Earthquake Forecasting

Ensemble Methods for Merging Models

Alexander K. Christensen

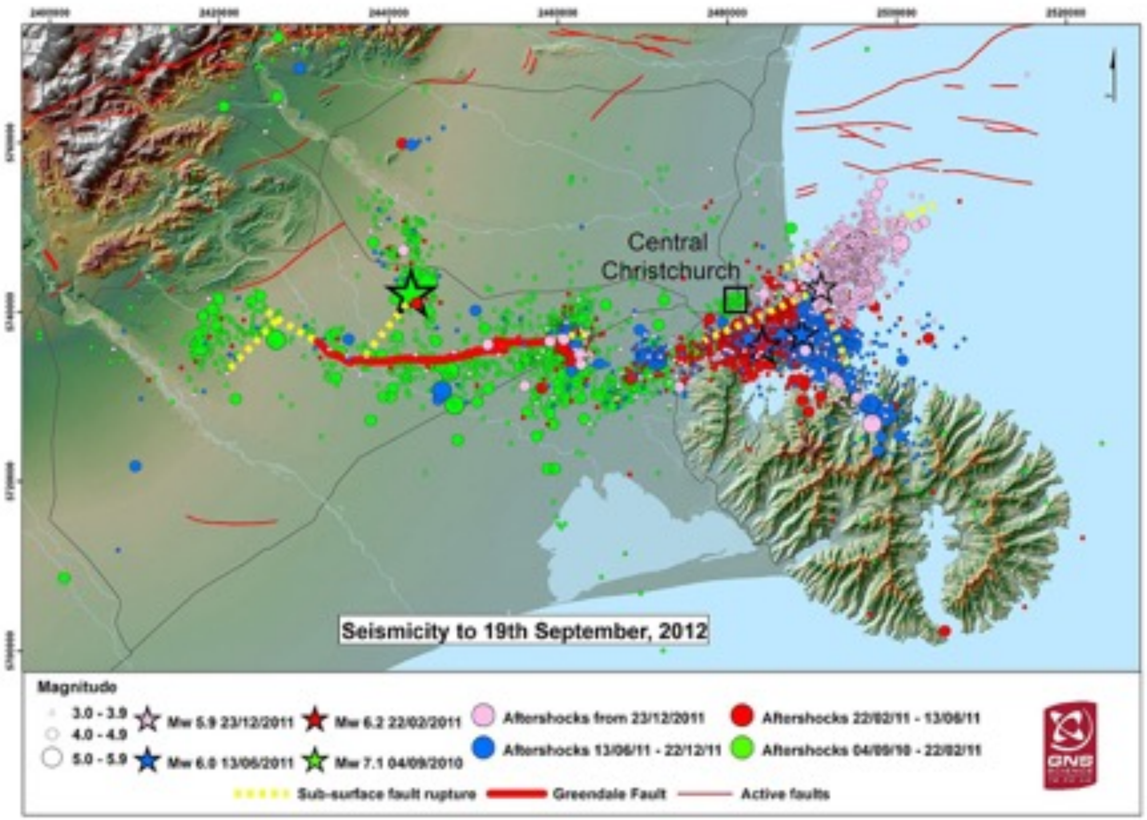
Dr. Maximilian J. Werner

Motivation

- What *is* ‘forecasting’?
- What can we actually forecast?
- What can we do with this?
- We developed a new strategy for combining models.

Canterbury Earthquake Sequence, NZ

- CSEP
- Sept 2010 – Dec 2011
- Why this Sequence?
  - Complex but well-documented series
  - Very destructive
  - Significant aftershocks
- Our dataset begins 1s after Darfield M7.1 event.



Experiment Design

- Produce a forecast
  - Input: "real-time" vs "best available" data
  - Output: Poisson rates

- How we test a model
  - Catalogue of # eqks
    per 0.05 x 0.05° region
    per 0.1 magnitude step
    per month

  - Likelihood: Model M, Data $D = \{D_1, \dots, D_n\}$ $L_M(D) = \prod_{i=1}^{n} P_M(D_i = x_i)$

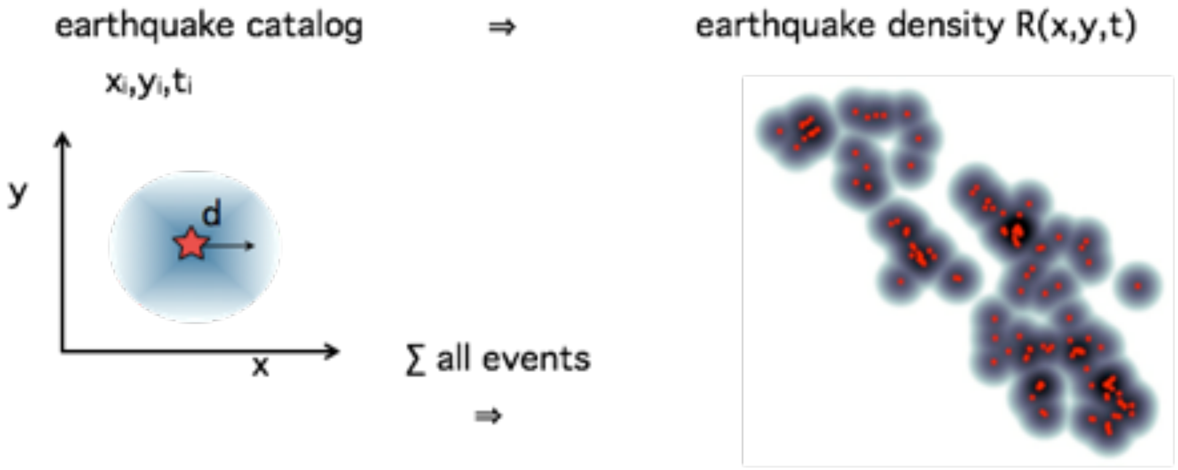
- How we compare models
  - Compare Likelihoods

  - Probability Gain $PG(M,N) = exp\left(\frac{LogL_M(D) - LogL_N(D)}{\#\ eqks}\right)$

Base Models

- 3 types

  1. Physical – e.g. stress modelling
  2. Statistical – e.g. smoothing/clustering
  3. Hybrids



- Our portfolio:

  - 5 physical
  - 6 statistical
  - 4 hybrid

  - 15 total

Model Ensembling

- Merge forecasts from different models
  - Weighted average
  - Advantages over 'Model Selection'

- Challenge is finding the right weights!

- Recent paper [Roades et al.] advocates multiplicative over additive averaging:

"The information gains of the best multiplicative ensembles are greater than those of additive ensembles constructed from the same models."

Additive: $F = \sum \omega_j F_j$ (F_j base models, ω_j weights)

Multiplicative: $F = k \prod F_j^{\omega_j}$ (k normalising constant)

Optimised Log-Linear Pooling

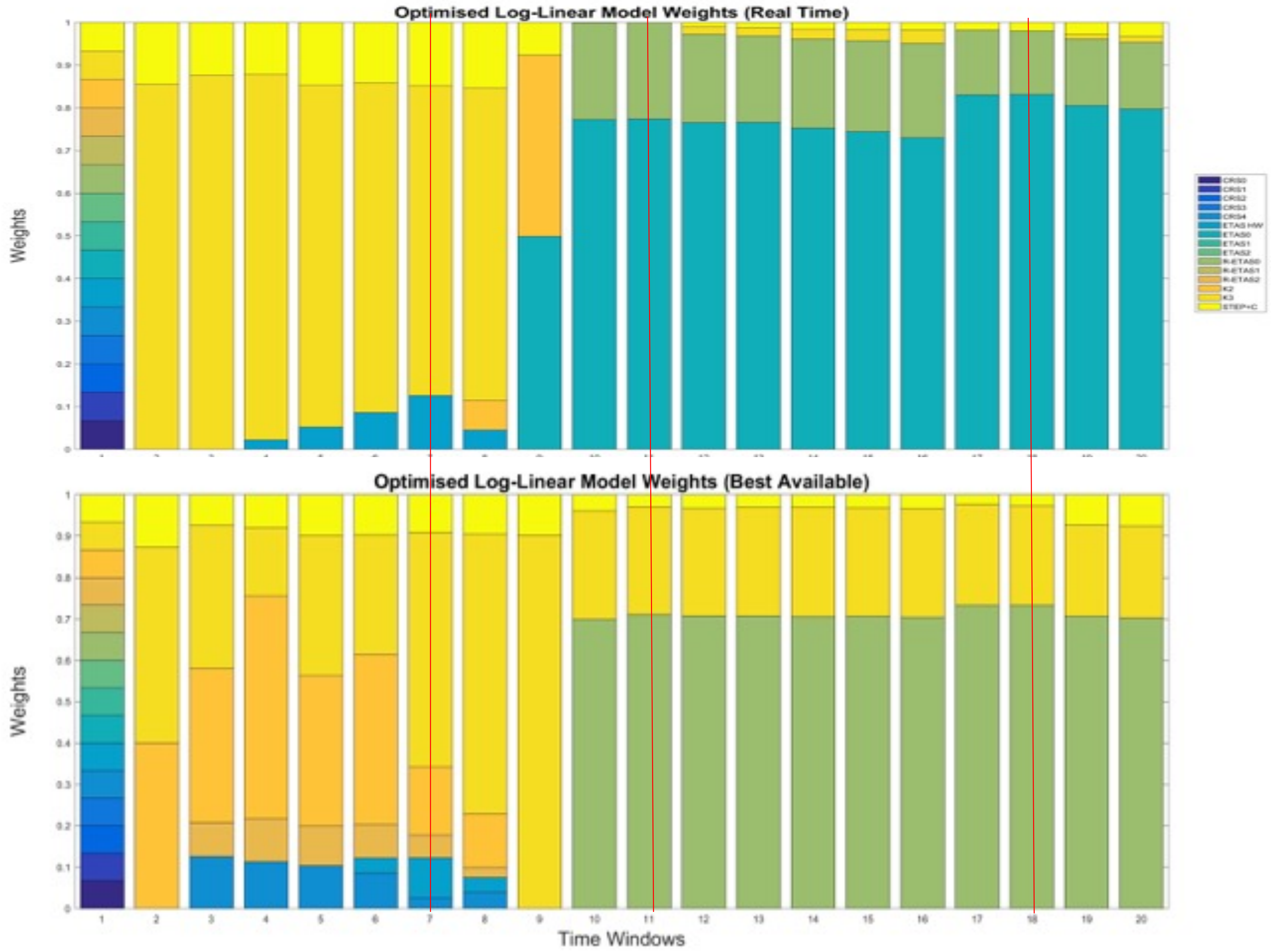
- Multiplicative ensemble
  - $F = \prod_{j=1}^{n} F_j^{\omega_j}$ $\leftrightarrow$ $logF = \sum_{j=1}^{n} \omega_j logF_j$
- How to choose weights?
  - What values of ω_j would have been 'best' up until now?
  - 'best' = greatest log-likelihood
  - Optimisation using `fmincon`

$$LogL = \sum_{i}^{(bins)} \sum_{s<t}^{(times)} \left[-\prod_{j}^{(models)} \lambda_{i,j,s}^{\omega_j} + n_{i,s} \left(\sum_{j}^{(models)} \omega_j log\lambda_{i,j,s} \right) - \log(n_{i,s}!) \right]$$

Existing Ensembles

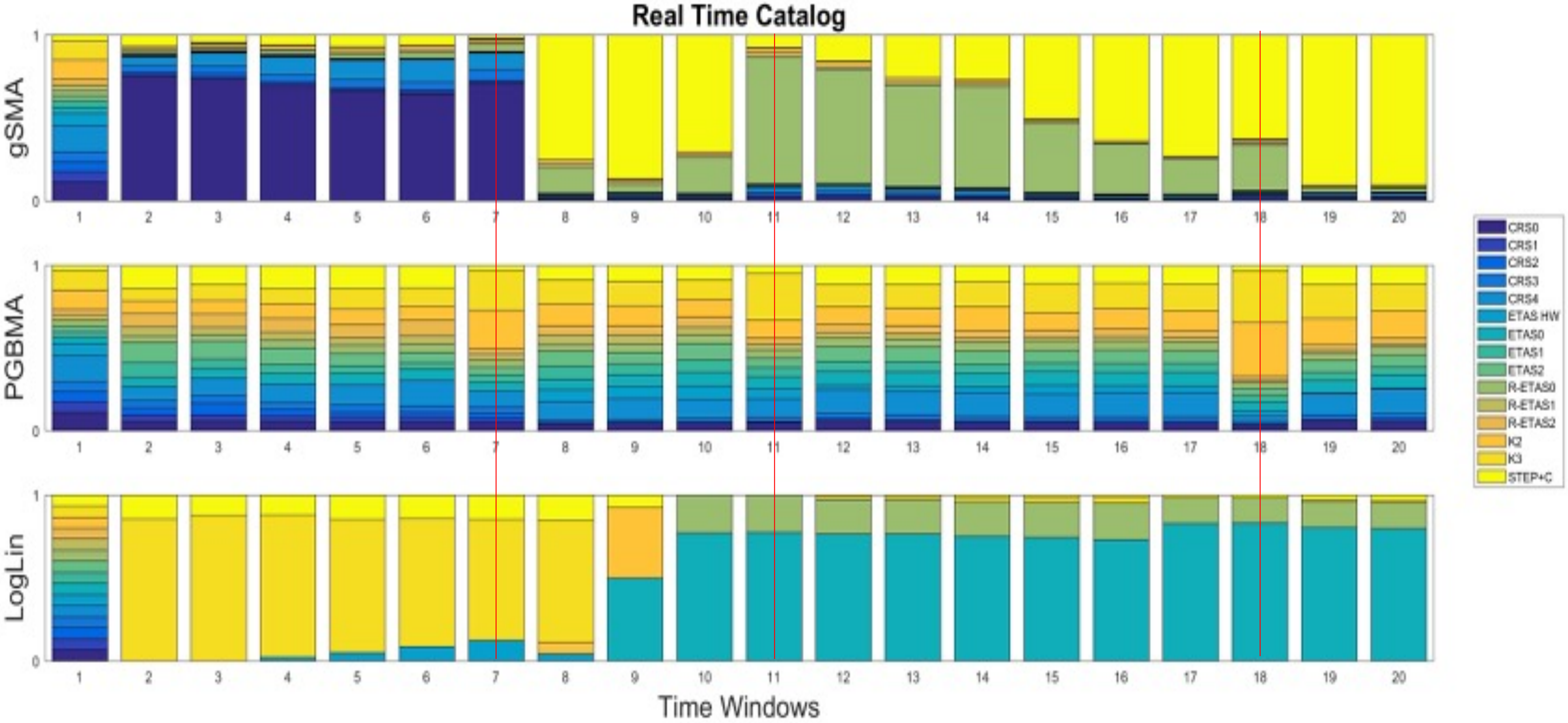
- Bayesian Model Averaging (BMA)
  - $\omega_i^{BMA} = L_i(D)$
- Score Model Averaging (SMA)
  - $\omega_i^{SMA} = \frac{1}{|LogL_i(D)|}$
- Generalised SMA (gSMA)
  - $\omega_i^{gSMA} = \frac{1}{|LogL_i(D) - LogL_0(D) + 1|}$
- Parimutuel Gambling SMA (PGSMA)
  - $\omega_i^{PGSMA} = 1 + \alpha V_i$

Results

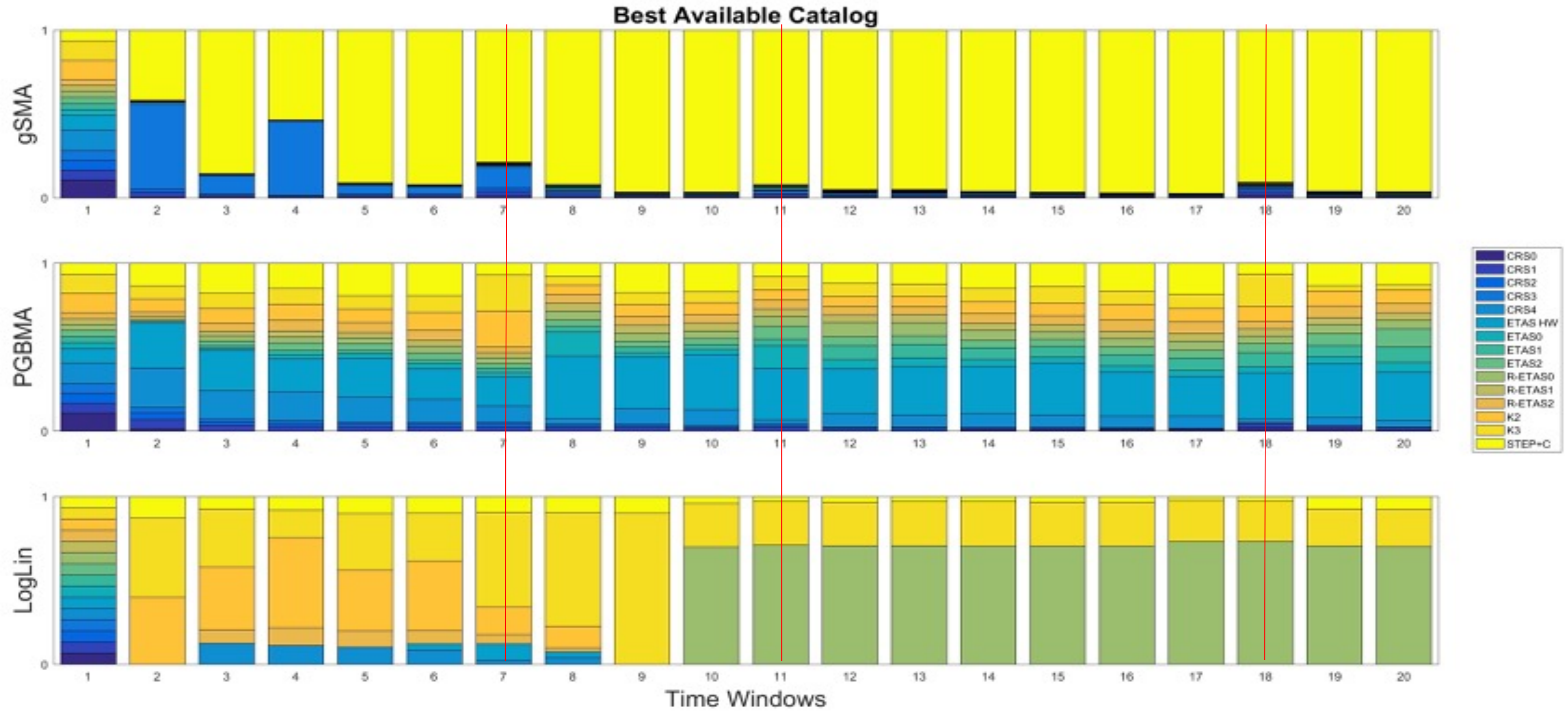


Optimised Log-Linear Model Weights (Real Time)
Weights
CRS0
CRS1
CRS2
CRS3
CRS4
ETAS HW
ETAS0
ETAS1
ETAS2
R-ETAS0
R-ETAS1
R-ETAS2
K2
K3
STEP+C
Optimised Log-Linear Model Weights (Best Available)
Weights
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Time Windows

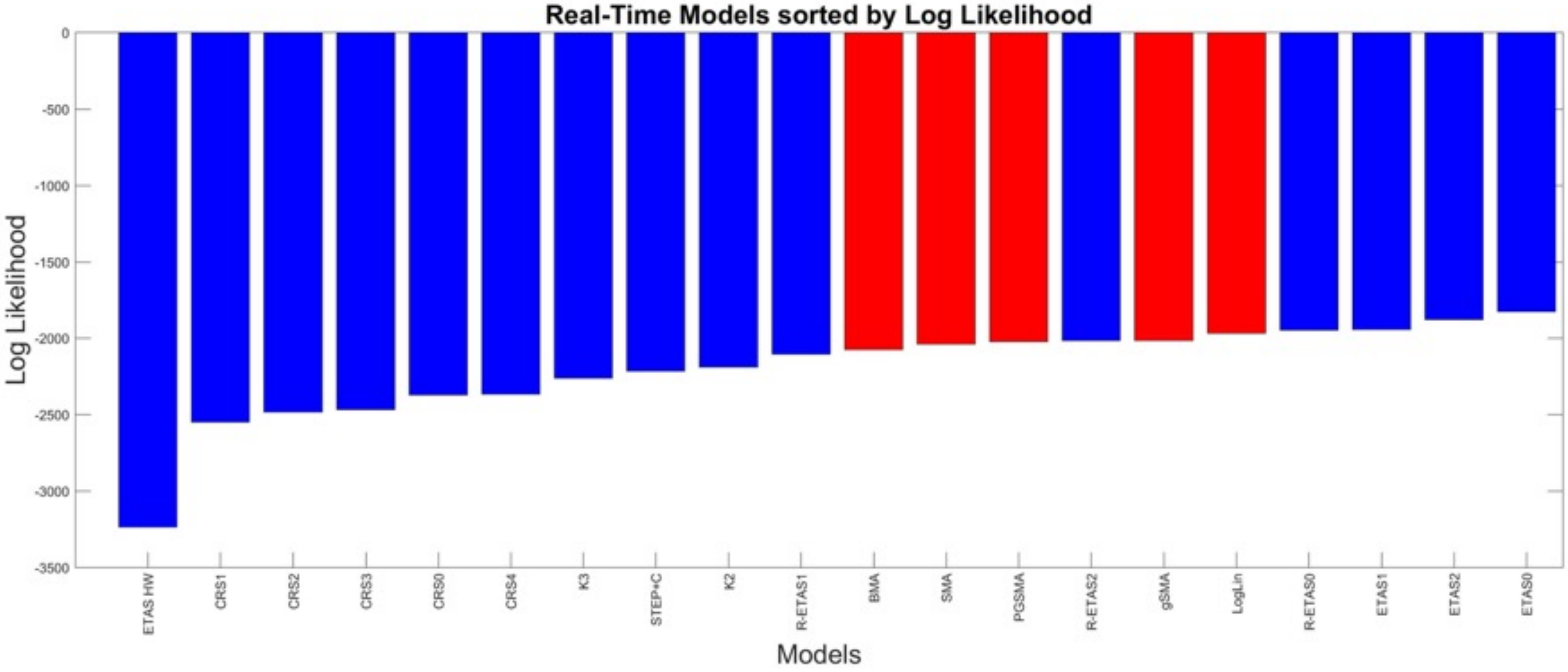
Weights from Existing Ensembles



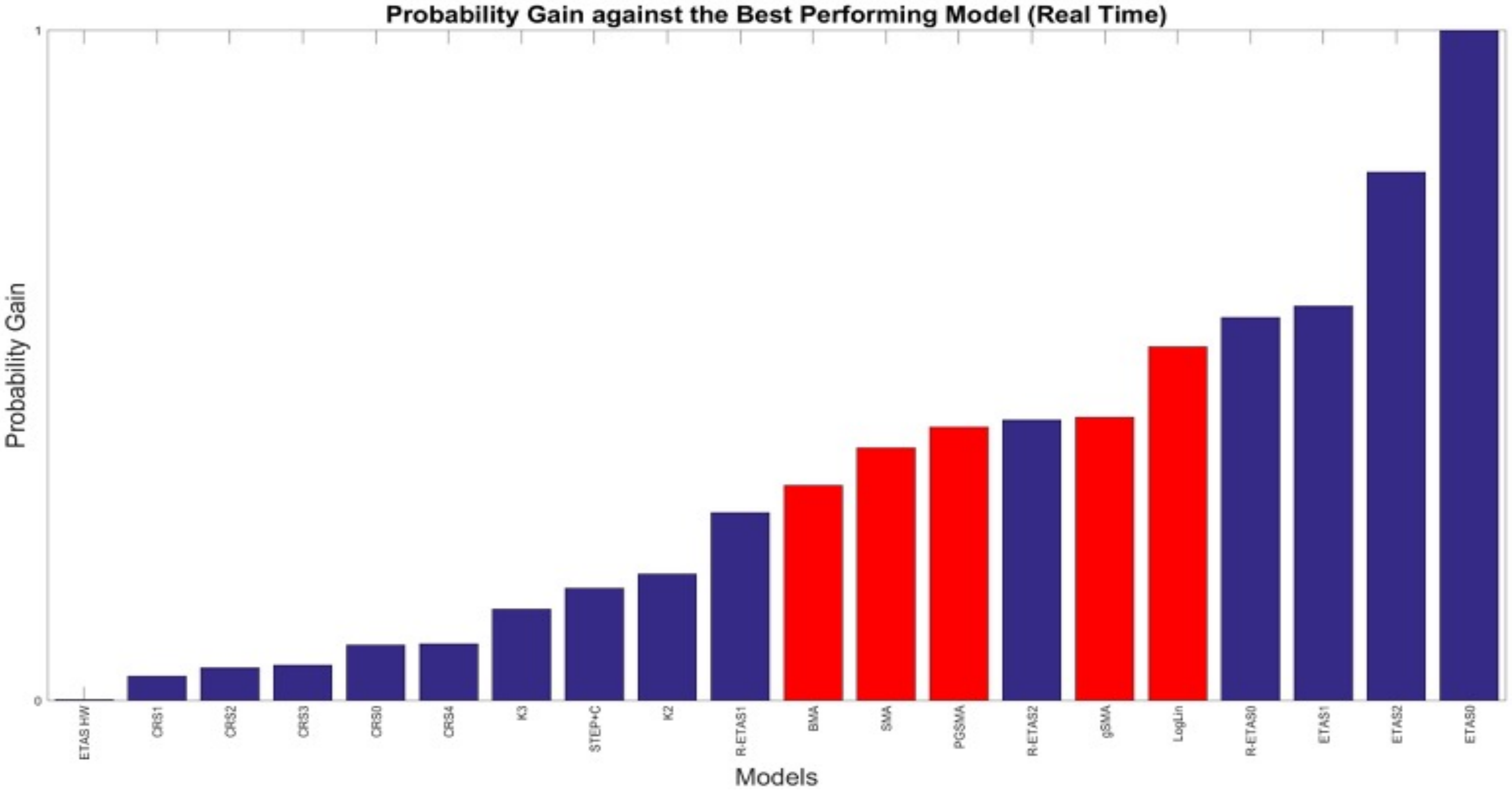
Comparison to Existing Ensembles



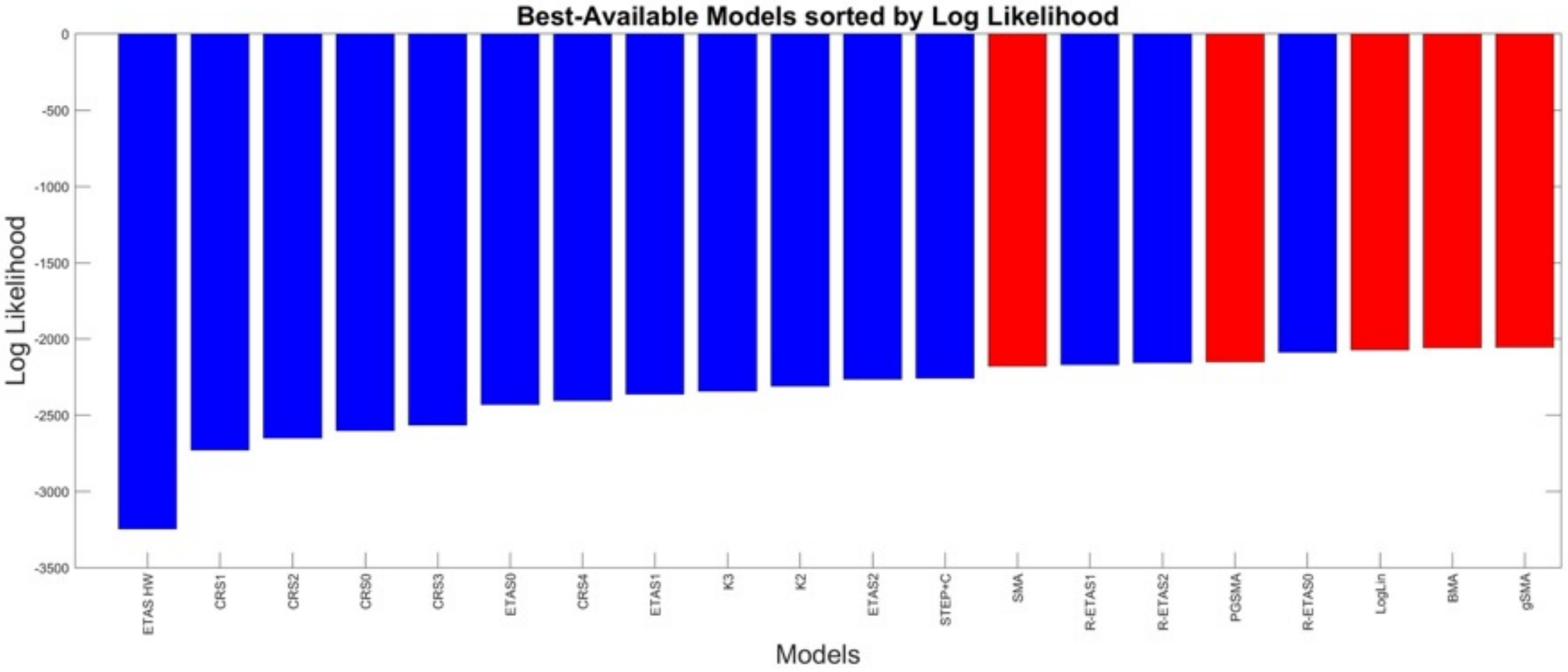
Performance Ranking



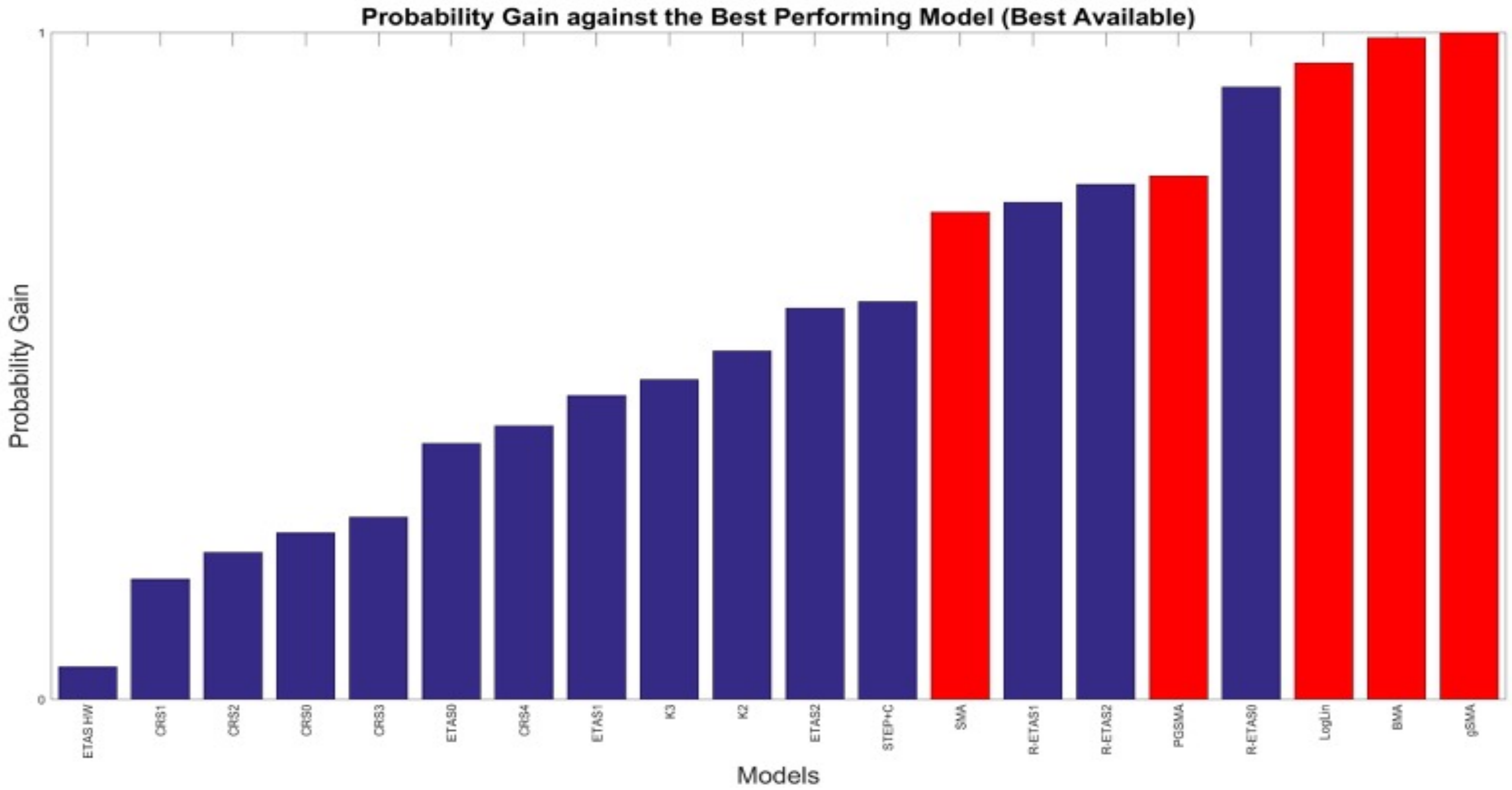
Performance Ranking



Performance Ranking



Performance Ranking

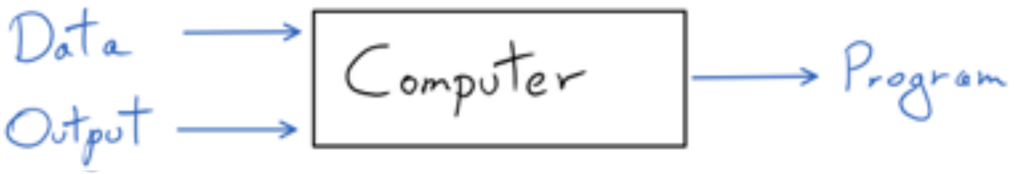


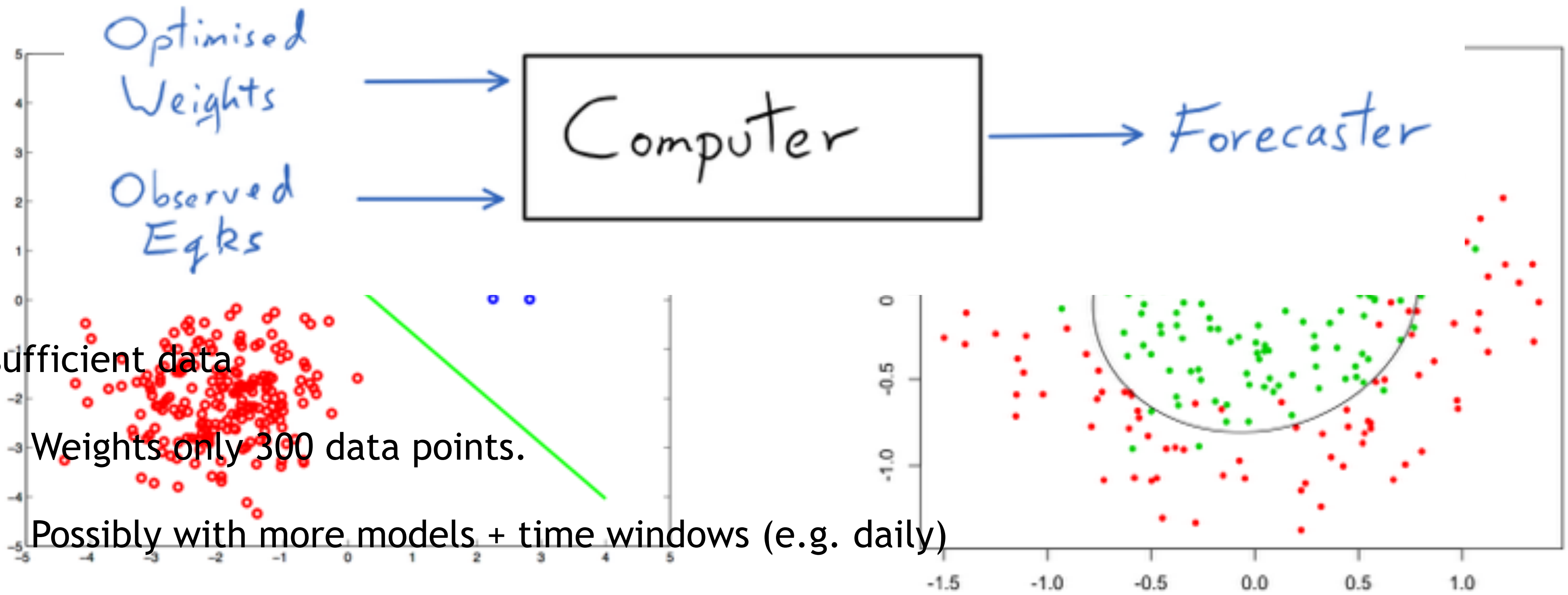
Discussion & Implications

- Multiplicative approach – verdict?
  - Competitive
  - First effort – could improve further!
  - Slower
  - Mustn’t overinterpret!

- Future directions
  - Other earthquake sequences
  - Deeper analysis

Machine Learning





- Insufficient data
  - Weights only 300 data points.
  - Possibly with more models + time windows (e.g. daily)

Conclusion

- Optimised Log-Linear Pooling is effective on this dataset
    - Merits further study/improvement
    - Other multiplicative approaches?

- Machine learning not yet appropriate
    - Need more data