Scalable and Robust Bayesian Inference via the Median Posterior

CS 584: Big Data Analytics

Material adapted from
David Dunson's talk (http://bayesian.org/sites/default/files/Dunson.pdf) &
Lizhen Lin's ICML talk (http://techtalks.tv/talks/scalable-and-robust-bayesian-inference-via-the-median-posterior/61140/)

Big Data Analytics

- Large (big N) and complex (big P with interactions) data are collected routinely

- Both speed & generality of data analysis methods are important

- Bayesian approaches offer an attractive general approach for modeling the complexity of big data

- Computational intractability of posterior sampling is a major impediment to application of flexible Bayesian methods

Existing Frequentist Approaches: The Positives

- Optimization-based approaches, such as ADMM or glmnet, are currently most popular for analyzing big data
- General and computationally efficient
- Used orders of magnitude more than Bayes methods
- Can exploit distributed & cloud computing platforms
- Can borrow some advantages of Bayes methods through penalties and regularization

Existing Frequentist Approaches: The Drawbacks

- Such optimization-based methods do not provide measure of uncertainty

- Uncertainty quantification is crucial for most applications

- Scalable penalization methods focus primarily on convex optimization — greatly limits scope and puts ceiling on performance

- For non-convex problems and data with complex structure, existing optimization algorithms can fail badly

Scalable Bayes Literature

- Number of posterior approximations have been proposed — expectation propagation, Laplace, variational approximations

- Variational methods are most successful in practice — recent thread on scalable algorithms for huge and streaming data

- Approaches provide an approximation to the full posterior but no theory on how good the approximation is

- Often underestimate the posterior variance and do not possess robustness

- Surprisingly good performance in many predictive applications not requiring posterior uncertainty

Efficient Implementations of MCMC

- Increasing literature on scaling up MCMC with various approaches

- One approach is to rely on GPUs to parallelize steps within an MCMC iteration (e.g., massively speed up time for updating latent variables specific to each data point)

    - GPU-based solutions cannot solve very big problems and time gain is limited by parallelization only within iterations

- Another approach is to accelerate bottles in calculating likelihoods and gradients in MCMC via stochastic approximation

MCMC and Divide-and-Conquer

- Divide-and-conquer strategy has been extensively used for big data in other contexts

- Bayesian computation on data subsets can enable tractable posterior sampling

- Posterior samples from data subsets are informatively combined depending on sampling model

- Limited to simple models such as Normal, Poisson, or binomial (see consensus MCMC of Scott et al., 2013)

Data Setting

- Corrupted with the presence of outliers
- Complex dependencies (interactions)
- Large size (doesn't fit on single machine)

https://www.hrbartender.com/wp-content/uploads/2012/11/Kronos-Thirsty-for-Data.jpg

Robust and Scalable Approach

- General: able to model complexity of big data and work with flexible nonparametric models
- Robust: robust to outliers and contaminations
- Scalable: computationally feasible

Attractive for Bayesian inference for big data

Basic Idea

- Each data subset can be used to obtain a noisy approximation to the full data posterior
    - Run MCMC, SMC, or your favorite algorithm on different computers for each subset
- Combine these noisy subset posteriors in a fast and clever way
- In the absence of outliers and model misspecification, the result is a good approximation to the true posterior

Two Fundamental Questions

- How to combine noisy estimates?
- How good is the approximation?
- Answer
  - Use notion of distance among probability distributions
  - Combine noisy subset posteriors through their median posterior
  - Working with subsets makes our approach scalable

Median Posterior

- Let $X_1, \ldots, X_N$ be i.i.d. draws from some distribution Π_0
- Divide data into R subsets ($U_1, \ldots, U_R$), each of size approximately N / R
- Update a prior measure with each data subset produces R subset posteriors $\Pi_1(\cdot \mid U_1), \cdots, \Pi_R(\cdot \mid U_R)$
- Median posterior is the geometric median of subset posteriors
  - One can think of geometric median as some generalized notion of median in general metric spaces

Geometric Median

- Define a metric space: $(\mathcal{M}, \rho)$ ← metric
  - set
- Example: Real space (set) and Euclidean distance (metric)
- Denote n points in the set as $p_1, \ldots, p_n$
- Geometric median of the n points (if it exists) is defined

$$p_M = \text{argmin}_{p \in \mathcal{M}} \sum_i \rho(p, p_i)$$

- For real line, this definition reduces to the usual median
- Can be applied in more complex spaces

Estimating Subset Posterior

- Run MCMC algorithms in an embarrassingly parallel manner for each subset

    - Independent MCMC chains for each data subset yields draws from subset posteriors for each machine

- Yields an atomic approximation to the subset posteriors

Median Posterior (3)

- View subset posteriors as elements in space of probability measures on parameter space
- Look for the ‘median’ of subset posterior measures
- Median posterior

distance between two probability measures

$$\Pi_M = \text{argmin}_{\Pi \in \Pi(\Theta)} \sum_r \rho(\Pi, \Pi(\cdot \mid U_r))$$

- Problem:
  - How to define distance metric?
  - How to efficiently compute median posterior?

Median Posterior (4)

Solution: Use Reproducing Kernel Hilbert Space (RKHS) after embedding the probability measures onto a Hilbert space via a reproducing kernel

- Computationally very convenient
- Allows accurate numerical approximation

Hilbert Space

- Generalizes the notion of Euclidean space to any finite or infinite number of dimensions
    - Fancy name for complete vector space with an inner product defined on space
- Can think of it as a linear inner product space (with several more additional mathematical niceties)
- Most practical computations in Hilbert spaces boil down to ordinary linear algebra

http://www.cs.columbia.edu/~risi/notes/tutorial6772.pdf

Kernel

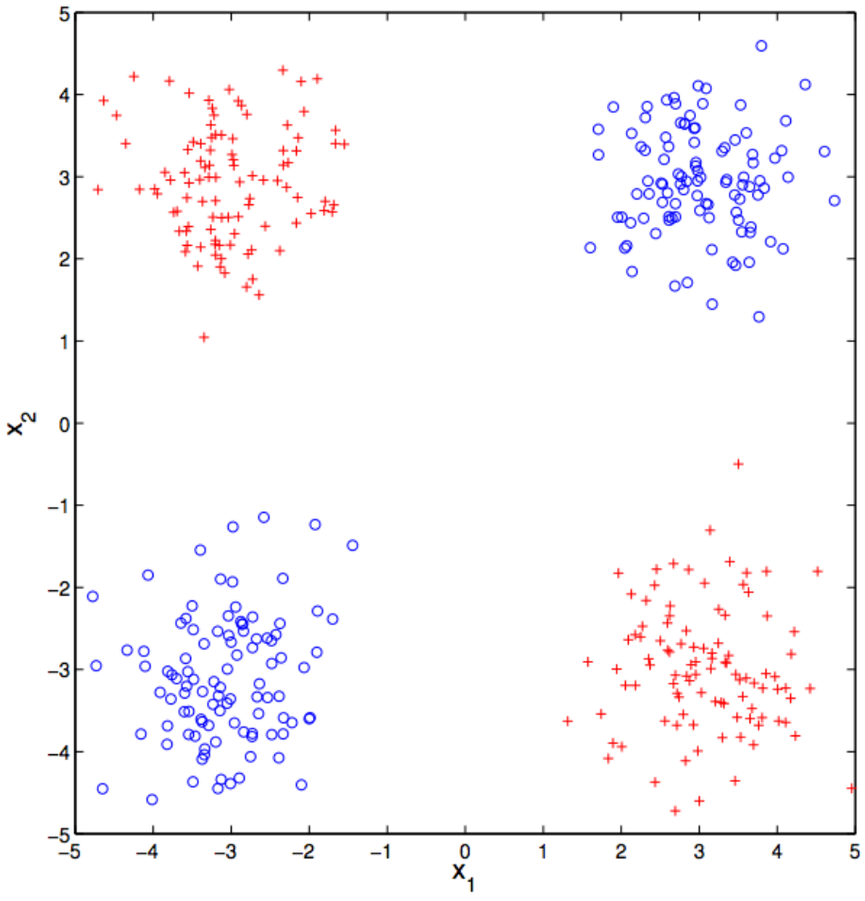
- Definition: Let X be a non-empty set. A function k is a kernel if there exists an R-Hilbert space and a map such that for all x, x' in X

$$k(x, x') =< \phi(x), \phi(x') >_H$$

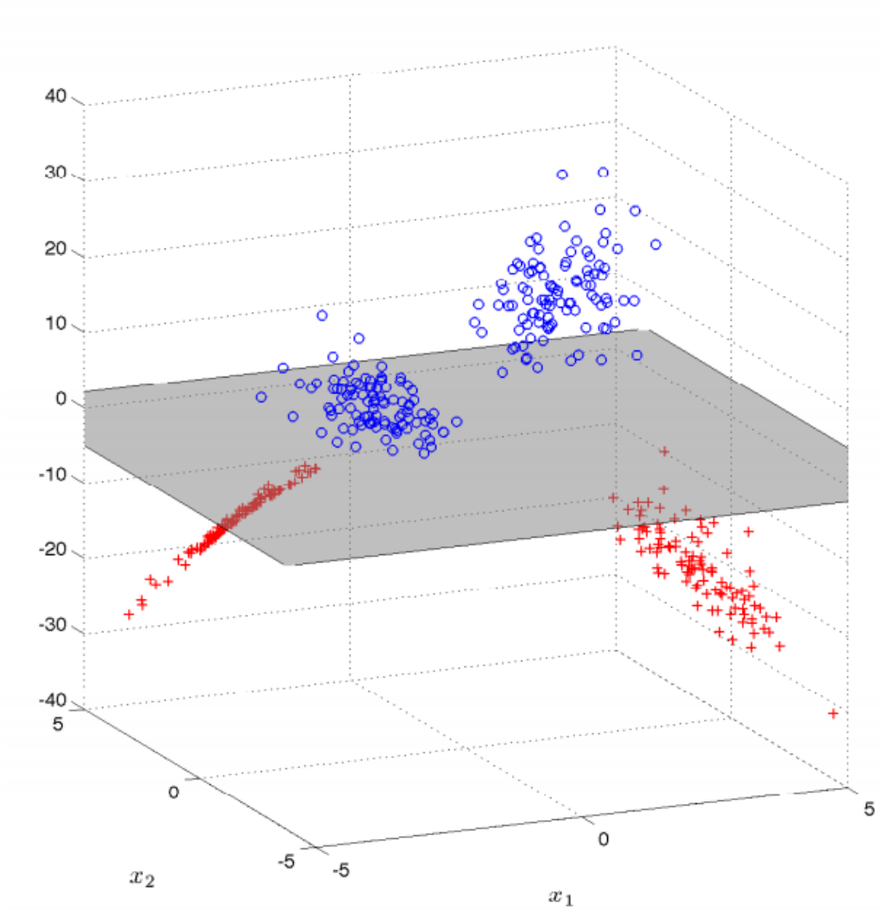
- A kernel give rise to a valid inner product (symmetric function) that is greater than or equal to 0

- Can think of it as a similarity measure

Kernels: XOR Example



$$\phi(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix}$$



No linear classifier separates red from blue

Map points to higher dimension feature space

http://www.gatsby.ucl.ac.uk/~gretton/coursefiles/Slides4A.pdf

Reproducing Kernel

A kernel is a reproducing kernel if it has two properties

- For every x_0 in X, k(y, x_0) as a function of y belongs to H (i.e., fix second variable to get function of first variable which should be a member of the Hilbert space)

- The reproducing property, for every x_0 in X and f in H,

$$f(x_0) =< f(\cdot), k(\cdot, x_0) >_H$$

  (i.e., pick any element from the set and a function from Hilbert space, then the inner product between these two should be equal to f(x_0))

Examples: Reproducing Kernels

- Linear kernel

$$k(x, x') = x \cdot x'$$

- Gaussian kernel

$$k(x, x') = e^{\frac{||x - x'||^2}{\sigma^2}}, \ \sigma > 0$$

- Polynomial kernel

$$k(x, x') = (x \cdot x' + 1)^2, d \in \mathbb{N}$$

Reproducing Kernel Hilbert Space

- A Hilbert space of complex-valued functions on a nonempty set X is RKHS if the evaluation functionals are bounded

$$|\mathcal{F}_t[f]| = |f(t)| \leq M||f||_H \forall f \in H$$

- RKHS if and only if it has a reproducing kernel

- Useful because you can evaluate functions at individual points

RKHS Distance

- A computationally "nice" distance by using a (RK) Hilbert space embedding $P \mapsto \int K(x, \cdot) P(dx))$

$$||P - Q||_{\mathcal{F}_x} = || \int_X k(x, \cdot) d(P - Q)(x)||_H$$

- P, Q empirical measures $P = \sum_{j=1}^{N_1} \beta_j \delta_{z_j}, \; Q = \sum_{j=1}^{N_2} \gamma_j \delta_{y_j}$

$$||P - Q||^2_{\mathcal{F}_k} = \sum_{i,j=1}^{N_1} \beta_i \beta_j k(z_i, z_j) +$$

$$\sum_{i,j=1}^{N_2} \gamma_i \gamma_j k(y_i, y_j) - 2 \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \beta_i \gamma_j k(z_i, y_j)$$

Calculate Geometric Median: Weiszfeld Algorithm

- Weiszfeld's algorithm is an iterative algorithm
- Initialize the point so you have equal weights and the estimate is the average of the posteriors
- Each iteration:
  - Update the weight $w_r^{(t+1)} = \frac{||Q_*^{(t)} - Q_r||_{\mathcal{F}_k}^{-1}}{\sum_{j=1}^{R} ||Q_*^{(t)} - Q_j||_{\mathcal{F}_k}^{-1}}$
  - Update your estimate $Q_*^{(t+1)} = \sum w_r^{(t+1)} Q_j$

Weiszfeld Algorithm: Practical Performance

- Advantages
  - Extremely stable iterations with provable global convergence
  - Simple implementation and easy extension for new data (ideal for big data)
  - Relatively insensitive to choice of Bandwidth parameter in RBF kernel (good for generic applications)
- Disadvantages:
  - Iterations can be slow if number of atoms across all subset posteriors are large (use SGD to avoid iterating through all atoms)
  - If all subset posteriors close to M-Posterior, Weiszfeld's weights are numerically unstable (use subset posterior as approximation)

Robustness of M-Posterior

- The median posterior can be proven to be robust which can handle gamma times R number of outliers of arbitrary nature for some appropriate constant, with R is the number of subsets

- Intuition for robustness - subset posteriors which contain the outliers contribute little to the median posterior calculation

Stochastic approximation for calibration

- Median posterior has higher variance compared to overall posterior

- Use stochastic approximation

- Idea: For each subset data, update the prior with a likelihood raised to the Rth power

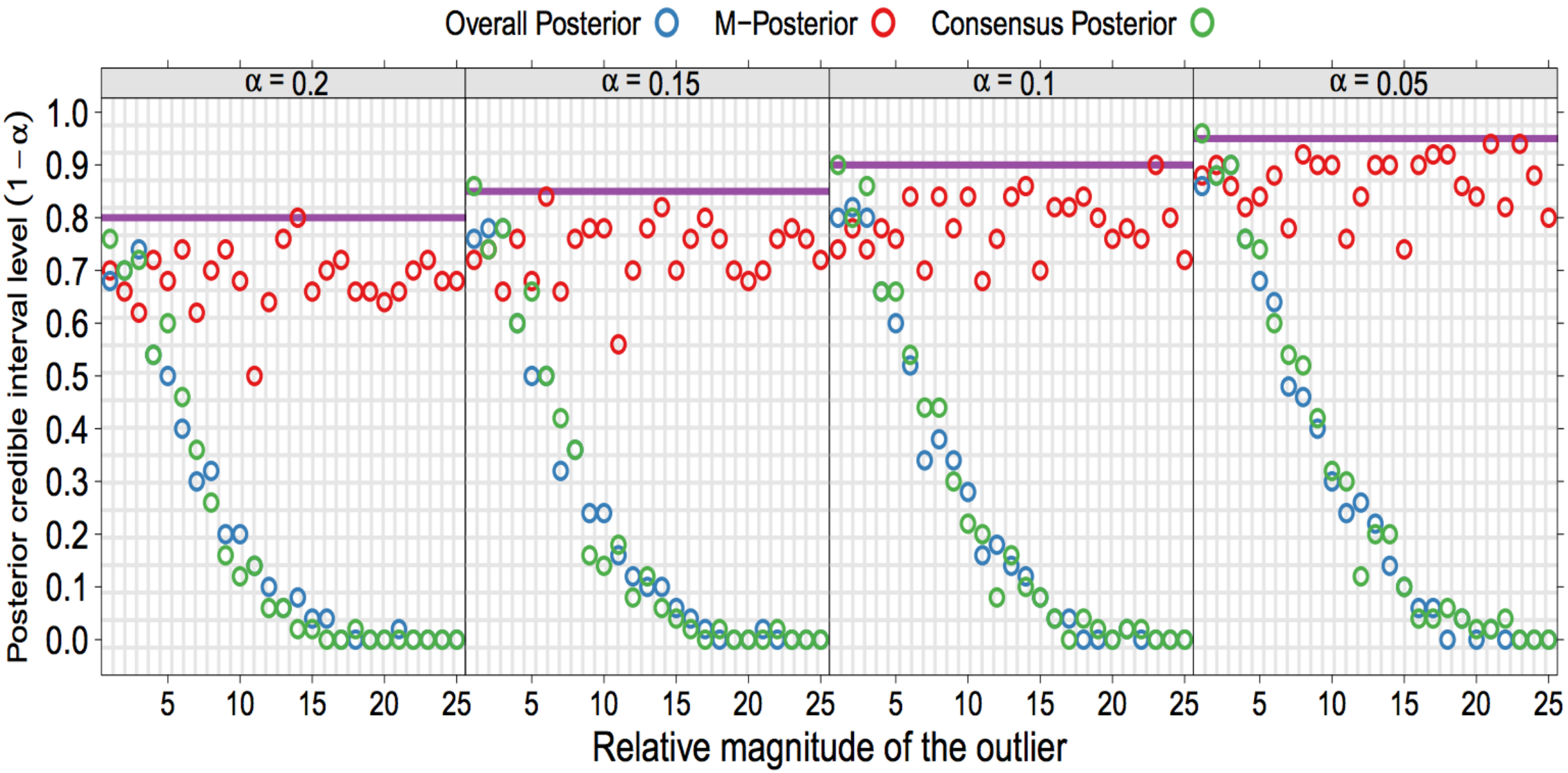
$$\text{posterior}_{\text{SA}} \propto \prod_{\text{subset}} \text{likelihood}_{\text{subset}}^{R} \times \text{prior}$$

approximation of the overall likelihood
(right order of variance)

Example: Simulated Gaussian Data

- 25 sets of 100 corrupted univariate Gaussian data
  - First 99 samples are simulated from standard Gaussian distribution
  - 100th sample is outlier whose value linearly increases from i=1,…, 25 such that $x_{i100} = i \max(x_{i1}, \cdots, x_{i99})$
- Estimate media posterior by randomly dividing data into 10 subsets
- Assume the variance is known to be 1, subset posteriors obtained via stochastic approximation
- 50 such replications are performed

Gaussian Simulation Results



M-posterior shows robustness to outliers!

Example: Simulated Gaussian Process Regression

- Simulate 100 (case 1) and 1000 (case 2) observations for x between 0 and 1 and Gaussian noise via function

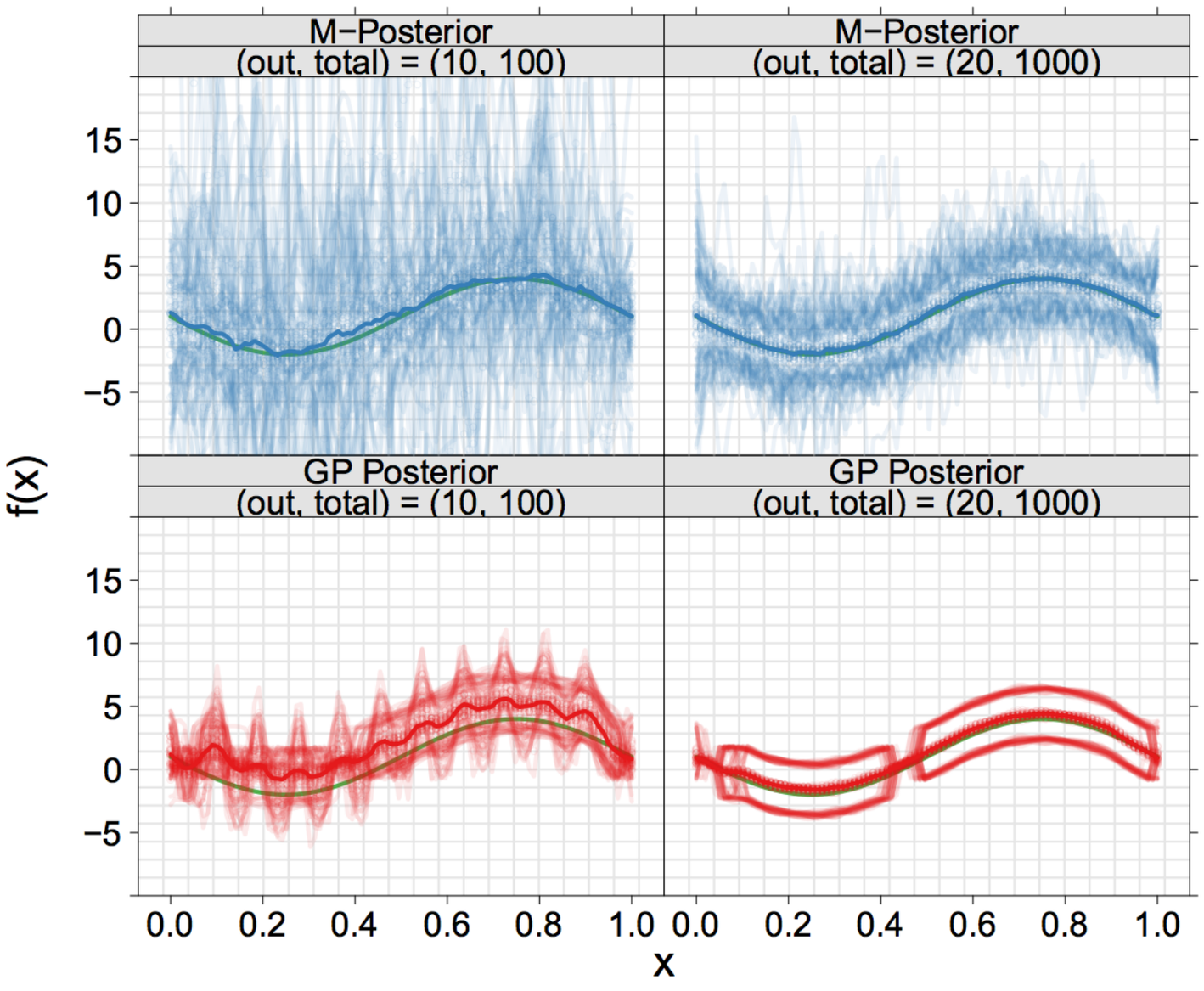
$$f_0(x) = 1 + 3\sin(2\pi x - \pi)$$

- Case 1 has 10 outliers, case 2 has 20 outliers (number of subsets equal to number of outliers)

- For observations 10^5 and above, GP fit fails due to numerical instability

- M-Posterior works with subsets so can always chose subsets to avoid numerical instabilities due to matrix inversion

GP Regression Results

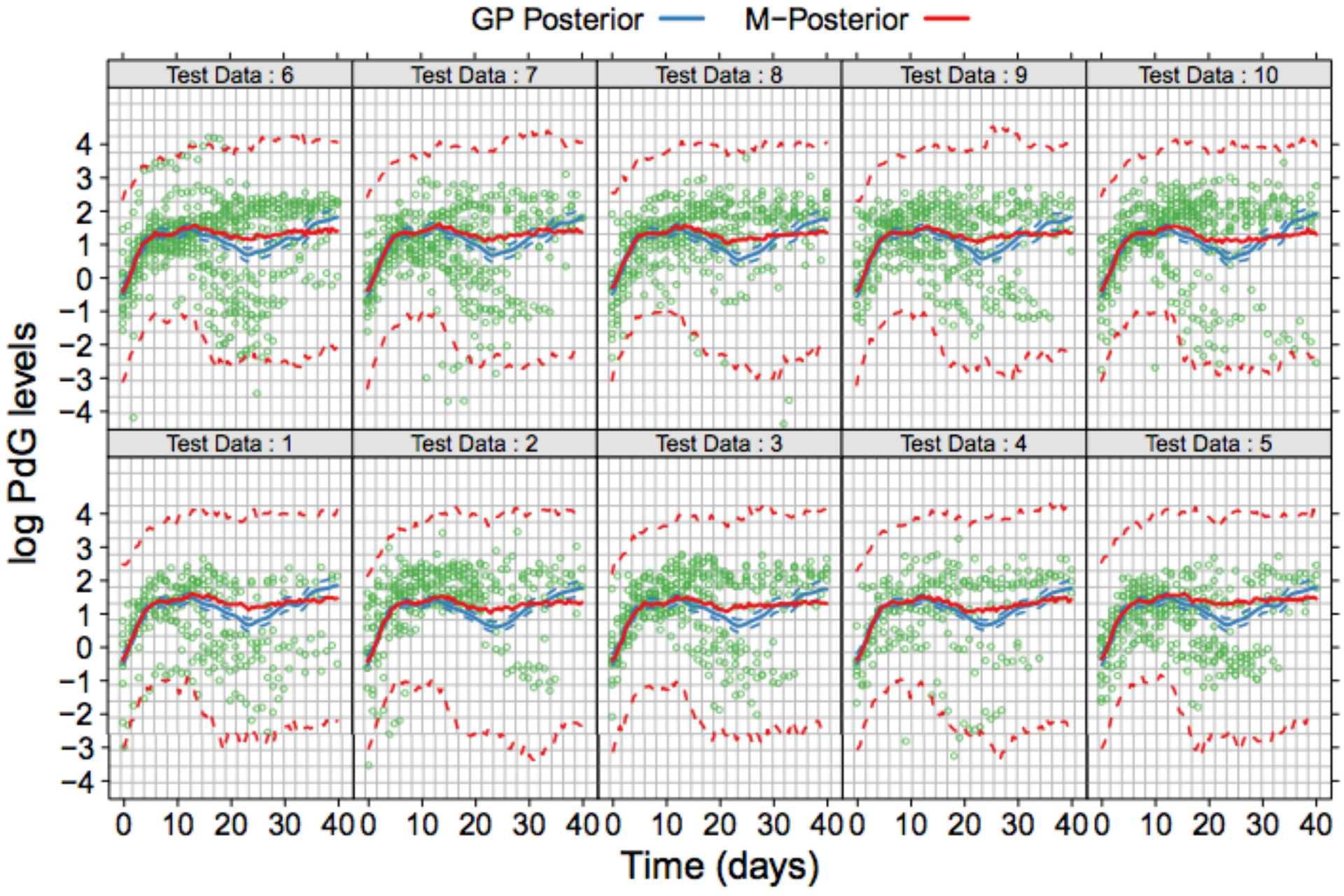


- Case 1: outliers is large compared to observations, so posterior inference ins unstable
- GP posterior is heavily influenced by outliers
- Both M-posterior and GP posterior yield similar results for case 2

Experiment: Hormone Data

- PdG hormone levels measured in 166 women from the day of ovulation across 41 time points
  - Information about different stages of conception and non-conception
  - Missing data and extreme observations are common
    - Late ovulation cycle data is sparse
- Discard data from women missing at least half the time points
- Fit GP regression of log PdG levels on time of ovulation for 124 women
- Both GP regression and M-Posterior to estimate f for 10 fold CV

Hormone Data: Results



Hormone Data: Discussion

- GP Posterior severely underestimates uncertainty
- M-Posterior CI levels include most of the data in the earlier part of the ovulation cycle
  - This region has most data so it leads to most reliable inference
  - Late ovulation cycle has very few points, so CI is wider
- M-Posterior accounts for outliers and model misspecification —> reliable uncertainty quantification across all folds

Summary

- Approach for scalable Bayesian inference using M-Posterior based on RKHS embedding of probability measures for estimating median posteriors
    - Distributed learning and scales naturally to massive data
    - Median provides robustness, stochastic approximation efficiency, and Weiszfield algorithm for easy implementation
- Extensions:
    - Extend Weiszfield using ADMM for distributed setting
    - Generalize to different choices of distances