STAT 30850 Final Report

Arjun Biddanda (abiddanda@uchicago.edu)
Joseph Marcus (jhmarcus@uchicago.edu)

March 16, 2016

Introduction

There are many contexts and applications where data is observed sequentially through time. For instance in high freqeuncy stock trading, investment firms have to make rapid descions in repsonse to new stock evaluations on micro-second timescales, or in A/B testing, techology companies often test the effect of varied advertisements on the "click behavior" of a user which is correlated with the effectiveness of the advertisement [CITE, CITE]. The setting in which hypotehsis testing must be performed on sequential streaming data is called "online testing". In online testing controlling the False Discovery Rate (FDR) at a given level has unique challenges as one doesn't observe all the data that could potentially be seen, later in the time series. Here we propose to use and implement a Bayesian model-based appraoch to control FDR in the online testing setting. We review and contrast our approach to previous commonly used heuristics / algorithims that are effectetive but conservative in online hyptohesis testing. We show our approach has higher power when compared to previous methods. Finally, we discuss future extensions and applications of our method.

Background

Broadly speaking, previous methods for controlling FDR in the online testing context use heuristics that increase or decrease the level at which one rejects a test depending on the number of previous discoveries made. Here we review three related, commonly used and well studied approaches to FDR control: α - investing, Levels Based on Number of Discoveries (LBOND), Levels Based on Recent Discoveries (LBORD) [CITE, CITE, CITE].

α-investing

Let:

t - be time
$w(t)$ - be a wealth function which changes through time
P_t - be a p-value output from an arbitrary test a time t
α - a global level that one would like to control FDR at
α_t - a time specific level

In alpha-investing ones defines a wealth function w. We imagine p-values are streaming in over time t which are provided by some abitrary test. We then proceed to run the α-investing procedure:

1. $w(t=0) = \alpha$

2. At time t choose $\alpha_t \leq \frac{w(t-1)}{1+w(t-1)}$

3. Rejct the null hypothesis if $P_t \leq \alpha_t$
4. Define $w(t)$ as a function of $w(t-1)$

$$w(t) = \begin{cases} w(t-1) + \alpha & P_t \leq \alpha_t \\ w(t-1) - \frac{\alpha_t}{1-\alpha_t} & P_t > \alpha_t \end{cases}$$

5. Repeat the procedure starting back at (2) for each new data point in the time series.

As we can see above when we reject the null, the wealth function grows and when we fail to reject the null the wealth function decays. Specifically at time 0 we set the wealth function to a "global level" alpha. We then proceed to set a time specific α_t. We then reject or fail to reject the p-value P_t from time t and redefine our wealth function w depeding on what desicion was made. This ensures that the more discoveries we make the less stringent we are through time and recipricaly the fewer discoveries we make the more stingent we are through time. For instance if we fail to reject for many sequential time points the signals have to be very strong to overcome the current state of the wealth function.

LBOND / LBORD

Let:

t - be time
P_t - be a p-value output from an arbitrary test a time t
α - a global level that one would like to control FDR at
β_t - a time specific weight
D_t - count of discoveries made up to time t

In Levels Based on Number of Discoveries (LBOND) we define a series of weights β_t which all sum up to the global level α. We then set a time specific α_t equal to the the weight at time t times the max of 1 and the number of discoveries made up to the last time step D_{t-1}. We reject a p-value P_t if its less that α_t and add to our discovery count.

1. At time t set $\alpha_t = \beta_t \cdot max\{1, D_{t-1}\}$ where $\sum_{t=1}^{\infty} \beta_t = \alpha$
2. Reject if $P_t \leq \alpha_t$
3. If discovery add to D
4. Repeat

Levels Based on Recent Discoveries follows a similar appraoch but uses weights from the time when the last discovery was made.

1. At time t set $\alpha_t = \beta_t \cdot max\{1, D_{t-\tau t}\}$ where $\sum_{t=1}^{\infty} \beta_t = \alpha$

2. Reject if $P_t \leq \alpha_t$
3. If discovery add to D
4. Repeat

Where $\tau(t)$ is the time of the most recent discovery before time t and $\tau(t)$ starts by being set to zero. LBORD has consistent pwoer over time becuase the β threshold is reset after each discovery.

Methods

Bayesian FDR

Here we propose to apply a Bayesian approach to FDR control to the online testing setting. Specifically we follow the work of Efron and model our streaming data as test statistics coming from a mixture model [CITE]. A Bayesian approach to FDR control considers an underlying mixture distribution consisting of *null* and *signal* components, and controls FDR based on the parameters of this distribution:

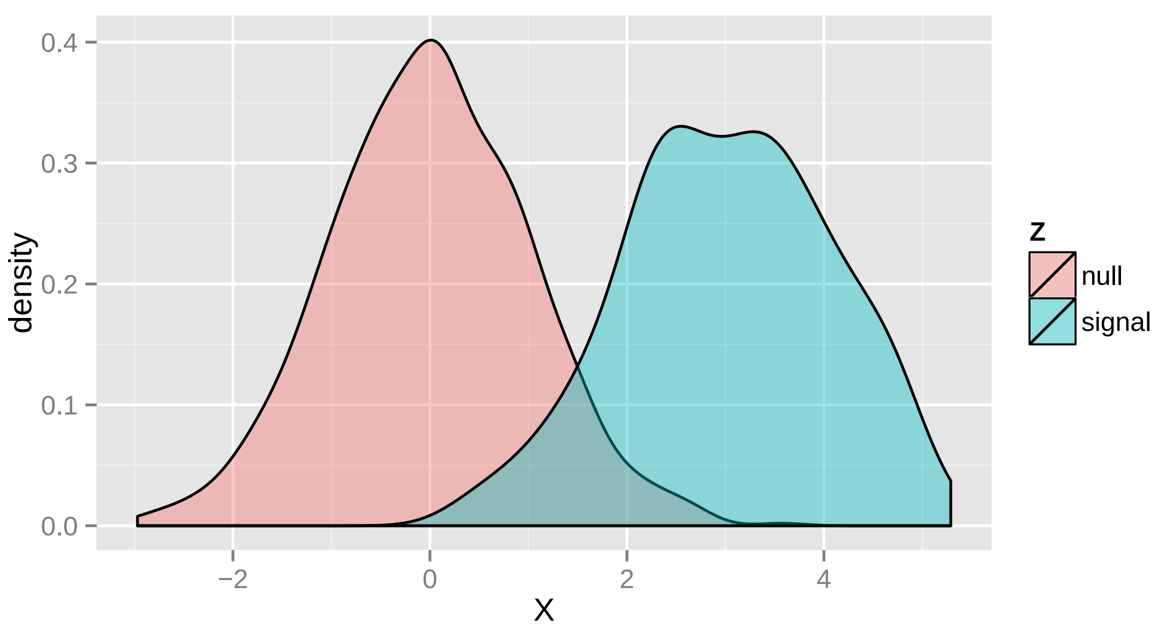

Figure 1*: Density of a mixture distribution of gaussians simulated with 80% proportion of nulls, the mean of the signal component at 3, variance of the signal component at 1, the mean of the null component at 0 and the variance of the null component at 1*

Let:

X - be a test statistic
π_0 - be the proportion of nulls

μ_1 - be the mean of the signals
σ_1^2 - be the variance of the signals

X can be modeled as a mixture of Gaussians:

$$X \mid \pi_0, \mu_1, \sigma_1^2 \sim \pi_0 N(0,1) + (1-\pi_0)N(\mu_1, \sigma_1^2)$$

In *Figure 1* we can see a plot the resulting density of a simulated mixture model with the underlying parameters $\theta = \{\mu_0 =, \sigma_0^2 = 1, \pi_0 = .8, \mu_1 = 3, \sigma_1^2 = 1\}$. We can see that assuming the data comes from an underlying mixture model with diverged means between the signal and null components can provide valuable information and flexibile approaches to controlling FDR. Paticularly if we assume that we only know the parameters of the null component. We can find the Bayesian interpretation of FDR by (following homework 2):

$$FD\hat{R}(x) = E[FD\hat{P}(x)]$$

the expectation of the denominator is

$$E[\#X \geq z] = nP(X \geq x)$$

becuase this should be binomially distrubed with size n and $p = P(X \geq x)$ thus one can rewrite the $FD\hat{R}(x)$ as

$$\begin{aligned} FD\hat{R}(x) &= E[\frac{n(1-\Phi(x))}{n(\pi_0(1-\Phi(x)) + (1-\pi_0)(1-\Phi(\frac{x-\mu_1}{\sigma_1}))}] \\ &= E[\frac{(1-\Phi(x))}{(\pi_0(1-\Phi(x)) + (1-\pi_0)(1-\Phi(\frac{x-\mu_1}{\sigma_1}))}] \end{aligned}$$

This is a constant thus

$$= \frac{(1-\Phi(x))}{(\pi_0(1-\Phi(x)) + (1-\pi_0)(1-\Phi(\frac{x-\mu_1}{\sigma_1}))}$$

We can see that if we estimate π_0, μ_1, σ_1^2 then we can control FDR at a given level:

$$\alpha = \frac{\pi_0(1-\Phi(\hat{x}))}{\pi_0(1-\Phi(\hat{x})) + (1-\pi_0)\left(1-\Phi\left(\frac{\hat{x}-\mu_1}{\sigma_1}\right)\right)}$$

Where we reject X if $X > \hat{x}$.

Markov Chain Monte Carlo (Gibbs Sampler)

We apply this mixture model framework to online testing by estimating the unknown parameters of the gaussian mixture model described above at each time point t. Specifically we use a Markov Chain Monte Carlo approach to sample from the posterior distributions of the unknown parameters $\theta = \{\pi_0, \mu_1, \sigma_1^2\}$.

Let:

t - time index of a test statistic streaming in
X - a vector of t test statistics that have streamed in
X_t - the test statistic at the t^{th} time point
Z - vector of latent states of X_t being a signal or null
Z_t - latent state at time t of X_t being a signal or null
π_0 - proportion of nulls
μ_1 - mean of the signals
σ_1^2 - variance of the signals

As described above we model X_t as a mixture of Gaussians:

$$X_t \mid \pi_0, \mu_1, \sigma_1^2 \sim \pi_0 N(0,1) + (1-\pi_0) N(\mu_1, \sigma_1^2)$$
$$X_t \mid Z_t = 0 \sim N(0,1)$$
$$X_t \mid Z_t = 1, \mu_1, \sigma_1^2 \sim N(\mu_1, \sigma_1^2)$$

We can reparameterize this model in terms of the precision ϕ_1 of the signals and write down the likelihood of the model conditioned on the latent indicators as:

$$L(\pi_0, \mu_1, \sigma_1^2 \mid X, Z) \propto (\pi_0)^{n_0} exp(-\frac{1}{2} \sum_{t:z_t=0} x_t^2) \cdot (1-\pi_0)^{n_1} exp(-\frac{\phi_1}{2} \sum_{t:z_t=1} (x_t - \mu_1)^2)$$

where n_0 and n_1 are the number of observed nulls and signals respectively. We can then set priors on π_0, μ_1, ϕ_1 which satisfy conjugacy:

$$\pi_0 \sim Beta(\alpha, \beta)$$
$$\phi_1 \sim Gamma(\frac{a}{2}, \frac{b}{2})$$
$$\mu_1 \mid \phi_1 \sim Normal(\mu^*, \frac{1}{\alpha^* \phi_1})$$

thus the posterior distributions of these parameters can be written as:

$$\pi_0 \mid X, Z = 0 \sim Beta(\alpha + n_0, \beta + n_1)$$
$$\phi_1 \mid X, Z \sim Gamma(\frac{a + n_1}{2}, b + \sum_{t:z_t=1} (x_t - \mu_1)^2)$$

$$\mu_1 \mid X, Z, \phi_1 \sim Normal(\frac{\alpha^*\mu^* + n_1 + \bar{x_1}}{\alpha^* + n_1}, \frac{1}{(\alpha^* + n_1)\phi_1})$$

We also need to sample from the posterior of Z due to the conditional dependencies above:

$$P(Z_t \mid X_t = x_t, \pi_0, \mu_1, \phi_1) = \frac{\pi_0 exp(-\frac{x_t^2}{2})}{\pi_0 exp(-\frac{x_t^2}{2}) + ((1-\pi_0)\phi_1 exp(-\frac{\phi_1}{2}(x_t - \mu_1)^2))}$$

Results

Simulation Study

We simulated independent and identically distributed data from a mixture model with the following parameters $\theta = \{\pi_0 = 0.80, \mu_1 = 3, \sigma_1^2 = 1\}$ for 1000 timesteps. The resulting simulated values can be seen in *Figure 2*.

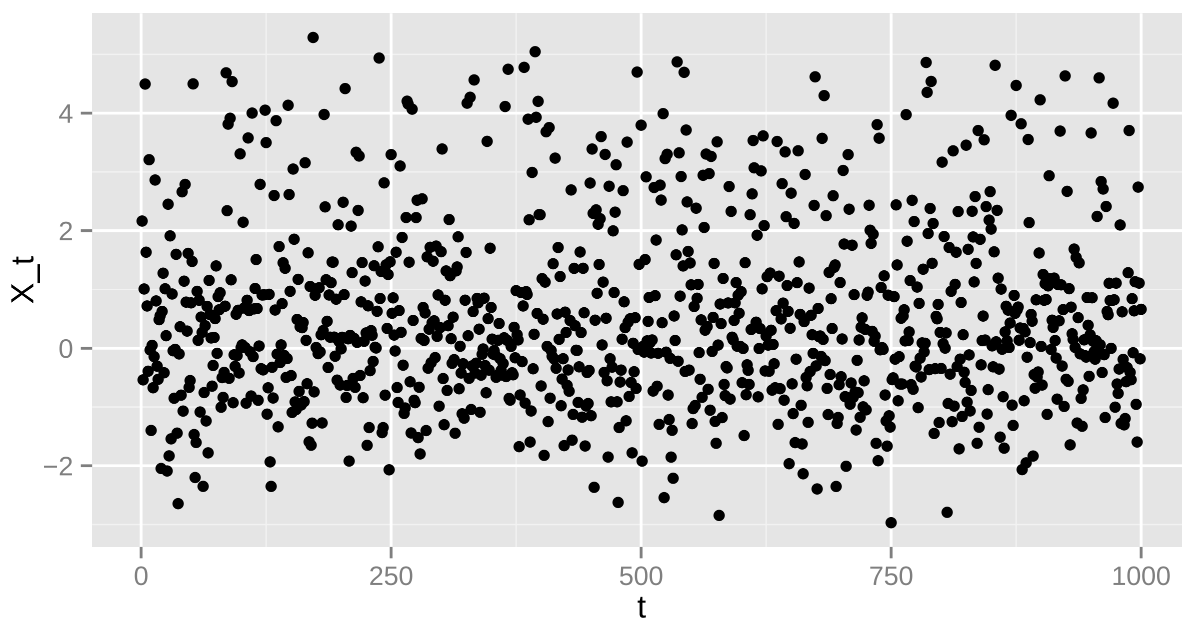

***Figure 2**: 1000 Simulated Z-scores from mixture distribution with $\theta = \{\pi_0 = 0.80, \mu_1 = 3, \sigma_1^2 = 1\}$

From these simulated values we were able to then use our Gibbs sampler to perform parameter estimations at each time point. The algorithm proceeds as follows:

1. At timestep t, treat $X = (X_1, ..., X_t)$
2. Run 1000 iterations of the Gibbs Sampler 2a. After a burn-in period of 50 iterations, we then choose one sample every 10 iterations.
3. Calculate 95% credible interval from the samples
4. repeat for the $t+1$ timestep

In *Figure 3* we can see the 95% credible intervals of the parameters as a function of the time. Note that we *a priori* would expect that the parameter estimation in the beginning is rather poor due to a small amount of data used for the Gibbs sampling to estimate the posterior distribution.

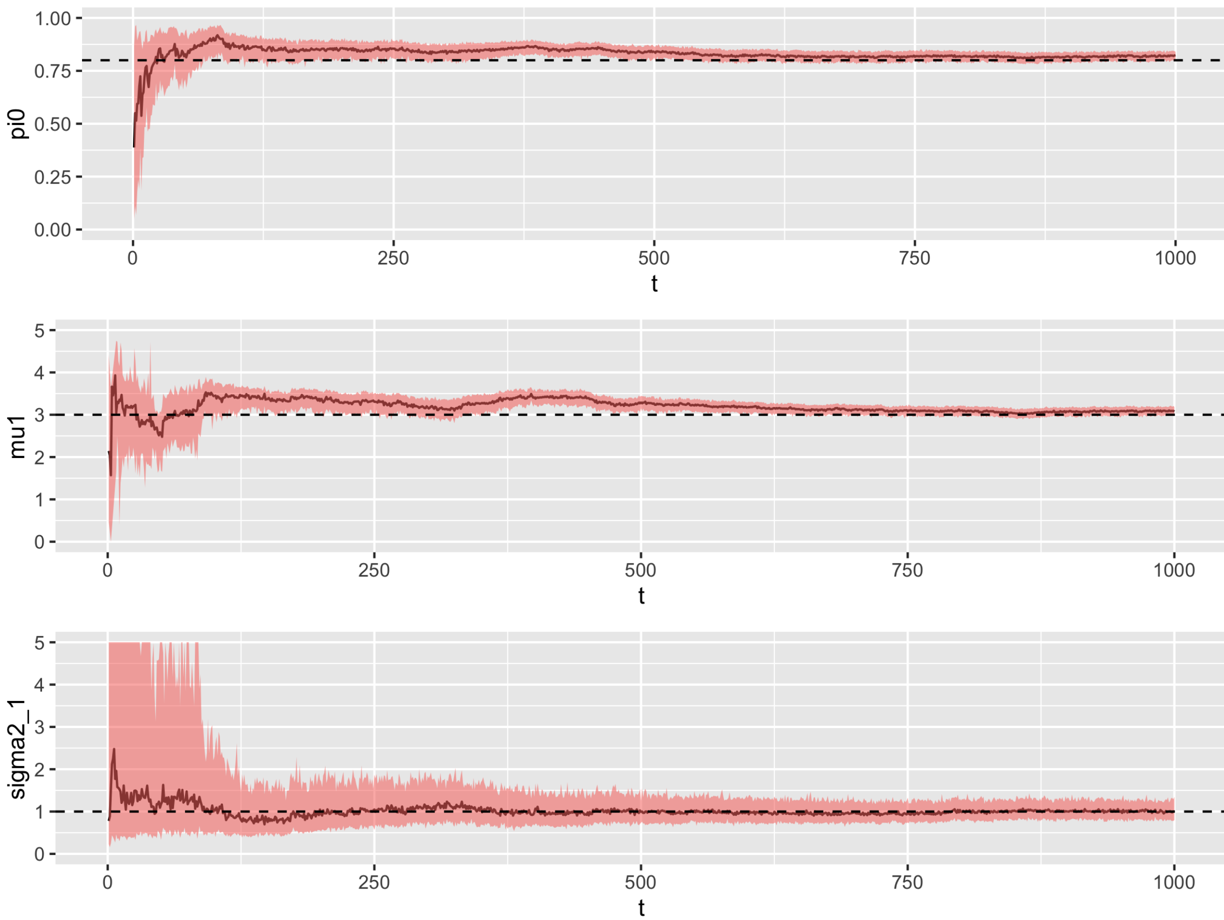

***Figure 3**: Sequentially estimated credible intervals of mixture model parameters under simulation conditions. Note that for σ_1^2 we truncated the 95^{th} quantile to 5 due to issues with numerical precision. We believe that this cutoff value does not sufficiently affect our analyses.

Estimating $\hat{x}$ Conservatively

Intuitively in the beginning of the time-series we have very little data to estimate our model, and thus our BayesFDR threshold may not be valid if we take a summary of our parameter estimates such as the mean. Thus we performed some numerical experiments in order to determine the appropriate bounds of $\theta = \{\pi_0, \mu_1, \sigma_1^2\}$. We simply varied only one of the parameters while keeping the others fixed to the parameters in our simulation. The results of these simulations are shown in *Figure 4*

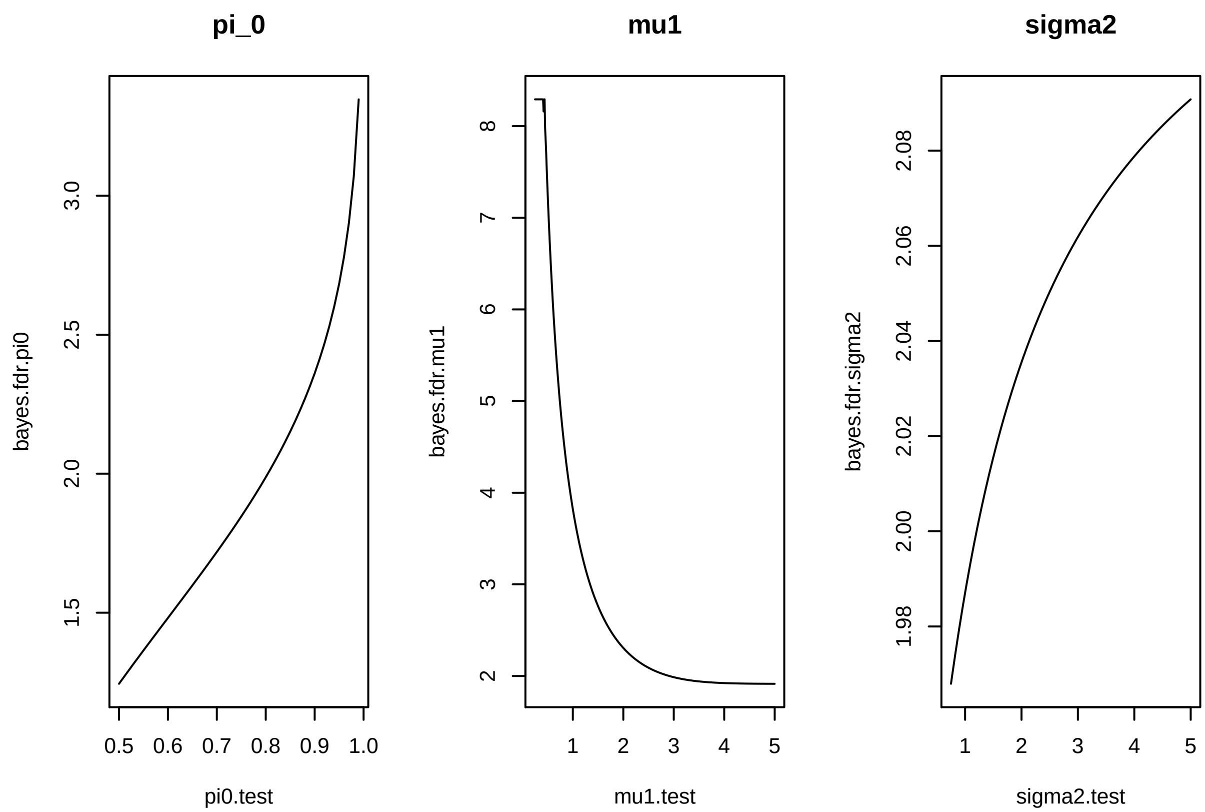

***Figure 4** : The

From the experiments above we have established that we are the most conservative when the following conditions hold:

1. π_0 is high (closer to 1)
2. μ_1 is low (closer to the null value)
3. σ_1^2 is high (the signal distribution is made more variable)

From these conclusions on the behavior of BayesFDR, we are able to get reasonably conservative bounds on the parameters that we desire. We take the 5% credible interval bound for μ_1, and the 95% credible interval bound (upper bound) for pi_0 and $sigma_1^2$. The resulting values of $\hat{x}$ using these bounds are shown in *Figure 5*. It is promising to see that these estimates are higher than the value of $\hat{x}$ if we knew the parameters of the mixture model *a priori*.

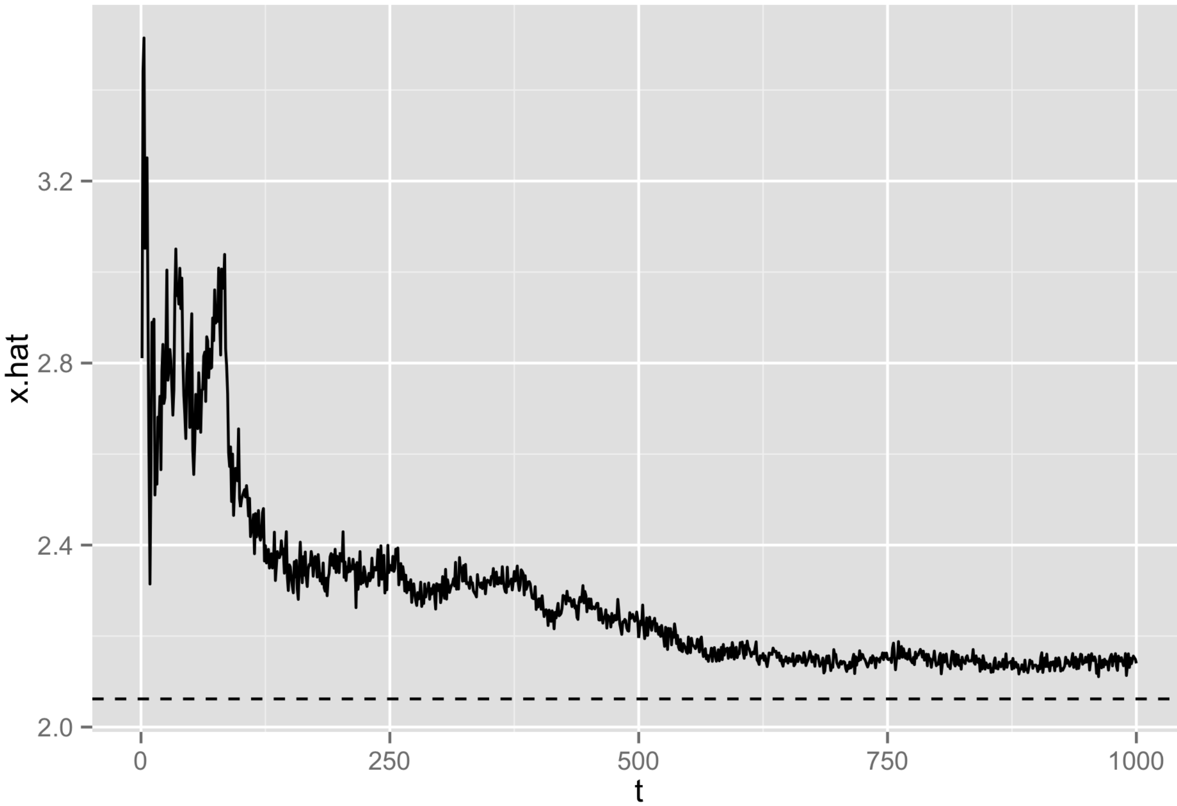

***Figure 5**:* Plot of $\hat{x}$ with respect to time. We take the 5% credible interval bound for μ_1, and the 95% credible interval bound (upper bound) for pi_0 and $sigma_1^2$. The dashed line represents the $\hat{x}$ value using the true parameters for the simulation.

Empircal Estimates of FDP and Power

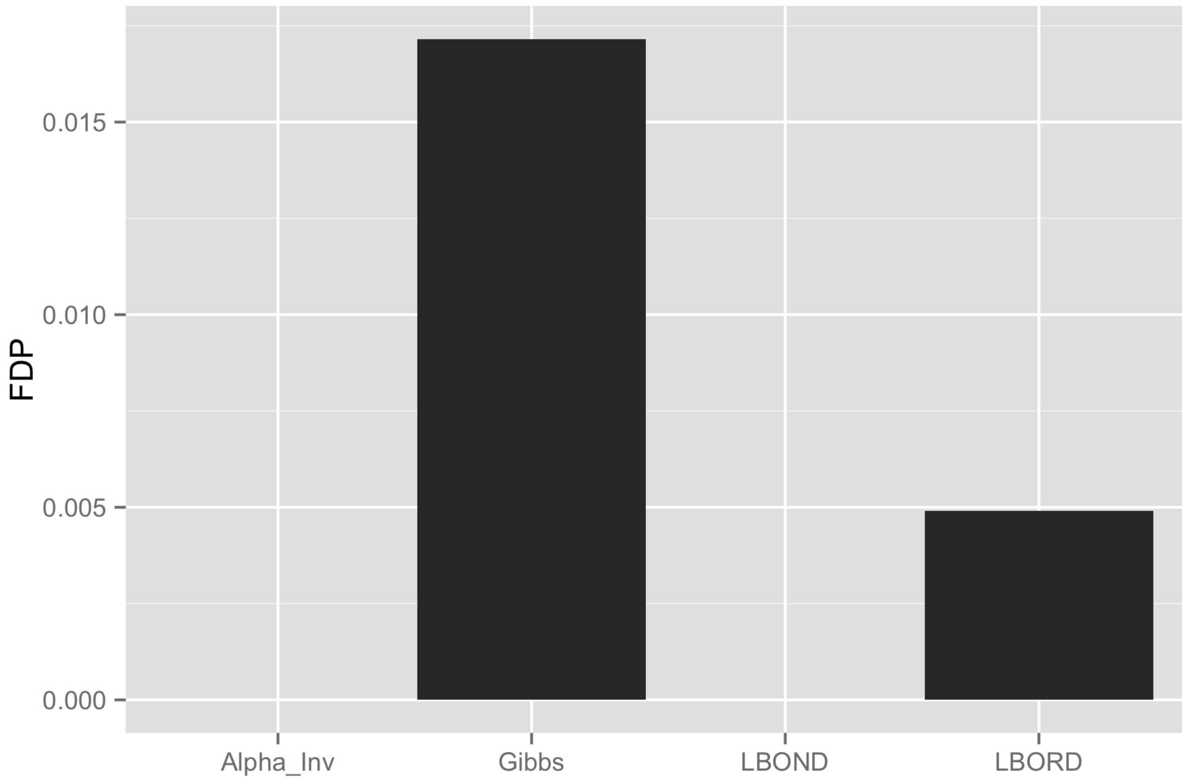

Figure 6:

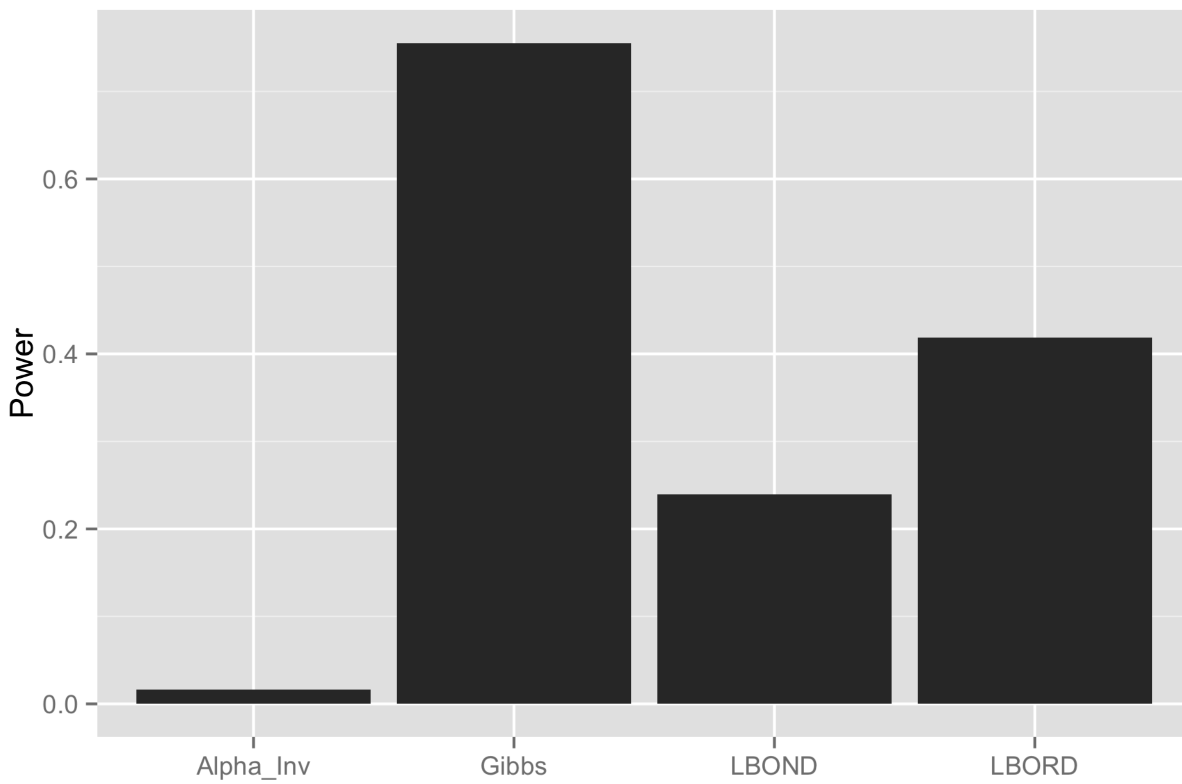

Figure 7:

Conclusion

References