# Approximate filtering of intensity process for Poisson count data

Naratip Santitissadeekorn[1] , David Lloyd[1] , Martin Short[2], Sylvain Delahaise[1]

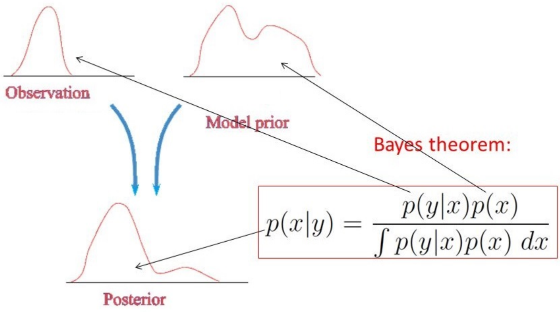
[1]Department of Mathematics
University of Surrey

[2]Mathematics Department
Georgia Institute of Technology

BIRS Mathematical criminology and security

# Bayesian inference

## Data assimilation: general formulation




$$p(x|y) = \frac{p(y|x)p(x)}{\int p(y|x)p(x)\, dx}$$

## Sequential DA (in general)

- $u_t$: unobserved states (e.g. crime rate, unknown parameters)
- $y_t$: Observations (e.g. $N$ events per day or Times of events)
- Try to recursively estimate $P(u_t|y_{1:t})$
- One-step-ahead prediction for $u_t$

$$P(u_t|y_{1:t-1}) = \int P(u_t|u_{t-1})P(u_{t-1}|y_{1:t-1})du_{t-1}$$

- One-step-ahead prediction for $y_t$

$$P(y_t|y_{1:t-1}) = \int P(y_t|u_t)P(u_t|y_{1:t-1})du_t$$

- Filtering

$$P(u_t|y_{1:t}) = \frac{P(y_t|u_t)P(u_t|y_{1:t-1})}{P(y_t|y_{1:t-1})}$$

- Kalman filter: linear model+Gaussian

## Dealing with Nonlinearity: Particle Filtering (PF)

**PF**: Empirical approximation of $P(u_t|y_{1:t}) \approx \{w_t^{(i)}, u_t^{(i)}\}$

- Use Important-sampling resampling to update the particle weight $w_t^{(i)}$

$$w_t^{(i)} \propto \frac{P(y_t|u_t^{(i)})P(u_t^{(i)}|u_{t-1}^{(i)})}{q(u_t^{(i)}|u_{1:t-1}^{(i)}, y_{1:t})} w_{t-1}^{(i)}$$

  where $u_t^{(i)} \sim q(u_t^{(i)}|u_{1:t-1}^{(i)}, y_{1:t}) \rightarrow$“proposal density"
- Resampling: select particles with informative weights, aiming to mitigate the weight degeneracy.
- Curse of dimensionality

## Conditional intensity Poisson process

- **Poisson intensity process**: the instantaneous conditional probability of an event satisfies

$$\lim_{\delta t \to 0} \frac{Pr(\Delta N_t = 1 | \lambda(\tau), t < \tau < t + \delta t)}{\delta t} = \lambda(t),$$

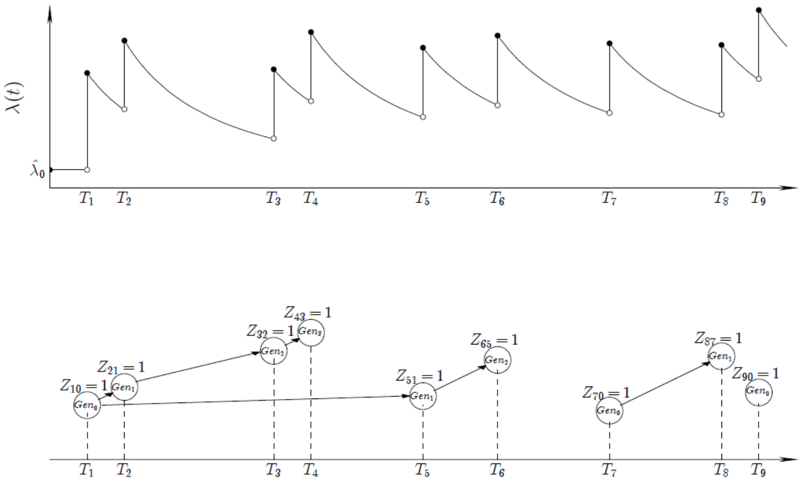
where $\Delta N_t = N_{t+\delta t} - N_t$ and $\lambda(t)$ is a stochastic intensity function [Cox and Lewis 1966, Ch. 7]

- **Hawkes process**: e.g. exponential decay

$$\lambda(t) = \mu + k \sum_{t_i < t} \exp\{-\omega(t - t_i)\}.$$

  - Self-exciting rate $k$, Decay rate $\omega$, Baseline rate $\mu$
  - $t_i$ the time of the $i$-th random event
  - Branching with immigration

# Self-excitation



# Parameter estimation

- **MLE**:
  - Need a large enough data set
  - Assume static parameters (what if parameters change?)
  - Multiple local maxima
  - Expectation Maximization (EM) is very popular for Hawkes process
- **Filtering**: "parameters" are stochastically-varying
  - If “true" parameters are static, the filter should converge the true parameters
  - Step change in parameters: the filter should adapt to the new parameters (at a learning rate)
  - Poor model: adaptive parameter may perform “better" than static ones?

# Approximate filter

- Motivated by extended Kalman filter (ExKF)
- **Extended Poisson-Kalman filter (ExPKF)**: “state-space" model for parameter tracking

**Prediction** $u(t) = F(u(t-1)) + \xi(t)$, $F$ is linear

**Likelihood** $y(t) \sim Poi(\lambda(t)) \quad \lambda(t) = g(u(t), y_{1:t-1})$

**Approx** “second-order approximation" $Poi(\lambda(t))$

# ExPKF

- Updating mean and covariance of $u_t$

$$\mathbf{P}_k^{-1} = \mathbf{P}_{k|k-1}^{-1} + \sum_{j=1}^{C} \Bigg[ \left( \frac{\partial \log \lambda_k^j}{\partial u_k} \right) \left( \frac{\partial \log \lambda_k^j}{\partial u_k} \right)^T \lambda_k^j \Delta t_k - (\Delta N_k^j - \lambda_k^j \Delta t_k) \frac{\partial^2 \log \lambda_k^j}{\partial u_k^2} \Bigg]$$

$$\bar{u}_k = \bar{u}_{k|k-1} + \mathbf{P}_k \sum_{j=1}^{C} \Bigg[ \left( \frac{\partial \log \lambda_k^j}{\partial u_k} \right) (\Delta N_k^j - \lambda_k^j \Delta t_k) \Bigg],$$

- Derivatives are evaluated at the forecast mean $\bar{\lambda}_{k|k-1}^j$
- If the Hessian term is an outer product, the rank-1 update can be used (very fast and efficient!)

- **Outer product form**

$$\left(\frac{\partial \log \lambda_k^j}{\partial \theta_k}\right)\left(\frac{\partial \log \lambda_k^j}{\partial \theta_k}\right)^T = -\frac{\partial^2 \log \lambda_k^j}{\partial \theta_k^2}.$$

- **Simplification**

$$\mathbf{P}_k^{-1} = \mathbf{P}_{k|k-1}^{-1} + \sum_{j=1}^{m} h_j h_j^T.$$

- $h_j = \sqrt{\Delta N_k^j}\left(\frac{\partial \log \lambda_k^j}{\partial \theta_k}\right)$.
- **Rank-1 update:**$\mathbf{B} = \mathbf{A} + hh^T$,

$$\mathbf{B}^{-1} = \mathbf{A}^{-1} - \frac{A^{-1} hh^T \mathbf{A}^{-1}}{1 + h_1^T \mathbf{A}^{-1} h_1}.$$

- $\mathbf{H}_0 = \mathbf{P}_{k|k-1}^{-1}, \mathbf{H}_1 = \mathbf{H}_0 + h_1 h_1^T, \mathbf{H}_2 = \mathbf{H}_1 + h_2 h_2^T, \ldots, \mathbf{H}_m = \mathbf{P}_k^{-1}$, we can efficiently compute $\mathbf{P}_k$.

## Example: EnPKF for Hawkes process

- A discrete-time Hawkes model on $m$ lattice nodes

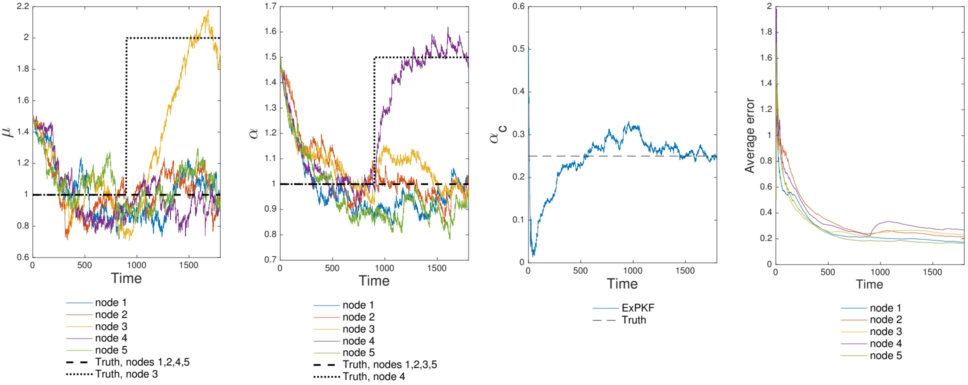
$$\lambda_j(k+1) = \mu_j + (1-\beta_j\delta t)(\lambda_j(k)-\mu_j) + \alpha_j \Delta N_j(k) + \sum_{j' \in \mathcal{N}(j)} \alpha_{j,j'} \Delta N_{j'}(k)$$

- **Simplification**:
  - $\alpha_{j,j'} = \alpha_c \rightarrow$ reducing complexity
  - known $\beta_j \rightarrow$ allowing the rank-1 update
- **Equivalent form**:

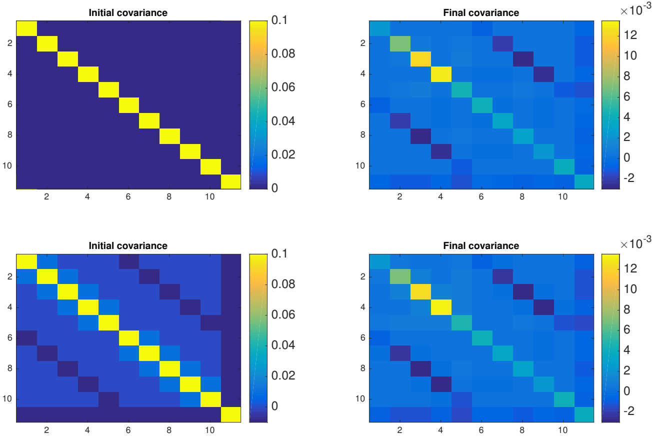
$$\lambda_j(k) = \mu_j + \sum_{\ell=1}^{k-1} \gamma^{k-\ell-1} \alpha_j \Delta N_j(k) + \sum_{\ell=1}^{k-1} \sum_{j' \in \mathcal{N}(j)} \gamma^{k-\ell-1} \alpha_c \Delta N_{j'}(k)$$

- $\gamma \equiv (1 - \beta\delta t)$.
- Interested in the scenario of the step change of parameters

# Result: Multi-cell Hawkes

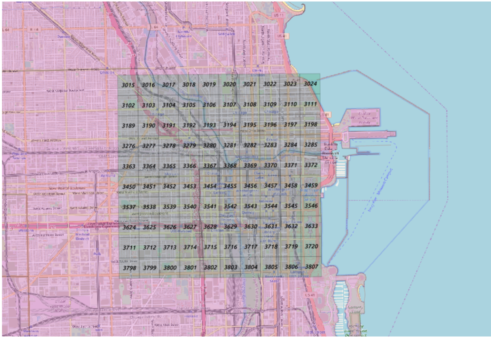


# Result: Multi-cell Hawkes



## Chicago Crime data

- https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present-Map/c4ep-ee5m
- robberies, burglaries, criminal damages and thefts from 1 January 2012 till 31 December 2017

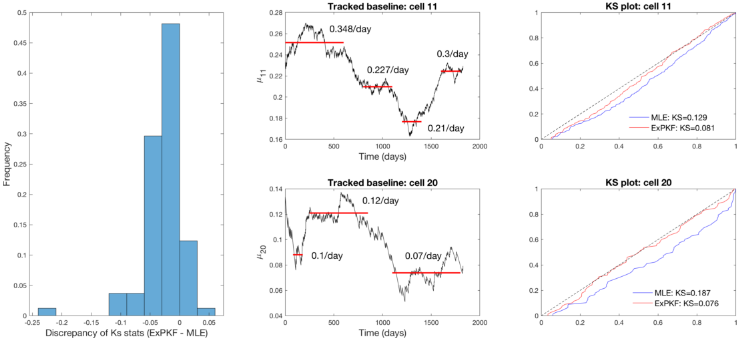


# Chicago Crime data

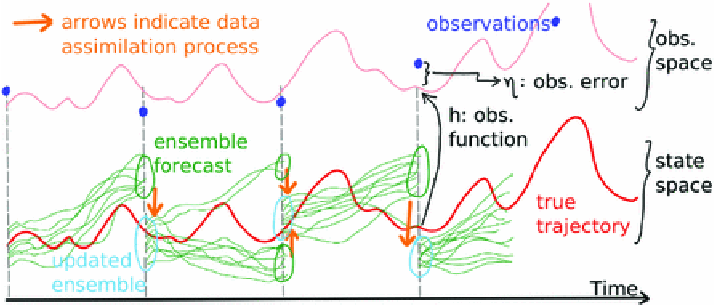
- Goodness-of-fit: Kolomogrov-Smirnov (KS) test

$$z_i = 1 - \exp(-\int_{t_i - 1}^{t_i} \lambda(t)dt),$$

- Good fit $\rightarrow z_i$ would lie close to the the 45 degree line on the unit box



# Ensemble filter: equal weight particles



# Ensemble filter

- Ensemble version of ExPKF is difficult to develop
- Motivated by GIG-filter [Bishop 17], use the conjugate pair

**Likelihood** $\Delta N(t) \sim Poi(\lambda(t))$
**Prior** $\lambda \sim Ga(a(t), b(t))$
**Posterior** $\lambda \sim Ga(a(t) + \Delta N(t), b(t) + 1)$

- Ensemble Poisso-Gamma filter: Update ensemble mean and relative variance $P_r = P/\langle\lambda\rangle^2$ ($P$ is variance)

$$\langle\lambda^a\rangle = \langle\lambda\rangle + \frac{\langle\lambda\rangle}{P_r^{-1} + \langle\lambda\rangle\delta t}(y - \langle\lambda\rangle\delta t)$$

$$(P_r^a)^{-1} = P_r^{-1} + y$$

- Need a way to update ensemble to satisfy the above equation

## Ensemble filter: stochastic update

- For $y = 0$ everywhere, updating only the ensemble mean
- For $y > 0$, updating each ensemble member by

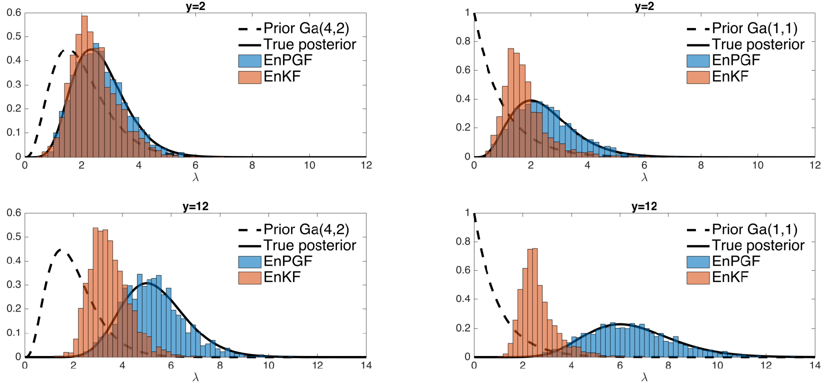
$$\frac{\lambda^{(i),a} - \bar{\lambda}^a}{\bar{\lambda}^a} = \frac{\lambda^{(i)} - \bar{\lambda}}{\bar{\lambda}} + P_r(P_r + (y)^{-1})^{-1}\left[\frac{\tilde{y}^{(i)} - \bar{\tilde{y}}}{\bar{\tilde{y}}} - \frac{\lambda^{(i)} - \bar{\lambda}}{\bar{\lambda}}\right],$$

  where $\tilde{y}^{(i)} \overset{iid}{\sim} Ga(y, 1)$ for $i = 1 \ldots, M$, see derivation in [NS,D.Lloyd,M.Short, CSDA 2018]
- Linear regression to update model parameters (similar to the serial-update version of EnKF)

## Test: EnPGF

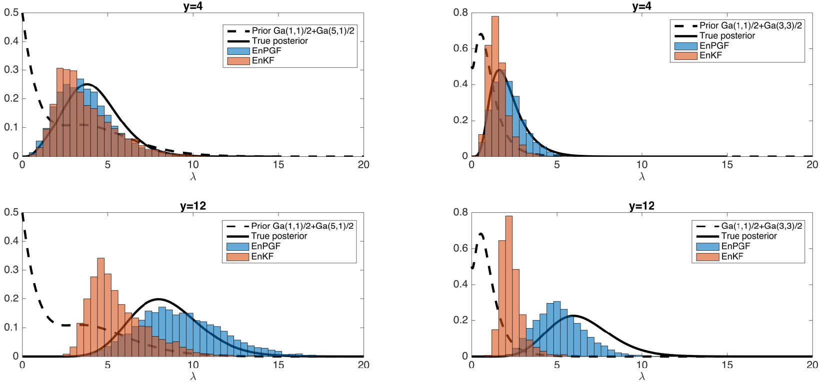
- Gamma Prior: EnPGF vs EnKF
- To avoid the homoskedasticity issue, apply the variance stabilizing transformation $z = \sqrt{y + 1/4} \sim N(\sqrt{\lambda}, 1/4)$
- Observation operator for EnKF

$$z = \sqrt{\lambda} + \eta, \qquad \eta \sim N(0, 1/4).$$



# Test: EnPGF

- Using mixture Gamma prior



# Test: Log-Gaussian Cox process (LGCP)

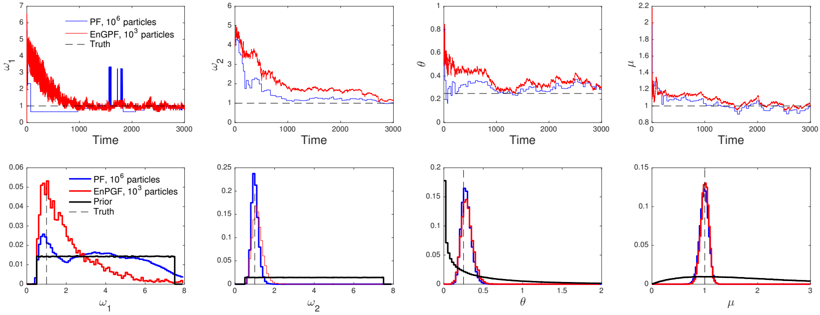
- Discretization of LGCP

$$
\begin{aligned}
x_{k+1} =& x_k - \omega_1(x_k - \mu)\delta t + \sigma\sqrt{\delta t} Z_k \qquad Z_k \sim N(0,1) \\
\lambda_{k+1} =& \exp(x_{k+1}) + (1 - \omega_2 \delta t)(\lambda_k - \exp(x_k)) + \theta y_k,
\end{aligned}
$$

- $v_k = [\lambda_k, x_k, \mu, \omega_1, \omega_2, \theta]$
- Difficult to apply ExPKF
- Difficult to estimate/track $\sigma$ (same issue as EnKF)
- Metropolis Adjusted Langavin algorithm (MALA) was used for this problem in [Mohler 14]

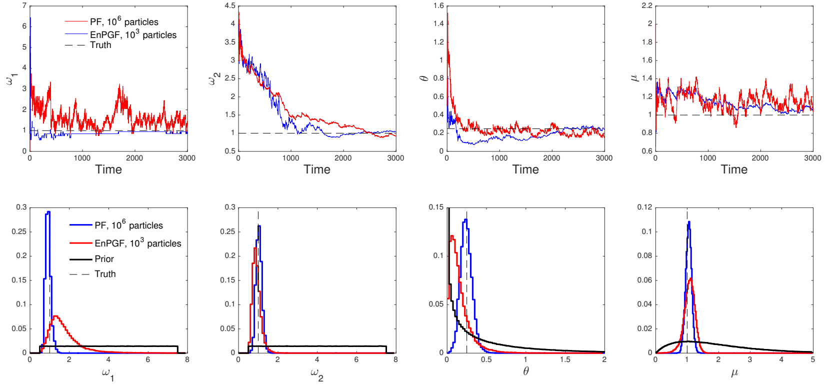
# Test: Log-Gaussian Cox process (LGCP)

- Tracked parameters: Small noise



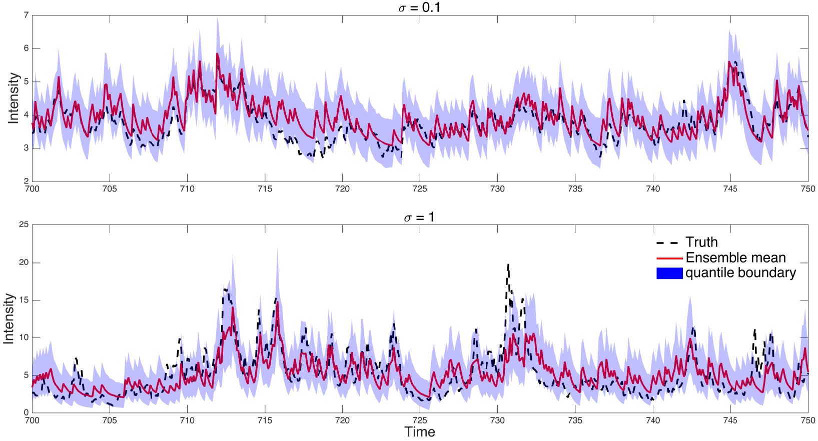
# Test: Log-Gaussian Cox process (LGCP)

- Tracked parameters: Large noise



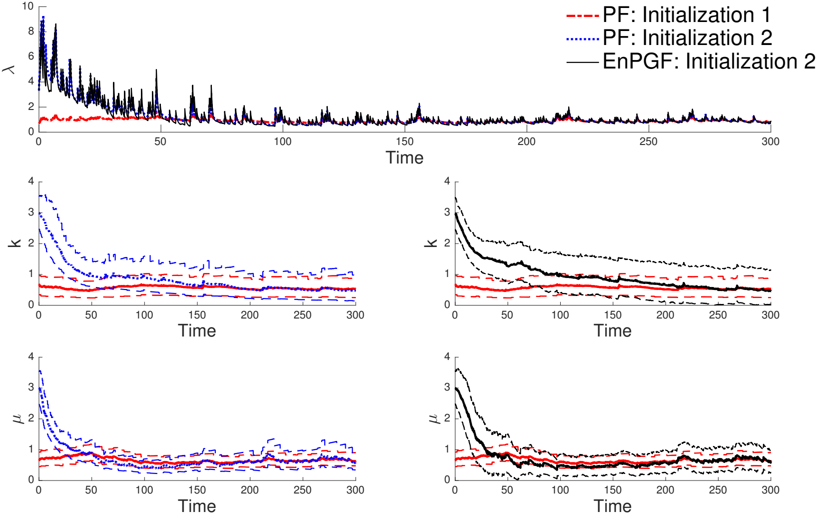
# Test: Log-Gaussian Cox process (LGCP)

- Tracked intensity



# LA crime data (Holenbeck area)

- A time-series of the time of occurences

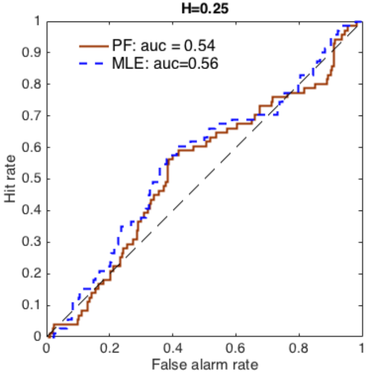


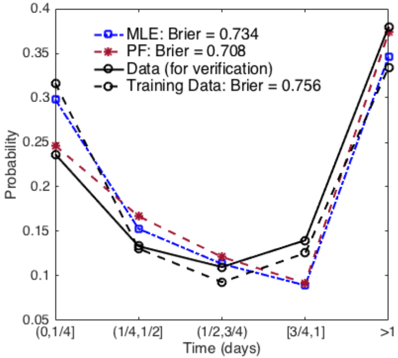
## LA crime data (Holenbeck area)

- Brier score: $Y_i \in \{0, 1\}$

$$B = N^{-1} \sum_{i=1}^{N} (\pi_i - y_i)^2$$

- $\pi_i$ is the forecast probability for $y_i = 1$ at the verification point



# Conclusion

- **Algorithm:**

  **ExPKF** fast and efficient (if rank-1 update is possible), multivariate

  **EnPGF** Derivative free, univariate, extension to multivariate is possible (but can be tricky)

- **Future work:**
  - Hawkes model on a network
  - Applying ExPKF/EnPGF to other applications (with Poisson count data)
  - Developing different filters for different data types (e.g. proportional data)