Adventures in sparsity and shrinkage with the normal means model

Matthew Stephens

November, 2019

The Normal Means problem

$$x_j | \theta_j, s_j \sim N(\theta_j, s_j^2)$$

MLE: $\hat{\theta}_j = x_j$.

Surprise: you can do better than the mle! (Stein, 1956)

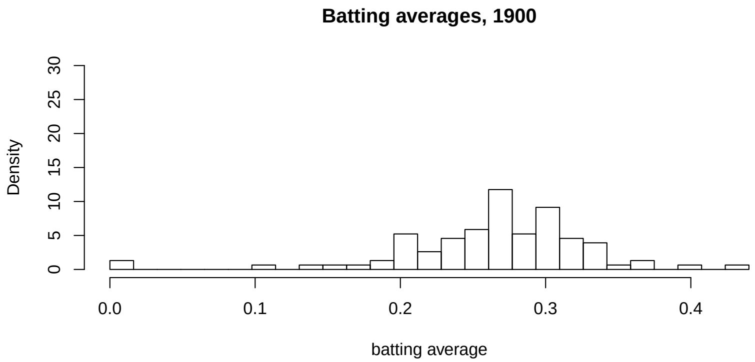
The Normal Means problem

$$x_j|\theta_j, s_j \sim N(\theta_j, s_j^2)$$

$$\text{MLE: } \hat{\theta}_j = x_j.$$

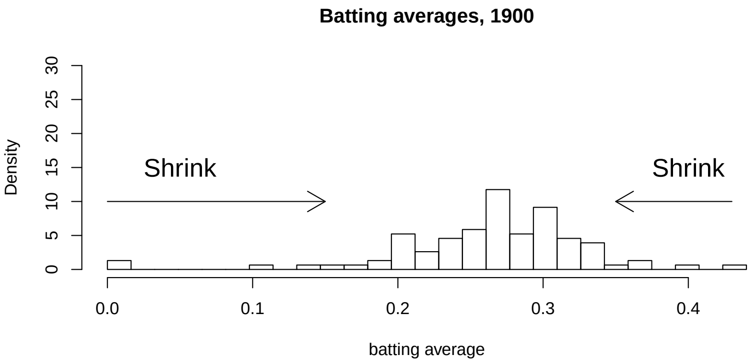
Surprise: you can do better than the mle! (Stein, 1956)

Shrinkage Estimation[1]



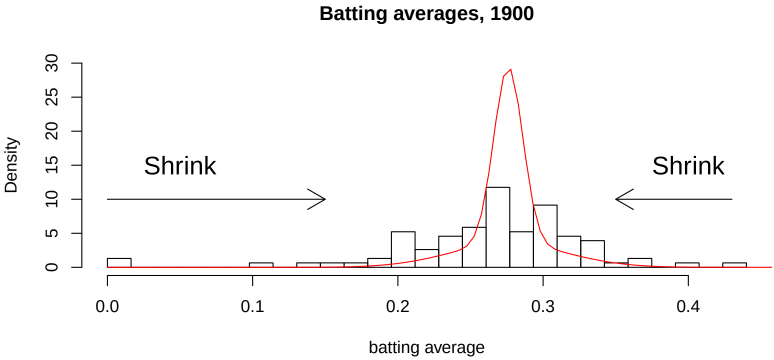
[1] http://varianceexplained.org/r/empirical_bayes_baseball/

Shrinkage Estimation[1]



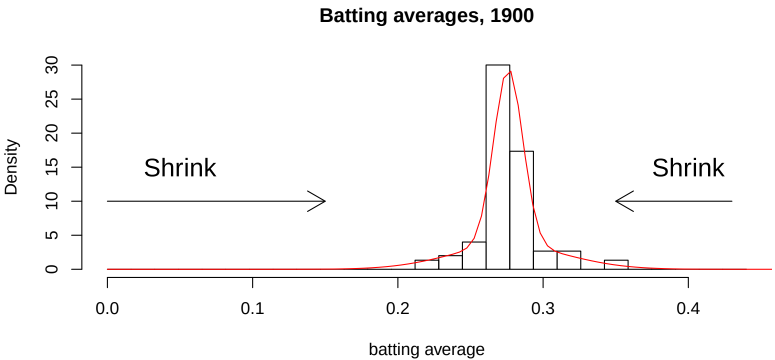
[1] http://varianceexplained.org/r/empirical_bayes_baseball/

Shrinkage Estimation[1]



[1] http://varianceexplained.org/r/empirical_bayes_baseball/

Shrinkage Estimation[1]



[1] http://varianceexplained.org/r/empirical_bayes_baseball/

Empirical Bayes Normal Means (EBNM)

$$x_j|\theta_j, s_j \sim N(\theta_j, s_j^2)$$

$$\theta_j \sim g \in \mathcal{G}$$

Fit this model in two steps:

1. Estimate g by maximizing (marginal) log-likelihood:

$$\widehat{g} = \arg\max \sum_j \log \int p(x_j|\theta_j, s_j) g(d\theta_j)$$

2. Compute posterior distributions $\theta_j \mid \widehat{g}, x_j, s_j$.

“Sparsity-inducing” choices for $\mathcal{G}$

- Point-normal: $\pi_0\delta_0 + (1-\pi_0)N(0,\sigma^2)$.
- Zero-centered scale mixtures of normals (non-parametric; includes point-normal, t, Laplace, horseshoe, ...).

Surprise: computations for latter are easier than former! (“convex relaxation”)

“Sparsity-inducing” choices for $\mathcal{G}$

- Point-normal: $\pi_0\delta_0 + (1-\pi_0)N(0,\sigma^2)$.
- Zero-centered scale mixtures of normals (non-parametric; includes point-normal, t, Laplace, horseshoe, ...).

Surprise: computations for latter are easier than former! (“convex relaxation”)

Simple non-parametric computations

Key idea: approximate non-parametric family by finite mixture with many components:

$$g(\cdot) = \sum_k^K \pi_k N(\cdot; 0, \sigma_k^2)$$

with K big; $\sigma_1, \ldots, \sigma_K$ fixed on a "dense grid".

So estimating g comes down to estimating π.

Illustration: scale mixture of normals

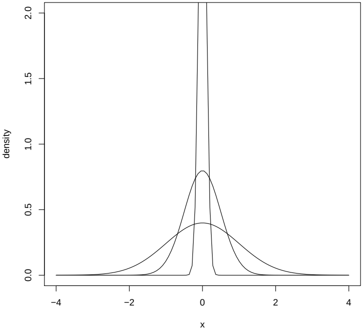
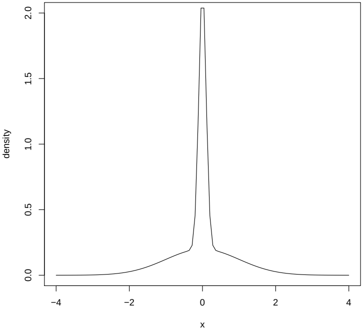


Illustration: scale mixture of normals



Simple non-parametric computations

This yields simple marginal distribution:

$$p(x_j|\pi) = \sum_k^K \pi_k N(x_j; 0, s_j^2 + \sigma_k^2).$$

And estimating $\pi = (\pi_1, \ldots, \pi_K)$, is a convex optimization problem (Koenker + Mizera, 2015; S. 2017; Kim et al, 2018).

Illustration:Bayesian shrinkage

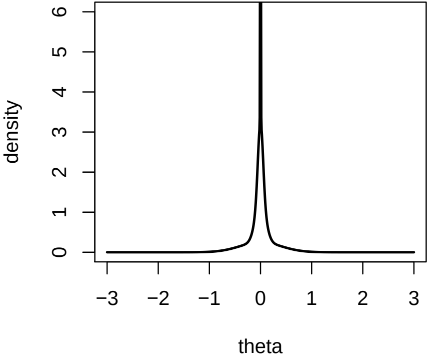


Illustration: Bayesian shrinkage

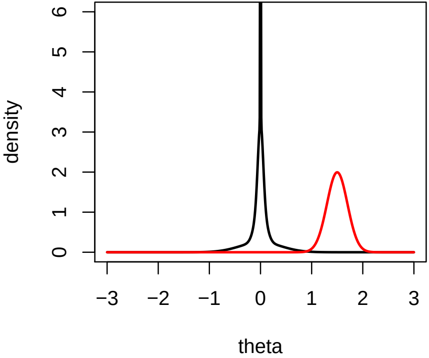


Illustration: Bayesian shrinkage

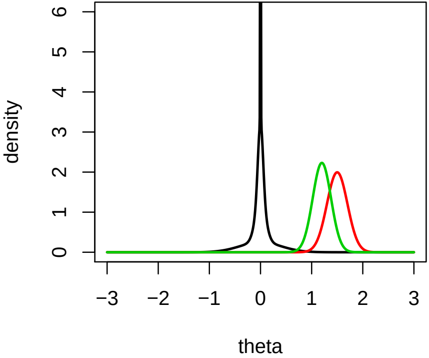


Illustration: Bayesian shrinkage

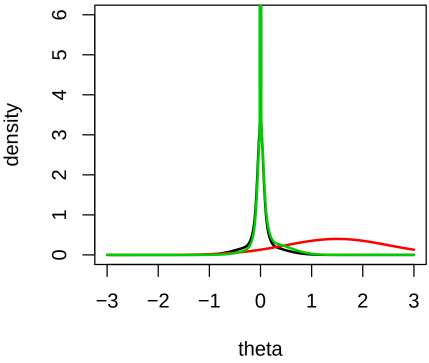
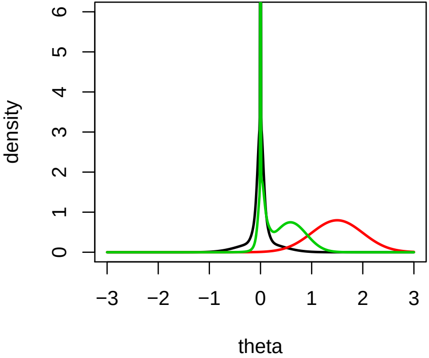


Illustration: Bayesian shrinkage



Bayesian shrinkage operators

Shrinkage obviously depends on prior g (and standard error s_j).

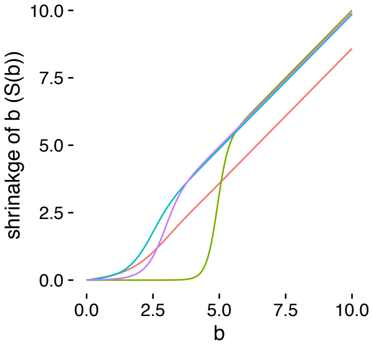
One way to summarize shrinkage behavior is to focus on how posterior mean changes with x:

$$S_{g,s}(x) := E(\theta_j | x_j = x, g, s_j = s)$$

Call this the "shrinkage operator" for prior g.

Bayesian shrinkage operators

Example shrinkage operators for different priors (scale mixtures of normals, $s = 1$):



Shrinkage operators via penalized likelihood

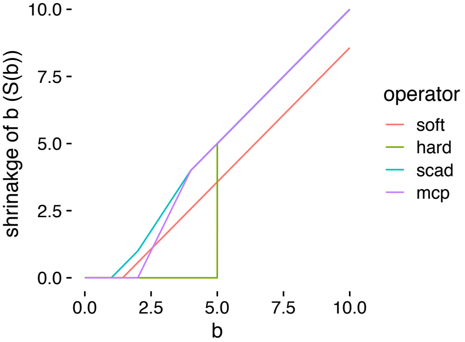
Another way to induce shrinkage/sparsity is penalized log-likelihood:

$$\hat{\theta}_j = S_{h,\lambda}(x) := \arg\min_{\theta} \left[0.5(x-\theta)^2 + \lambda h(\theta)\right]$$

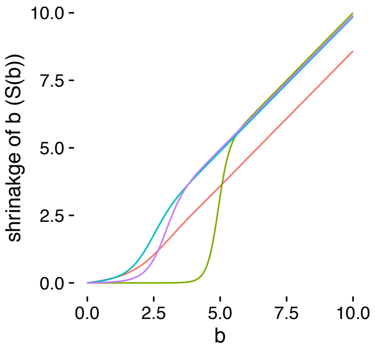
where h a penalty function, and λ a tuning parameter.

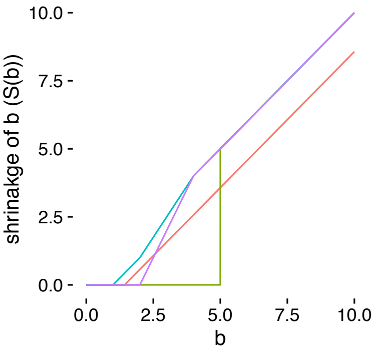
[Can think of these as posterior mode under some prior, but I don't recommend it!]

Penalty-based shrinkage operators



Bayesian vs Penalty-based shrinkage operators





Key features of EB shrinkage

1. Shrinkage determined by g, which is estimated by maximum likelihood, rather than CV.
2. Very flexible: can mimic a range of penalty functions.
3. Posterior distribution $\theta_j \mid \widehat{g}, x_j, s_j$ gives not only shrunken point estimates but also "shrunken" interval estimates.

...despite this, until recently little attention paid to EB shrinkage in practical applications.

Key features of EB shrinkage

1. Shrinkage determined by g, which is estimated by maximum likelihood, rather than CV.
2. Very flexible: can mimic a range of penalty functions.
3. Posterior distribution $\theta_j \mid \widehat{g}, x_j, s_j$ gives not only shrunken point estimates but also "shrunken" interval estimates.

...despite this, until recently little attention paid to EB shrinkage in practical applications.

Example Applications

- Multiple testing
- Linear Regression
- Matrix factorization

Multiple Testing

Typical set-up (e.g. Benjamini and Hochberg, 1995):

- Large number of tests $j = 1, \ldots, n$.
- Test j yields p value p_j.
- Reject all tests with $p_j < \gamma$ with $\gamma(p)$ chosen to control FDR.

Multiple Testing via EBNM

In many applications p values are derived from effect estimates, $\hat{\beta}_j$, and standard errors s_j, satisfying:

$$\hat{\beta}_j \sim N(\beta_j, s_j^2).$$

Aim: identify β_j that are different from zero.

Ideally suited to EBNM!

Multiple Testing via EBNM

$$\hat{\beta}_j | \beta_j \sim N(\beta_j, s_j^2)$$

$$\beta_j \sim g() \in \mathcal{G}$$

Estimate $\hat{g}$ by maximum likelihood; compute posterior 90% interval for each β_j; reject if interval does not contain 0.

Details: S. (2017); see also Thomas (1985), Efron (200x).

EBNM vs BH for multiple testing

- EBNM slightly more powerful.
- BH more robust to correlated tests (but see Sun + S. (2019)).
- EBNM provides shrinkage interval estimates! (e.g. address winner's curse)

But real benefit of EBNM maybe comes in multivariate extensions...

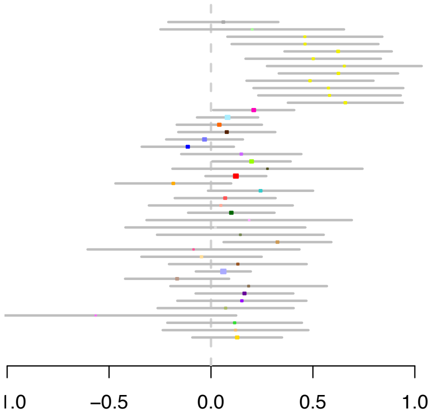
Multivariate multiple testing (Urbut et al, 2018)

$$\hat{\boldsymbol{\beta}}_j | \boldsymbol{\beta}_j \sim N_r(\boldsymbol{\beta}_j, V_j)$$

$$\boldsymbol{\beta}_j \sim g(\cdot) = \sum_k \pi_k N_r(0, \Sigma_k)$$

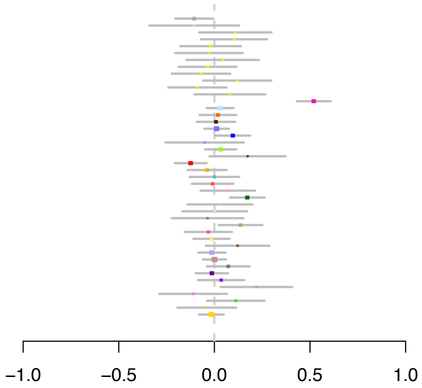
Multivariate multiple testing (Urbut et al, 2018)

Eg: eQTL effect sizes across 44 tissues (GTEx Consortium, 2017).



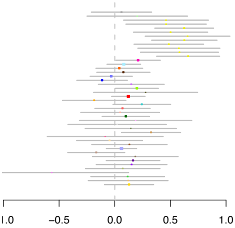
Multivariate multiple testing

Eg: eQTL effect sizes across 44 tissues (GTEx Consortium, 2017).

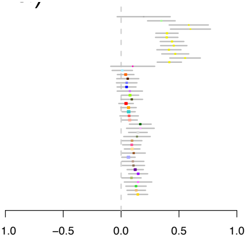


Multivariate multiple testing

a) Data

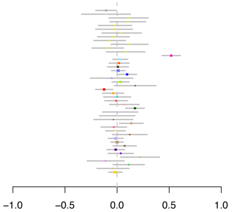


b) Posterior

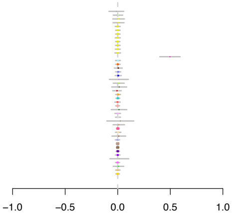


Multivariate multiple testing

a) Data



b) Posterior



Linear regression

$$\mathbf{y}_{n\times 1} = X_{n\times p}\mathbf{b}_{p\times 1} + \mathbf{e}_{n\times 1}$$

$$\mathbf{e} \sim N_n(0, \sigma^2 I_n)$$

$$b_1, \ldots, b_p \sim g() \in \mathcal{G}$$

Challenge: how to apply EBNM ideas here?

An analogy: Penalized regression

Penalized linear regression solves:

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b}} 0.5||\mathbf{y} - X\mathbf{b}||_2^2 + \lambda \sum_j h(b_j)$$

E.g. $h(b) = b^2$ gives ridge regression; $h(b) = |b|$ gives lasso.

Coordinate Ascent Iterative Shrinkage Algorithm (CAISA)

For each coordinate j, update b_j as follows:

- Compute residuals $\mathbf{r}_j := \mathbf{y} - X_{-j}\mathbf{b}_{-j}$
- Compute $\hat{b}_j = (\mathbf{x}_j^T \mathbf{x}_j)^{-1} \mathbf{x}_j^T \mathbf{r}_j$
- Shrink: $b_j := S_{h,\lambda}(\hat{b}_j)$

where S is a shrinkage operator for h, λ:

$$S_{h,\lambda}(b) := \arg\min_a (b-a)^2 + \lambda h(a).$$

g-CAISA:

For each coordinate j, update b_j as follows:

- Compute residuals $\mathbf{r}_j := \mathbf{y} - X_{-j}\mathbf{b}_{-j}$
- Compute $\hat{b}_j := (\mathbf{x}_j^T \mathbf{x}_j)^{-1} \mathbf{x}_j^T \mathbf{r}_j$
- Compute $s_j := (\mathbf{x}_j^T \mathbf{x}_j)^{-1} \sigma^2$
- Shrink: $b_j := S_{g,s_j}(\hat{b}_j)$

where S is the posterior mean shrinkage operator determined by prior g.

What is this doing?

Define:

$$F(q) := -KL(q \to p(b|X, \mathbf{y}, g, \sigma^2))$$

$$\mathcal{Q} := \{q : q(\mathbf{b}) = \prod_{j=1}^{p} q_j(b_j)\}$$

Proposition (Kim et al, in prep): The g-CAISA algorithm is a coordinate ascent algorithm for maximizing $F(q)$ (i.e. minimizing KL) over $q \in \mathcal{Q}$, with $\mathbf{b}$ the expectation of q.

Estimating g?

Recall algorithm:

- Compute residuals $\mathbf{r}_j := \mathbf{y} - X_{-j}\mathbf{b}_{-j}$
- **Compute** $\hat{b}_j := (\mathbf{x}_j^T \mathbf{x}_j)^{-1} \mathbf{x}_j^T \mathbf{r}_j$
- **Compute** $s_j := (\mathbf{x}_j^T \mathbf{x}_j)^{-1} \sigma^2$
- Shrink: $b_j := S_{g,s_j}(\hat{b}_j)$

Idea: after computing $\hat{b}_j, s_j$ for $j = 1, \ldots, p$, apply EBNM to estimate g.

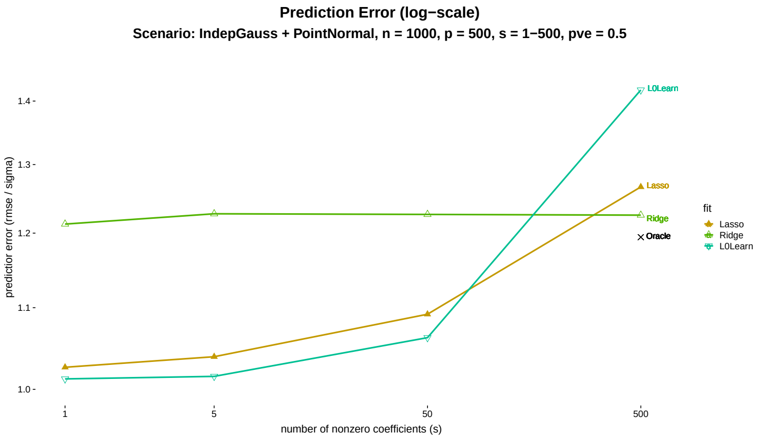
Estimating g?

Recall algorithm:

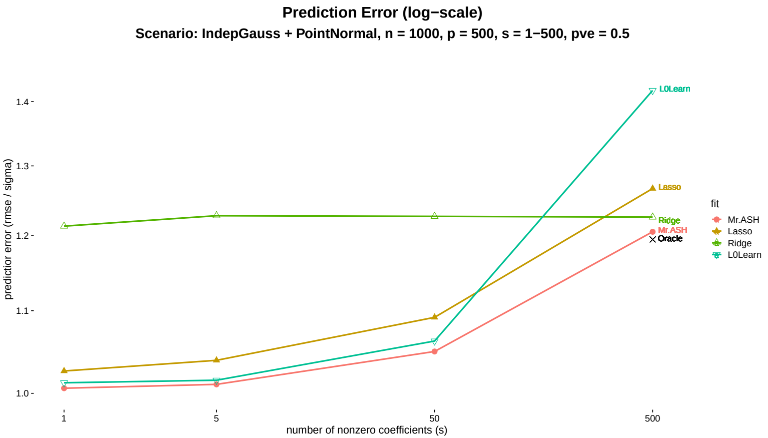
- Compute residuals $\mathbf{r}_j := \mathbf{y} - X_{-j}\mathbf{b}_{-j}$
- **Compute** $\hat{b}_j := (\mathbf{x}_j^T \mathbf{x}_j)^{-1} \mathbf{x}_j^T \mathbf{r}_j$
- **Compute** $s_j := (\mathbf{x}_j^T \mathbf{x}_j)^{-1} \sigma^2$
- Shrink: $b_j := S_{g,s_j}(\hat{b}_j)$

Idea: after computing $\hat{b}_j, s_j$ for $j = 1, \ldots, p$, apply EBNM to estimate g.

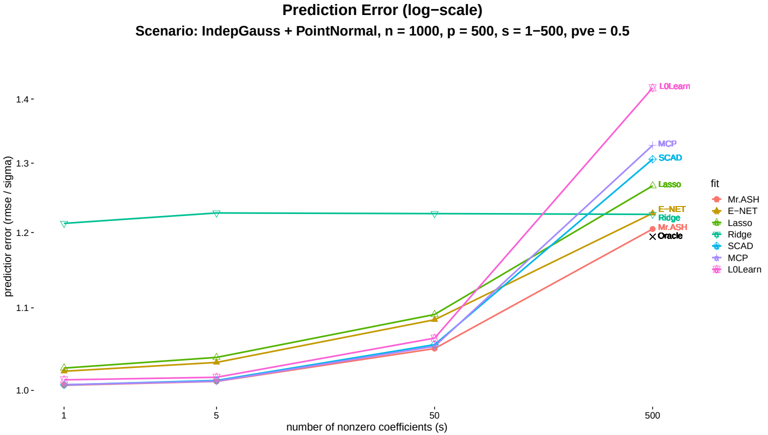
Simulation Results



Simulation Results



Simulation Results - more penalties



Matrix factorization

$$Y_{n \times p} = L_{n \times K} F^T_{K \times p} + E_{n \times p}$$

Common Assumption: F and/or L are sparse.

But how sparse?

Empirical Bayes Matrix Factorization: rank $K = 1$

$$Y = lf^T + E$$

$$l_1, \ldots, l_n \sim g^l(\cdot) \in \mathcal{G}$$
$$f_1, \ldots, f_p \sim g^f(\cdot) \in \mathcal{G}$$

Algorithm, in outline, iterates:

- Given f, estimate g^l, l by solving EBNM problem.
- Given l, estimate g^f, f by solving EBNM problem.

Optimizes a variational approximation to the posterior.

Empirical Bayes Matrix Factorization: rank $K = 1$

$$Y = lf^T + E$$

$$l_1, \dots, l_n \sim g^l(\cdot) \in \mathcal{G}$$
$$f_1, \dots, f_p \sim g^f(\cdot) \in \mathcal{G}$$

Algorithm, in outline, iterates:

- Given f, estimate g^l, l by solving EBNM problem.
- Given l, estimate g^f, f by solving EBNM problem.

Optimizes a variational approximation to the posterior.

Empirical Bayes Matrix Factorization: rank $K > 1$

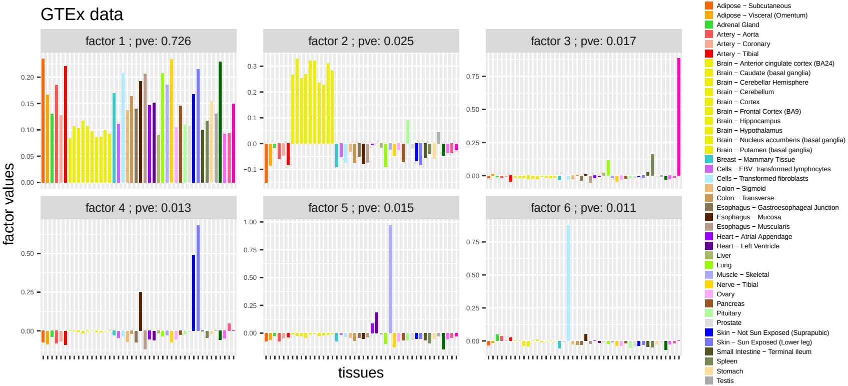
$$Y = \sum_{k=1}^{K} l_k f_k^T + E$$

$$l_{k1}, \ldots, l_{kn} \sim g_k^l(\cdot) \in \mathcal{G}$$
$$f_{k1}, \ldots, f_{kp} \sim g_k^f(\cdot) \in \mathcal{G}$$

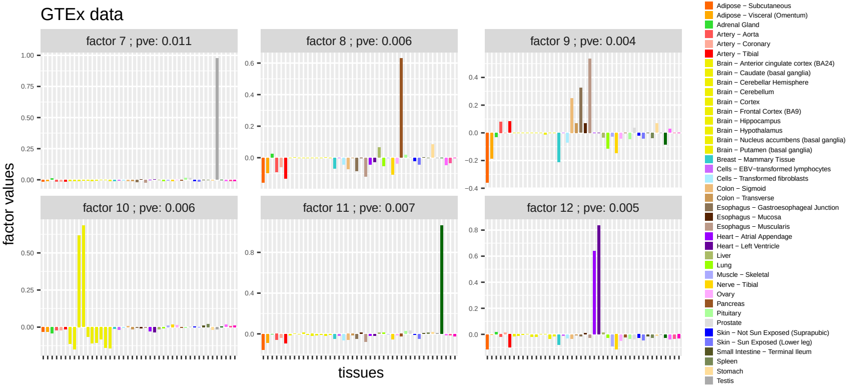
Iterative solution, updating $k = 1, \ldots, K$ using rank 1.

Details: Wang + S. (2018)

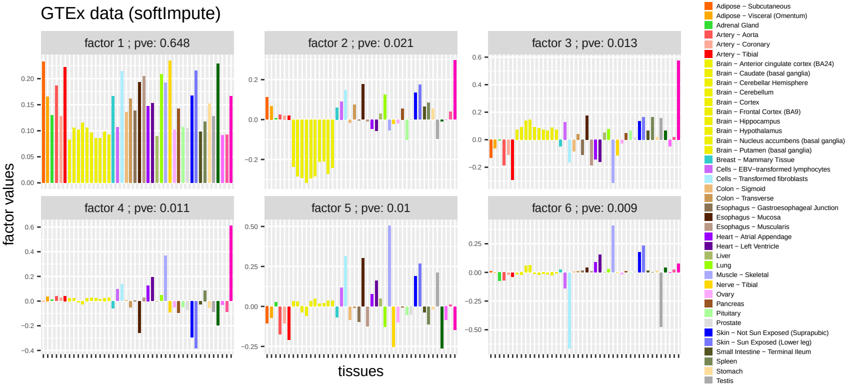
GTEx data: first 6 factors



GTEx data: next 6 factors



Comparison: softImpute (nuclear norm penalty)



Summary

EBNM provides a flexible and convenient way to induce shrinkage and sparsity in a range of applications.

More Details

http://stephenslab.uchicago.edu/publications.html

- Multiple Testing: Efron (200x); S. (2017); Urbut et al (2017), Gerard + S. (2018).
- Linear Regression: Wang (2018); Kim et al (in prep).
- Matrix factorization: Wang and S. (2018).
- Wavelets: Johnstone + Silverman (2004); Xing, Carbonetto + S. (2017).
- Correlation: Dey and S. (2018).