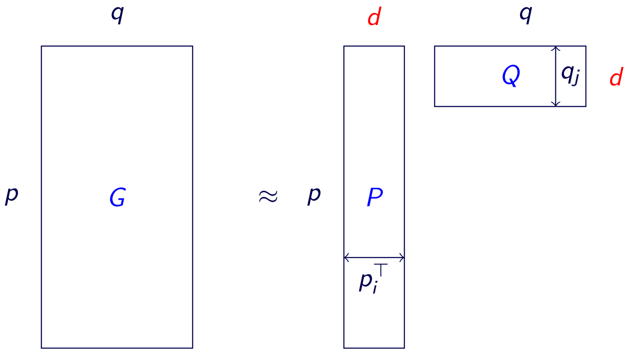
All other values $\langle p_i, q_j \rangle$ for $(i,j) \notin \Omega$ are used to complete the *users/items* matrix
Objective function : minimize

$$(p_i)_{1\leq i\leq p}, (q_j)_{1\leq j\leq q} \longrightarrow \sum_{(i,j)\in\Omega} \left(grade_{ij} - \langle p_i, q_j \rangle\right)^2 = \|P_\Omega(G - PQ)\|_2^2$$

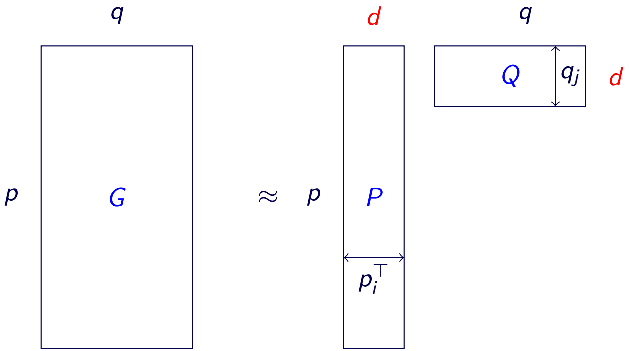
where

$$G = \left(grade_{ij}\right)_{i,j} \quad P = (p_i^\top)_{1\leq i\leq p} \quad Q = (q_i)_{1\leq i\leq q}.$$

Matrix factorization : Factorization of G through $\mathbb{R}^d$



Matrix factorization : Factorization of G through $\mathbb{R}^d$



Minimize

$$P \in \mathbb{R}^{p \times d}, Q \in \mathbb{R}^{d \times q} \to \|P_\Omega(G - PQ)\|_2^2$$

Matrix factorization via ALS (Alternating least Squares)

$$(P, Q) \in \mathbb{R}^{p\times d} \times \mathbb{R}^{d\times q} \to f(P, Q) := \|P_\Omega(G - PQ)\|_2^2$$

is **not** a convex function of (P, Q), but :

Matrix factorization via ALS (Alternating least Squares)

$$(P, Q) \in \mathbb{R}^{p \times d} \times \mathbb{R}^{d \times q} \rightarrow f(P, Q) := \|P_\Omega(G - PQ)\|_2^2$$

is **not** a convex function of (P, Q), but :

1. for a fixed P^*, $Q \rightarrow f(P^*, Q)$ is convex
2. for a fixed Q^*, $P \rightarrow f(P, Q^*)$ is convex

We perform ALS to minimize f :

Data: $\{G_{ij}, (i, j) \in \Omega\}$
Init : $P_0 \in \mathbb{R}^{p \times d}$, $Q_0 \in \mathbb{R}^{d \times q}$
Output : P and Q
while *stopping criteria* **do**

$$P_{k+1} = P_k - \eta_k \partial_1 f(P_k, Q_k)$$

$$Q_{k+1} = Q_k - \mu_k \partial_2 f(P_{k+1}, Q_k)$$

end

Non-Negative Matrix factorization (NMF)

Rem : Coordinates of p_i and q_j may be negative and so are some inner products $\langle p_i, q_j \rangle \Rightarrow$ Not easy to interpret when we are looking for positive rates

Idea : We restrict the Matrix Factorization to vectors with non-negative coordinates : minimize

$$(P, Q) \in \mathbb{R}_+^{p \times d} \times \mathbb{R}_+^{d \times q} \to f(P, Q) = \| P_\Omega(G - PQ) \|_2^2$$

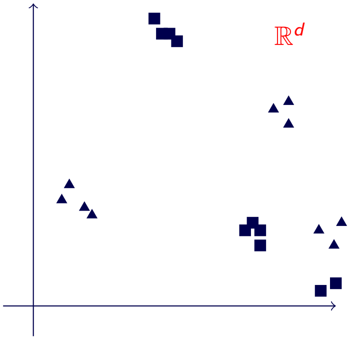
Rem : There is a closed-form solution to the projected ALS (cf. Lee & Seung, NIPS 2001)

Users and *Items* representation in the Latent Space

1. Unlike the classical way to use NMF, we will not construct a recommendation system based on $\langle p_i, q_j \rangle$ for $(i, j) \notin \Omega$

Users and *Items* representation in the Latent Space

1. Unlike the classical way to use NMF, we will not construct a recommendation system based on $\langle p_i, q_j \rangle$ for $(i, j) \notin \Omega$
2. but, we will use the *Users* and *Items* representation in the Latent Space to construct users and items clusters (based on the consumption of items by the users)



Cluster clients and bonds in the latent space

Problem

Given the p_i's and the q_j's (two sets of vectors in $\mathbb{R}^d$), construct clusters of users and items :

Problem

Given the p_i's and the q_j's (two sets of vectors in $\mathbb{R}^d$), construct clusters of users and items :

1. *Users Clusters :*

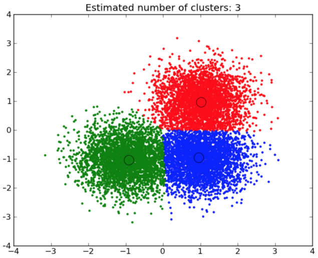
$$CU_1 = \{p_i : i \in I_1\}, \dots, CU_r = \{p_i : i \in I_r\}$$

such that $I_1 \cup \dots \cup I_r = \{1, \dots, p\}$

2. *Bonds Clusters :*

$$CB_1 = \{q_i : i \in J_1\}, \dots, CB_s = \{q_j : j \in J_s\}$$

such that $J_1 \cup \dots \cup J_s = \{1, \dots, q\}$



A clustering algorithm : k-means (++)

Data: $\mathcal{D} = \{p_i, 1 \leq i \leq p\}$
Init (++) : $\bar{p}_1 = p^{(1)}, \ldots, \bar{p}_k = p^{(k)} \in \mathcal{D}$
Output : $CU_1, \ldots, CU_k$ with centroids $\bar{p}_1, \ldots, \bar{p}_k$
while *Stopping Criteria* **do**
  for $i \leftarrow 1$ **to** k **do**

$$CU_i = \{p_j : \|p_j - \bar{p}_i\| \leq \|p_j - \bar{p}_{i^*}\|, \forall i^* = 1, \ldots, k\}$$

$$\bar{p}_i = \text{ barycenter of } CU_i = \frac{1}{|CU_i|} \sum_{p \in CU_i} p$$

  end
end

A Leaders/Followers algorithm for (multidimensional) Hawkes processes

Problem formulation

We have clustered users and bonds according to the taste of the users :

$$CU_1, \ldots, CU_r \text{ and } CB_1, \ldots, CU_s$$

For each cluster of users CU_i find the cluster of bonds CB_j which is the closest. For instance by computing the inner products between centroids :

$$\langle \bar{p}_i, \bar{q}_j \rangle_{\mathbb{R}^d}$$

and take $j_i \in \text{argmax}_{j \in \{1,\ldots,q\}} \langle \bar{p}_i, \bar{q}_j \rangle_{\mathbb{R}^d}$
We end up with couples :

$$(CU_1, CB_{j_1}), \ldots, (CU_r, CB_{j_r})$$

Problem : For each couple (CU_i, CB_{j_i}), find the leaders and their followers in CU_i. So that when a leader (in CU_i) buy a bond (in CB_{j_i}), we will recommend this bond to its followers.

Modelisation of RFQ time series by Hawkes processes

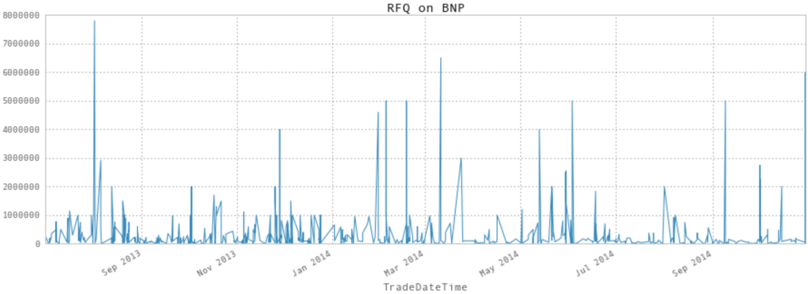
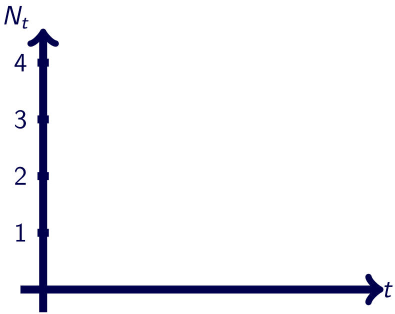


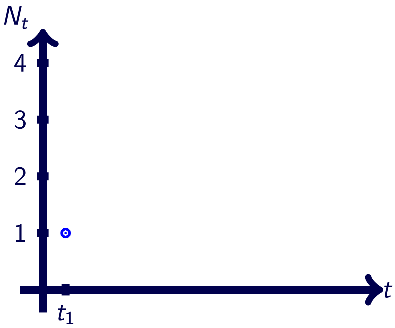
Figure : RFQ with 'BuySell' = 'Sell' on 'Ticker' = 'BNP'

We choose to model RFQ series by Hawkes processes because of its branching structure

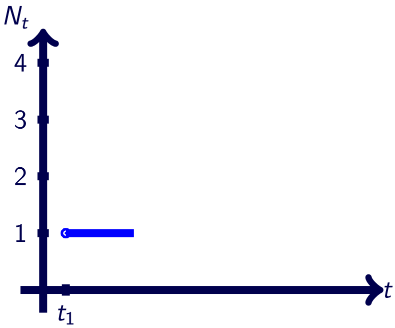
Poisson Process $(N_t)_{t>0}$



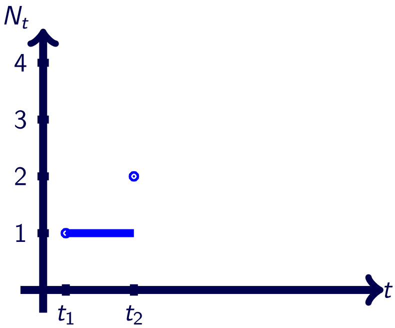
Poisson Process $(N_t)_{t>0}$



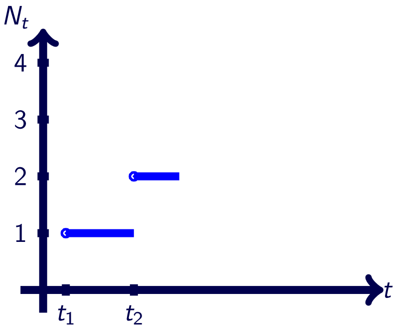
Poisson Process $(N_t)_{t>0}$



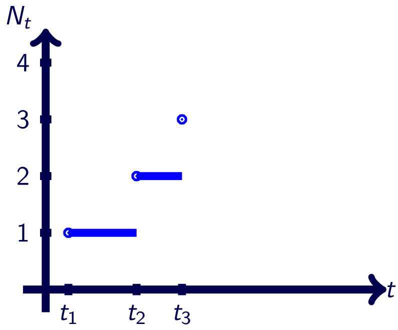
Poisson Process $(N_t)_{t>0}$



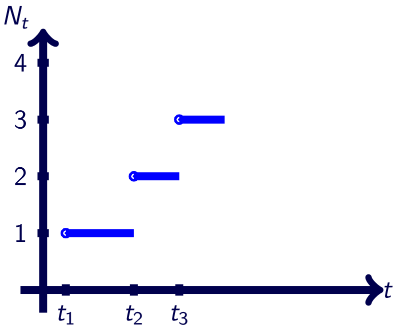
Poisson Process $(N_t)_{t>0}$



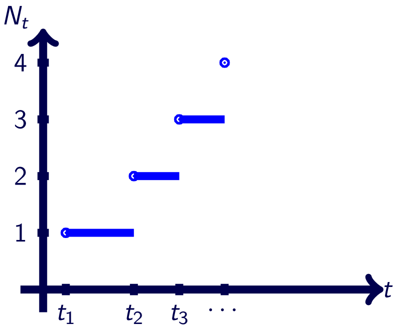
Poisson Process $(N_t)_{t>0}$



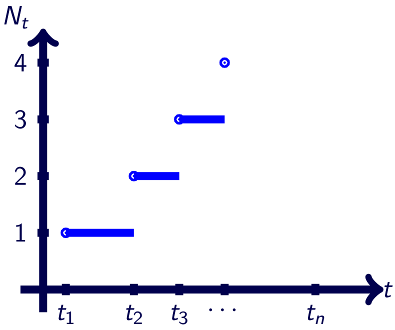
Poisson Process $(N_t)_{t>0}$



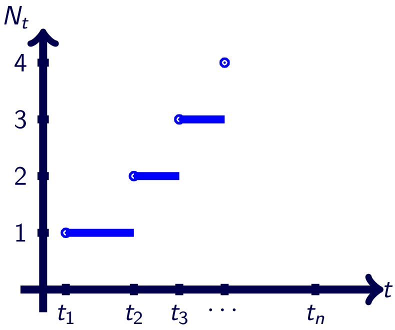
Poisson Process $(N_t)_{t>0}$



Poisson Process $(N_t)_{t>0}$

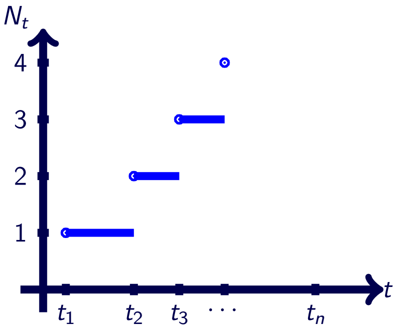


Poisson Process $(N_t)_{t>0}$



1. time elapsed between events $\sim \mathcal{E}(\mu)$

Poisson Process $(N_t)_{t>0}$



1. time elapsed between events $\sim \mathcal{E}(\mu)$
2. $N_t \sim \text{Poisson}(\mu t)$

Hawkes processes

μ is the intensity of the Process ;
We can take an intensity depending on :

1. time t
2. previous jumps $\{t_i : t_i < t\}$

Hawkes processes

μ is the intensity of the Process ;
We can take an intensity depending on :

1. time t
2. previous jumps $\{t_i : t_i < t\}$

and we can make jumps with a magnitude (or more generally with a mark)

Hawkes processes

μ is the intensity of the Process ;
We can take an intensity depending on :

1. time t
2. previous jumps $\{t_i : t_i < t\}$

and we can make jumps with a magnitude (or more generally with a mark)

$$\{(t_i, \kappa_i) : i \in \mathbb{N}\}, \quad t_1 < t_2 < \cdots, \text{ is a marked point process}$$

with intensity

$$\lambda(t) = \lim_{dt \to 0} \frac{\mathbb{E}\left[N(t+dt) - N(t) | \mathcal{F}_t\right]}{dt}$$

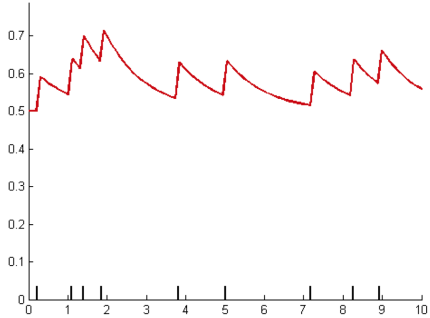
For Hawkes process, we take

$$\lambda(t) = \mu + \sum_{i : t_i < t} \beta(t - t_i, \kappa_i)$$

(and a conditional mark density $\gamma(\kappa|t)$)

Conditional intensity function of a Hawkes process with an exponential kernel :

$$\lambda(t) = 0.5 + \sum_{t_i < t} 0.1 \exp(-2(t - t_i))$$



Every event increases the probability of a new event but its influence decreases exponentially fast. Hawkes processes are used in

1. Seismology (Hawkes, 1971)
2. Epidemiology (ETAS model : 'epidemic-type aftershock sequence')
3. finance market microstructure, Social science, etc.

Conditional intensity of a Hawkes process

$$\lambda(t) = \mu + \sum_{t_i < t} \beta(t - t_i, \kappa_i)$$

1. μ : **Poissonian component** or **immigrant intensity**
2. $\beta(\cdot, \cdot)$: **kernel** or **offspring intensity**

Example of the ETAS model (without geographic mark) :

$$\beta(\tau, \kappa) = \frac{K_0 \exp(a\kappa)}{(\tau + b)^{1+\omega}} I(\kappa > 0)$$

with parameter $\theta = (\mu, K_0, a, b, \omega)$

Hawkes as a marked Poisson cluster process

An alternative way to define a Hawkes process $\{(t_i, \kappa_i)\}$ is as follow :

1) Immigrant events I that follows a Poisson Process with intensity μ

Hawkes as a marked Poisson cluster process

An alternative way to define a Hawkes process $\{(t_i, \kappa_i)\}$ is as follow :

1) Immigrant events I that follows a Poisson Process with intensity μ
2) Each immigrant $t_i \in I$ has an associated mark κ_i with density $\gamma(\cdot|t_i)$

Hawkes as a marked Poisson cluster process

An alternative way to define a Hawkes process $\{(t_i, \kappa_i)\}$ is as follow :

1) Immigrant events I that follows a Poisson Process with intensity μ
2) Each immigrant $t_i \in I$ has an associated mark κ_i with density $\gamma(\cdot|t_i)$
3) Each immigrant $t_i \in I$ generates a **cluster** C_i of events (and these clusters are independent)

Hawkes as a marked Poisson cluster process

An alternative way to define a Hawkes process $\{(t_i, \kappa_i)\}$ is as follow :

1) Immigrant events I that follows a Poisson Process with intensity μ
2) Each immigrant $t_i \in I$ has an associated mark κ_i with density $\gamma(\cdot|t_i)$
3) Each immigrant $t_i \in I$ generates a **cluster** C_i of events (and these clusters are independent)
4) Each cluster C_i consists of events of different generations :

Hawkes as a marked Poisson cluster process

An alternative way to define a Hawkes process $\{(t_i, \kappa_i)\}$ is as follow :

1) Immigrant events I that follows a Poisson Process with intensity μ
2) Each immigrant $t_i \in I$ has an associated mark κ_i with density $\gamma(\cdot|t_i)$
3) Each immigrant $t_i \in I$ generates a **cluster** C_i of events (and these clusters are independent)
4) Each cluster C_i consists of events of different generations :
    - Generation 0 = the immigrant and its mark (t_i, κ_i)

Hawkes as a marked Poisson cluster process

An alternative way to define a Hawkes process $\{(t_i, \kappa_i)\}$ is as follow :

1) Immigrant events I that follows a Poisson Process with intensity μ
2) Each immigrant $t_i \in I$ has an associated mark κ_i with density $\gamma(\cdot|t_i)$
3) Each immigrant $t_i \in I$ generates a **cluster** C_i of events (and these clusters are independent)
4) Each cluster C_i consists of events of different generations :
    - Generation 0 = the immigrant and its mark (t_i, κ_i)
    - Recursively, each t_j of Generation k generates a Poisson Process with intensity $\beta(t - t_j, \kappa_j)$ populating Generation $k + 1$. Each offspring generated from t_j has a mark following density $\gamma(\cdot|t_j)$

Hawkes as a marked Poisson cluster process

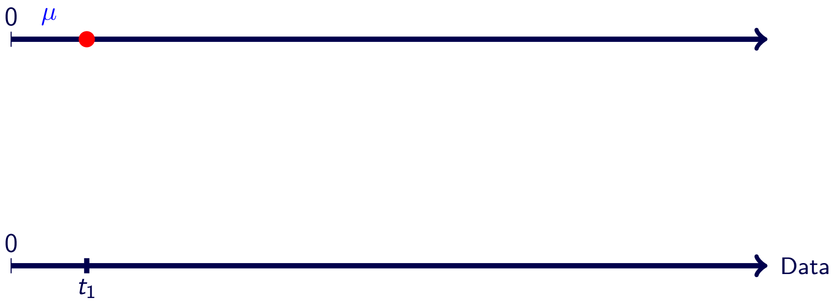
An alternative way to define a Hawkes process $\{(t_i, \kappa_i)\}$ is as follow :

1) Immigrant events I that follows a Poisson Process with intensity μ
2) Each immigrant $t_i \in I$ has an associated mark κ_i with density $\gamma(\cdot|t_i)$
3) Each immigrant $t_i \in I$ generates a **cluster** C_i of events (and these clusters are independent)
4) Each cluster C_i consists of events of different generations :
   - Generation $0 =$ the immigrant and its mark (t_i, κ_i)
   - Recursively, each t_j of Generation k generates a Poisson Process with intensity $\beta(t - t_j, \kappa_j)$ populating Generation $k + 1$. Each offspring generated from t_j has a mark following density $\gamma(\cdot|t_j)$
5) Union of all clusters is a Hawkes process

Clustering model

Clustering model

Clustering model



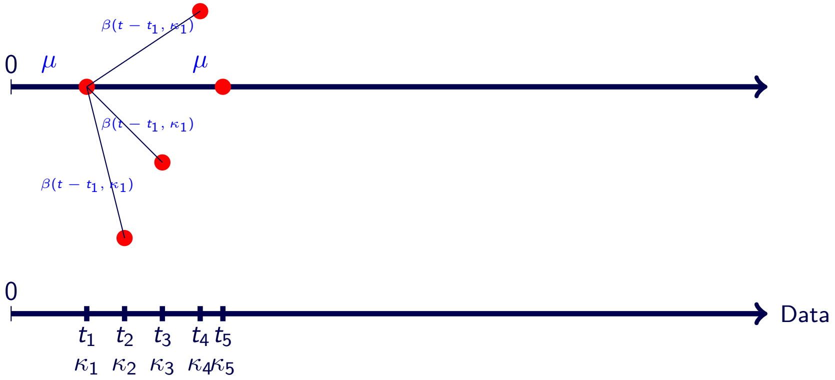
Clustering model

Clustering model

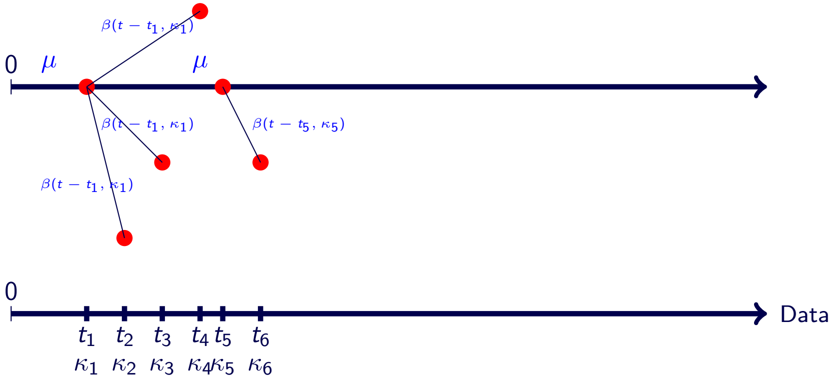
Clustering model

Clustering model

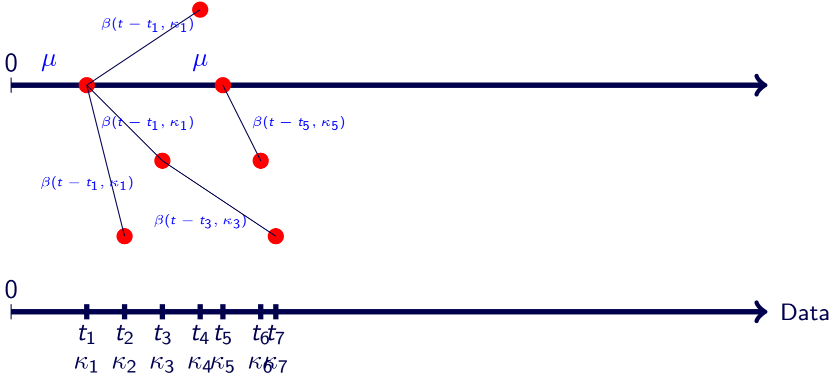
Clustering model



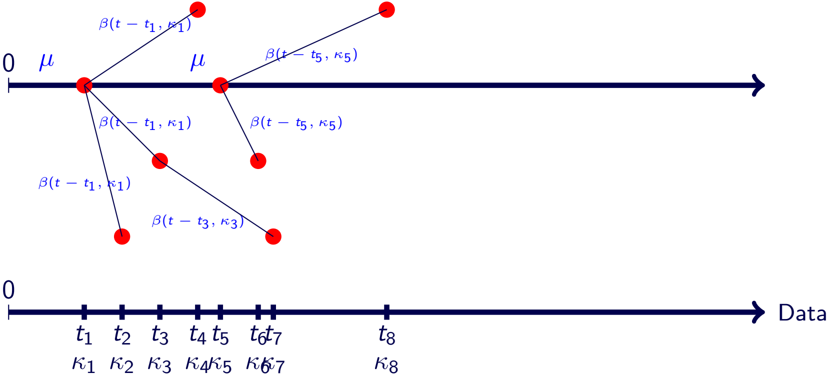
Clustering model



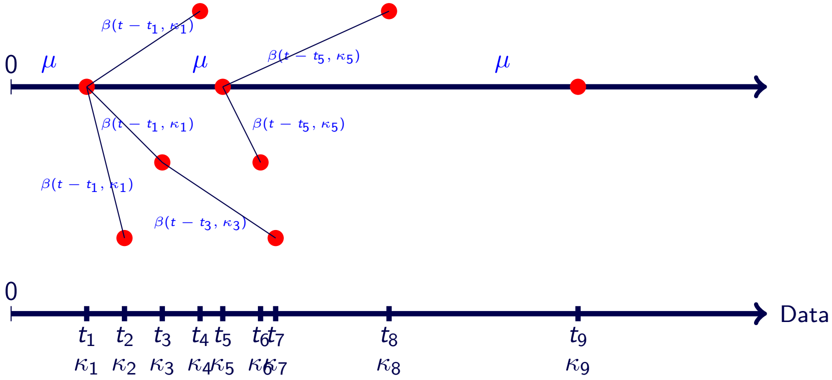
Clustering model



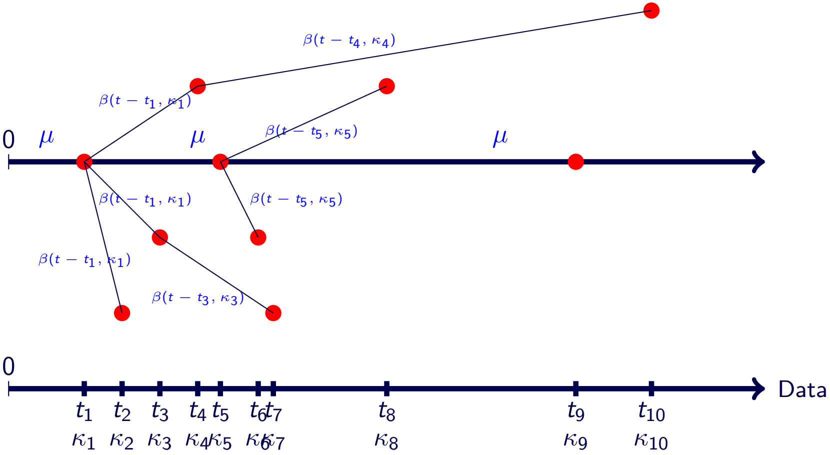
Clustering model



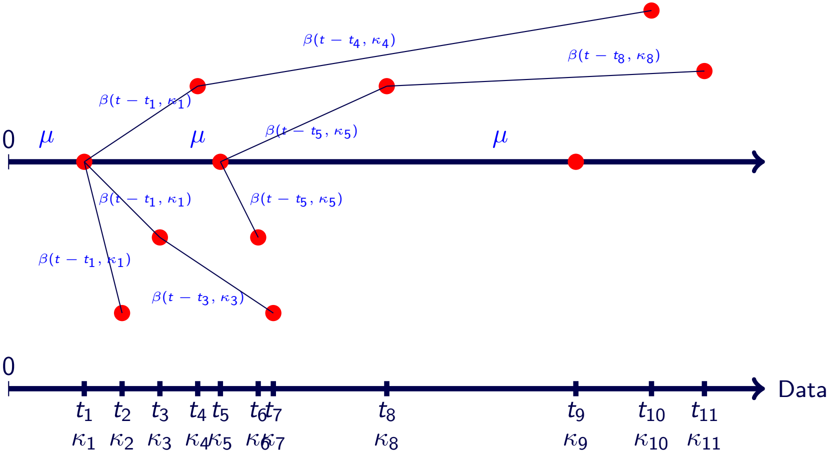
Clustering model



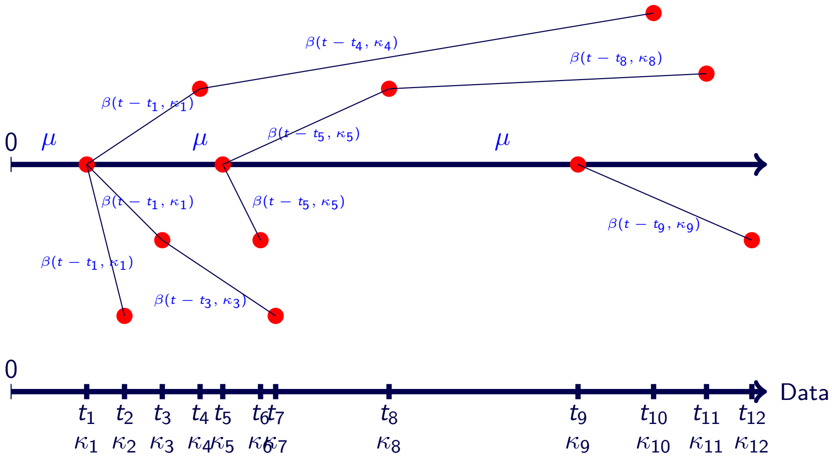
Clustering model



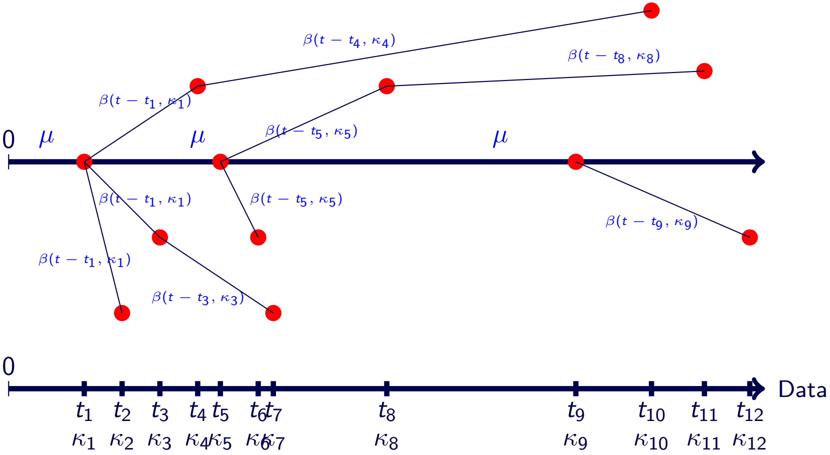
Clustering model



Clustering model

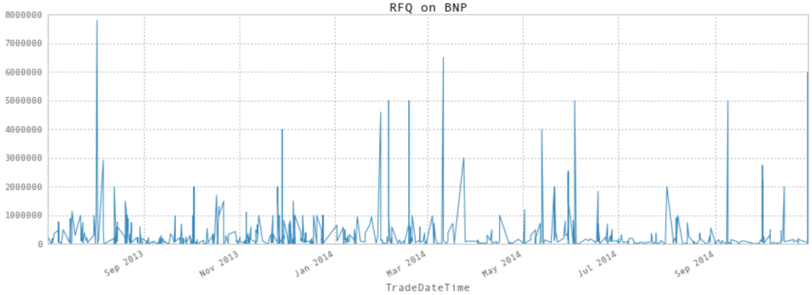


Clustering model



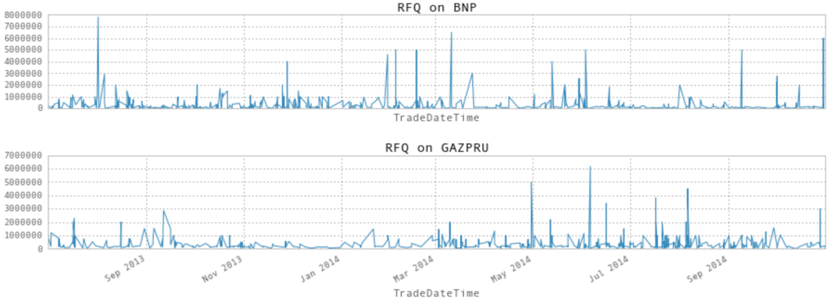
Problem : Given $\{(t_i, \kappa_i) : t_i \leq T\}$, find the branching structure

Branching structure = detection of Leader/followers



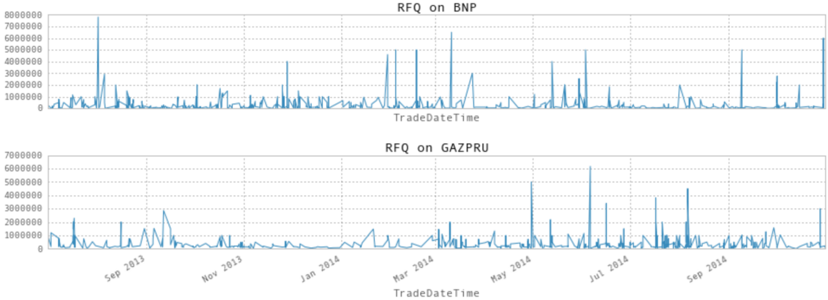
Detection of Leader/followers $\Longrightarrow$ Recommendation to followers

Note on multidimensional Hamkes processes



Each couple (CU_i, CB_{j_i}) is associated with a multidimensional Hawkes process of dimension $2|CB_{j_i}|$ (for 'Buy' and 'Sell').

Note on multidimensional Hamkes processes



Each couple (CU_i, CB_{j_i}) is associated with a multidimensional Hawkes process of dimension $2|CB_{j_i}|$ (for 'Buy' and 'Sell').By doing so, we can infer parenthood relationship like

a leader makes a 'Sell' RFQ on 'BNP' then a follower makes a 'Buy' RFQ on 'GAZPRU'

Estimating the branching structure via an EM algorithm (cf. Veen and Schoenberg)

Idea : We see the branching structure as missing variables :

1. $u_i = 0$ means t_i is an immigrant event
2. $u_i = j$ means t_i was triggered by t_j

A new look at the data :

- Complete data $\{(t_i, \kappa_i, u_i)\}$
- Incomplete data $\{(t_i, \kappa_i)\}$

EM : Expectation-Maximization algorithm

$X^c = (X, X^m) \in \mathcal{O} \times \mathcal{M}$ where

- X^c : complete variables
- X^m : missing variables
- X : observed data (incomplete variables)

X^c is distributed according to $f((x, x^m), \theta^*)$ where

$$f(\cdot, \theta^*) \in \{f(\cdot, \theta) : \theta \in \Theta\}$$

EM : Expectation-Maximization algorithm

$X^c = (X, X^m) \in \mathcal{O} \times \mathcal{M}$ where

- X^c : complete variables
- X^m : missing variables
- X : observed data (incomplete variables)

X^c is distributed according to $f((x, x^m), \theta^*)$ where

$$f(\cdot, \theta^*) \in \{f(\cdot, \theta) : \theta \in \Theta\}$$

Problem : We want to estimate θ^* from the (incomplete) data X
EMV (for complete data) : $\hat{\theta}^{EMV} \in \text{argmax}_{\theta \in \Theta} \log f((X, X^m), \theta)$
but X^m is **not** observed so we integrate according to X^m : let $\theta^{old} \in \Theta$,

$$\log f((X, X^m), \theta) \rightsquigarrow Q(\theta, \theta^{old}) = \int_{\mathcal{M}} \log f((X, x^m), \theta) f(x^m | x, \theta^{old}) dx^m$$

EM : Expectation-Maximization algorithm

$X^c = (X, X^m) \in \mathcal{O} \times \mathcal{M}$ where

- X^c : complete variables
- X^m : missing variables
- X : observed data (incomplete variables)

X^c is distributed according to $f((x, x^m), \theta^*)$ where

$$f(\cdot, \theta^*) \in \{f(\cdot, \theta) : \theta \in \Theta\}$$

Problem : We want to estimate θ^* from the (incomplete) data X
EMV (for complete data) : $\hat{\theta}^{EMV} \in \text{argmax}_{\theta \in \Theta} \log f((X, X^m), \theta)$
but X^m is **not** observed so we integrate according to X^m : let $\theta^{old} \in \Theta$,

$$\log f((X, X^m), \theta) \rightsquigarrow Q(\theta, \theta^{old}) = \int_{\mathcal{M}} \log f((X, x^m), \theta) f(x^m | x, \theta^{old}) dx^m$$

EM algorithm does recursively the following :

E Expectation step : compute $Q(\theta, \theta^{old})$

EM : Expectation-Maximization algorithm

$X^c = (X, X^m) \in \mathcal{O} \times \mathcal{M}$ where

- X^c : complete variables
- X^m : missing variables
- X : observed data (incomplete variables)

X^c is distributed according to $f((x, x^m), \theta^*)$ where

$$f(\cdot, \theta^*) \in \{f(\cdot, \theta) : \theta \in \Theta\}$$

Problem : We want to estimate θ^* from the (incomplete) data X
EMV (for complete data) : $\hat{\theta}^{EMV} \in \text{argmax}_{\theta \in \Theta} \log f((X, X^m), \theta)$
but X^m is **not** observed so we integrate according to X^m : let $\theta^{old} \in \Theta$,

$$\log f((X, X^m), \theta) \rightsquigarrow Q(\theta, \theta^{old}) = \int_{\mathcal{M}} \log f((X, x^m), \theta) f(x^m | x, \theta^{old}) dx^m$$

EM algorithm does recursively the following :

E Expectation step : compute $Q(\theta, \theta^{old})$

M Maximization step : maximize $\theta \in \Theta \mapsto Q(\theta, \theta^{old})$
(and update $\theta^{new} \hookrightarrow \theta^{old}$)

The E step – I

To compute

$$Q(\theta, \theta^{old}) = \int_{\mathcal{M}} \left[\log f\left(\{t_i, \kappa_i\}, (u_i), \theta\right)\right] f\left((u_i)|\{t_i, \kappa_i\}, \theta^{old}\right) d(u_i)$$

we compute (in general and when possible)

1. the likelihood of the complete data :

$$f\left(\{t_i, \kappa_i\}, (u_i), \theta\right)$$

The E step – I

To compute

$$Q(\theta, \theta^{old}) = \int_{\mathcal{M}} \left[\log f\left(\{t_i, \kappa_i\}, (u_i), \theta\right)\right] f\left((u_i) | \{t_i, \kappa_i\}, \theta^{old}\right) d(u_i)$$

we compute (in general and when possible)

1. the likelihood of the complete data :

$$f\left(\{t_i, \kappa_i\}, (u_i), \theta\right)$$

2. the conditional density :

$$f\left((u_i) | \{t_i, \kappa_i\}, \theta^{old}\right) \quad (1)$$

⚠ branching structure is deduced from (1) by computing

$$\mathbb{P}\left[u_i = j | \{t_i, \kappa_i\}, \theta^{old}\right] \text{ for all } i, j$$

The E step – II

The density function of a marked point process $\{(t_i, \kappa_i)_{i=1}^n\}$ observed in interval $[0, T)$ (for some fixed T) with :

1. conditional intensity $\lambda(t) = \lambda(t|\mathcal{F}_t)$
2. conditional mark density $\gamma(\cdot|t)$

is given by

$$f(\{(t_i, \kappa_i)\}) = \Big(\prod_{i=1}^{n} \lambda(t_i)\gamma(\kappa_i|t_i)\Big) \exp\big(-\Lambda(T)\big)$$

where

$$\Lambda(t) = \int_0^t \lambda(s)ds$$

(cf. Daley and Vere-Jones, 2003 ; Prop. 7.3)

The E-step – III

The log-likelihood of the complete data (over $[0, T)$) is

$$\begin{aligned}
&\log f\big(\{t_i, \kappa_i\}, (u_i), \theta\big) \\
=&\Big(\sum_{i: u_i = 0} \log\big(\mu \gamma_I(\kappa_i | t_i)\big)\Big) - \mu T \\
&+ \sum_j \Big[\Big(\sum_{i: u_i = j} \log\big(\beta(t_i - t_j, \kappa_j)\gamma_O(\kappa_i | t_i)\big)\Big) - \int_{t_j}^{T} \beta(t - t_j, \kappa_j) dt\Big]
\end{aligned}$$

where

1. $\gamma_I(\kappa_i | t_i)$: (conditional) density function of marks of Immigrants
2. $\gamma_O(\kappa_i | t_i)$: (conditional) density function of marks of Offspring
3. $\theta = (\mu$, parameters of $\gamma_I, \gamma_O, \beta)$

The E-step IV

$$Q(\theta,\theta^{old}) = \int_{\mathcal{M}} \log f\big(\{t_i,\kappa_i\},(u_i),\theta\big) f\big((u_i)|\{t_i,\kappa_i\},\theta^{old}\big)\, d(u_i)$$

$$= \Big(\sum_i \mathbb{P}[u_i = 0] \log\big(\mu\gamma_I(\kappa_i|t_i)\big)\Big) - \mu T$$

$$+ \sum_{j\geq 1}\Big[\Big(\sum_{i\geq j+1} \mathbb{P}[u_i = j] \log\big(\beta(t_i - t_j,\kappa_j)\gamma_O(\kappa_i|t_i)\big)\Big) - \int_{t_j}^{T} \beta(t - t_j,\kappa_j) dt\Big]$$

The E-step IV

$$Q(\theta,\theta^{old}) = \int_{\mathcal{M}} \log f\left(\{t_i,\kappa_i\},(u_i),\theta\right) f\left((u_i)|\{t_i,\kappa_i\},\theta^{old}\right) d(u_i)$$

$$= \left(\sum_i \mathbb{P}[u_i = 0] \log\left(\mu\gamma_I(\kappa_i|t_i)\right)\right) - \mu T$$

$$+ \sum_{j\geq 1}\left[\left(\sum_{i\geq j+1} \mathbb{P}[u_i = j] \log\left(\beta(t_i - t_j,\kappa_j)\gamma_O(\kappa_i|t_i)\right)\right) - \int_{t_j}^{T} \beta(t - t_j,\kappa_j)dt\right]$$

where, for all $1 \leq j \leq i-1$

$$\mathbb{P}[u_i = j] \approx \mathbb{P}[u_i = j|\{t_i,\kappa_i\},\theta^{old}] = \frac{\beta(t_i - t_j,\kappa_j)}{\mu + \sum_{r=1}^{i-1}\beta(t_i - t_r,\kappa_r)}$$

and

$$\mathbb{P}[u_i = 0] \approx \mathbb{P}[u_i = 0|\{t_i,\kappa_i\},\theta^{old}] = 1 - \sum_{j=1}^{i-1}\mathbb{P}[u_i = j]$$

The M-step

Depending on the model for

1. the kernel β
2. the densities of the marks (Immigrants and Offspring) : γ_I, γ_O

we maximize

$$\theta \mapsto Q(\theta, \theta^{old})$$

The M-step

Depending on the model for

1. the kernel β
2. the densities of the marks (Immigrants and Offspring) : γ_I, γ_O

we maximize

$$\theta \mapsto Q(\theta, \theta^{old})$$

and then update

$$\theta^{new} \hookrightarrow \theta^{old}$$

Explicit M-step can be performed for the ETAS model with parameter $\theta = (\mu, K_0, a, b, \omega)$ (cf. Veen and Schoenberg).

Good parametrization of the Hawkes model for RFQ ?

ETAS :

$$\beta(\tau, \kappa) = \frac{K_0 \exp(a\kappa)}{(\tau + b)^{1+\omega}} I(\kappa > 0)$$

with Gutenber-Richter law for Immigrant events

$$\gamma_I(\kappa|t) = \frac{\beta e^{-\beta(\kappa-\kappa_0)}}{1 - e^{-\beta(\kappa_1-\kappa_0)}}$$

with $\kappa_1 = 8, \kappa_0 = 2$.

Good parametrization of the Hawkes model for RFQ ?

ETAS :

$$\beta(\tau, \kappa) = \frac{K_0 \exp(a\kappa)}{(\tau + b)^{1+\omega}} I(\kappa > 0)$$

with Gutenber-Richter law for Immigrant events

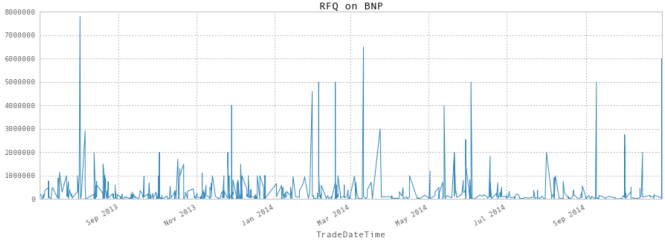
$$\gamma_I(\kappa|t) = \frac{\beta e^{-\beta(\kappa-\kappa_0)}}{1 - e^{-\beta(\kappa_1-\kappa_0)}}$$

with $\kappa_1 = 8, \kappa_0 = 2$.

Exponential kernel : cf. ' "Market making" in an order book model and its impact on the spread' by Ioane Muni Toke

$$\beta(\tau, \kappa) = \alpha e^{-\beta\tau}$$

Example



1. Jump 1922 has [1923, 1924] for followers
2. Jump 1924 has [1925, 1926, 1927, 1928, 1929, 1930, 1931, 1932, 1933, 1934, 1935, 1936] for followers
3. Jump 1936 has [1937, 1938, 1939, 1940] for followers

there are 371 leaders among the 5000 jumps

Performance evaluation of recommendation systems

Risk measure

Performance measures are used to fit unknown parameters via **Cross-validation** :

1. dimension d of the Latent space
2. number k of clusters
3. parameters of the Hawkes process that are too hard to find by optimization

Two types of mistakes :

first We don't want to recommend a bond to a client who does not want it : False Positive

second We don't want to miss a recommendation : the client buy a bond but we did not recommend it : False Negative

Risk measure

Performance measures are used to fit unknown parameters via **Cross-validation** :

1. dimension d of the Latent space
2. number k of clusters
3. parameters of the Hawkes process that are too hard to find by optimization

Two types of mistakes :

first We don't want to recommend a bond to a client who does not want it : False Positive

second We don't want to miss a recommendation : the client buy a bond but we did not recommend it : False Negative

Two types of success :

first We recommend a bond that is bought by the client : True Positive

second We don't recommend a bond that is not bought by the client : True Negative

AUC criterium

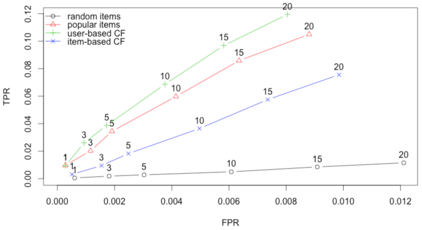


Figure : ROC curves for different Recommendation systems

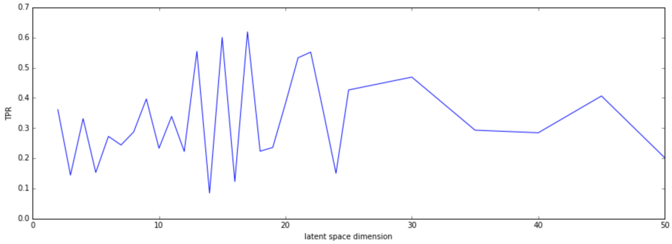
$$TPR = \text{True Positive Rate} = \frac{\text{Nb of True Positive } (= \text{Nb of Good reco})}{\text{Nb of Positive } (= \text{Nb of reco})}$$

$$FPR = \text{False Positive rate} = \frac{\text{Nb of False Positive } (= \text{Nb of wrong reco})}{\text{Nb of Negative } (= \text{Nb of items - Nb of reco})}$$

For every $t \in \{0, \dots, 936\}$ ($t =$Nb of recommendations), we compute a point $(FPR(t), TPR(t))$. Then, we compute the area under the curve = AUC (the larger the better).

Example

1. keep the 100 closest users to BNPP (renormalized) in the latent space
2. run the leader / follower algorithm on the RFQs of these users (on BNPP)
3. make 10 recommendations for each detected leader and count the number of TPR.



Thanks !