Sparse Horseshoe Estimation via EM

Daniel F. Schmidt
Joint work with Shu Yu Tew, Enes Makalic and Mario Boley

Department of Data Science and AI
Monash University

Aalto University
May 21st, 2025

Outline

1. Global-Local Shrinkage Hierarchies
2. EM Algorithm for Sparse Horseshoe Estimation
3. Results and Extensions

About Monash University

- Largest university in Australia
- I am from the Department of Data Science and AI in the Faculty of IT
- Department has researchers working in:
  - Time series (classification, forecasting)
  - Bayesian machine learning (deep learning)
  - Vision and language
  - Bioinformatics and medical health
- Most famous staff member is Prof. Chris Wallace, inventor of minimum message length
- Department always looking for collaborations, and able to host visitors

Outline

1. Global-Local Shrinkage Hierarchies

2. EM Algorithm for Sparse Horseshoe Estimation

3. Results and Extensions

Bayesian Sparse Estimation (1)

- Consider the usual linear model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \beta_0 \mathbf{1}_n + \boldsymbol{\varepsilon}$$

  where

  - $\mathbf{y} \in \mathbb{R}^n$ is a vector of targets;
  - $\mathbf{X} \in \mathbb{R}^{n \times p}$ s a matrix of features;
  - $\boldsymbol{\beta} \in \mathbb{R}^p$ is a vector of regression coefficients;
  - β_0 is the intercept;
  - $\boldsymbol{\varepsilon} \in \mathbb{R}^n$, $\varepsilon_i \sim N(0, \sigma^2)$, is a vector of normally distributed random errors.

- Aim: Bayesian (sparse) estimation of $\boldsymbol{\beta}$ ($p > n$ is a possibility)

Bayesian Sparse Estimation (2)

- There are a number of approaches (e.g., model selection, ad-hoc thresholding, projection approaches)
- We will focus on the use of sparsity promoting priors
- A prior is sparsity promoting if it concentrates a large amount of prior probability around $\beta_j = 0$
  - Sometimes called the "one-component" approach
- These aim to avoid computational intractability by converting a discrete problem into a continuous one
- The most well known sparsity promoting prior is probably the Laplace distribution (i.e., the "lasso prior")
- We will use the general framework of global-local shrinkage priors

Global-Local Shrinkage Hierarchies (1)

- The global-local shrinkage hierarchy
  ⇒ generalises many popular Bayesian regression priors

$$\begin{aligned}
\mathbf{y} \,|\, \boldsymbol{\beta} &\sim p(\mathbf{y} \,|\, \boldsymbol{\beta}, \ldots) d\mathbf{y}, \\
\beta_j \,|\, \lambda_j^2, \tau^2, \sigma^2 &\sim N(0, \lambda_j^2 \tau^2 \sigma^2) \\
\lambda_j &\sim \pi(\lambda_j) d\lambda_j \\
\tau &\sim \pi(\tau) d\tau
\end{aligned}$$

- Models priors for β_j as scale-mixtures of normals
  ⇒ choice of $\pi(\lambda_j)$, $\pi(\tau)$ controls behaviour of the estimator

Global-Local Shrinkage Hierarchies (2)

- The global-local shrinkage hierarchy
  ⇒ generalises many popular Bayesian regression priors

$$\begin{aligned} \mathbf{y} \,|\, \boldsymbol{\beta} &\sim p(\mathbf{y} \,|\, \boldsymbol{\beta}, \ldots) d\mathbf{y}, \\ \beta_j \,|\, \lambda_j^2, \tau^2, \sigma^2 &\sim N(0, \lambda_j^2 \tau^2 \sigma^2) \\ \lambda_j &\sim \pi(\lambda_j) d\lambda_j \\ \tau &\sim \pi(\tau) d\tau \end{aligned}$$

- Local shrinkers λ_j control selection of variables; e.g.,
  $\lambda_j^2 \sim \mathrm{Exp}(1) \implies$ Bayesian lasso
  $\lambda_j \sim C^+(0,1) \implies$ Bayesian horseshoe
  $\lambda_j \sim \delta_1 \implies$ ridge regression (point mass at $\lambda_j = 1$)

Global-Local Shrinkage Hierarchies (3)

- The global-local shrinkage hierarchy
  ⇒ generalises many popular Bayesian regression priors

$$
\begin{aligned}
\mathbf{y} \,|\, \boldsymbol{\beta} &\sim p(\mathbf{y} \,|\, \boldsymbol{\beta}, \ldots) d\mathbf{y}, \\
\beta_j \,|\, \lambda_j^2, \tau^2, \sigma^2 &\sim N(0, \lambda_j^2 \tau^2 \sigma^2) \\
\lambda_j &\sim \pi(\lambda_j) d\lambda_j \\
\tau &\sim \pi(\tau) d\tau
\end{aligned}
$$

- Global shrinker τ controls overall shrinkage (and multiplicity)

Global-Local Shrinkage Hierarchies (4)

- To see how λ_j and τ affect estimation, consider the linear model

$$\mathbf{y} \,|\, \boldsymbol{\beta} \sim N(\mathbf{X}\boldsymbol{\beta} + \beta_0 \mathbf{1}_n, \sigma^2 \mathbf{I}_n)$$

- If predictors are orthogonal, then conditional on $\lambda_1, \ldots, \lambda_p$, we have

$$\begin{aligned} \mathbb{E}\left[\beta_j \,|\, \lambda_j, \tau\right] &= \left(\frac{\lambda_j^2}{n + \lambda_j^2 \tau^2}\right) \hat{\beta}_j \\ &= (1 - \kappa_j)\hat{\beta}_j \end{aligned}$$

where $\hat{\beta}_j$ is the least-squares estimate; so
  - large λ_j (κ_j near zero) implies little shrinkage;
  - small λ_j (κ_j near one) implies a lot of shrinkage

$\Longrightarrow$ setting τ affects the overall degree of shrinkage

Global-Local Shrinkage Hierarchies (5)

- A weakness of the GLS approach is that standard posterior estimates that are easy to obtain from sampling, such as the posterior mean, are not actually sparse
    - The posterior may concentrate around $\beta_j = 0$, but $\mathbb{E}\left[\beta_j \mid \mathbf{y}\right] \neq 0$
- The posterior mode is frequently sparse
    - For example, the posterior mode of the Bayesian lasso is (essentially) equivalent to the usual lasso procedure
- However, posterior modes are not easily found via posterior sampling
- The expectation-maximization (EM) algorithm can be used

Expectation-Maximization for GLS Hierarchies (1)

- Consider a probability model

$$p(\mathbf{y}, \mathbf{z} \,|\, \boldsymbol{\theta})$$

  where $\mathbf{y}$ is observed data, $\mathbf{z}$ is some unobserved (latent) data and $\boldsymbol{\theta}$ is a vector of model parameters

- The EM algorithm finds a local maxima of the marginal likelihood $p(\mathbf{y} \,|\, \boldsymbol{\theta})$ by iterating the following two steps:

  1. E-step: compute the expected log-likelihood given current values of $\boldsymbol{\theta}$:

$$Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}^{\text{old}}) = \mathbb{E}\left[\log p(\mathbf{y}, \mathbf{z} \,|\, \boldsymbol{\theta}) \,|\, \mathbf{y}, \boldsymbol{\theta}^{\text{old}}\right]$$

  2. M-step: maximize the Q-function to update the parameter estimate

$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}^{\text{old}})$$

Expectation-Maximization for GLS Hierarchies (2)

- The EM algorithm has been applied to GLS for mode finding
- Consider the conditional joint posterior

$$p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \sigma^2 \,|\, \mathbf{y}, \tau^2) \propto p(\mathbf{y} \,|\, \boldsymbol{\beta}, \sigma^2)\pi(\boldsymbol{\beta} \,|\, \boldsymbol{\lambda}^2, \tau^2, \sigma^2)\pi(\boldsymbol{\lambda}^2)$$

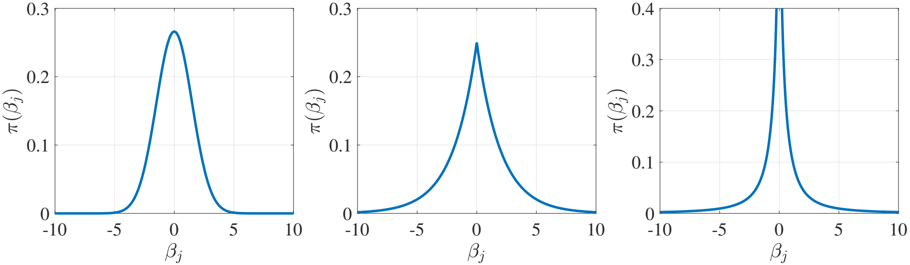
- EM can be applied by treating the local shrinkage parameters $\boldsymbol{\lambda}$ as latent variables, i.e.,

$$\boldsymbol{\beta}_{(t+1)} = \arg\max_{\boldsymbol{\beta}} \mathbb{E}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \sigma^2 \,|\, \boldsymbol{\beta}_{(t)}, \mathbf{y}, \tau^2)\right]$$

  and solving for $\boldsymbol{\beta}_{(t+1)}$
  $\Longrightarrow$ M-step is equivalent to a least-squares update
- There are several drawbacks
  - If you change the prior for $\boldsymbol{\lambda}^2$, the E-step changes
  - Does not give a sound basis for estimating τ^2

The Horseshoe Prior (1)

- The specific case that $\lambda_j \sim C^+(0,1)$ yields the horseshoe prior
- The horseshoe prior has several attractive properties *vis a vis* priors such as the Laplace:
  - Infinitely tall spike at the origin strongly promotes shrinkage of weak effects
  - Heavy, Cauchy-like tails leads to asymptotically (in effect size) unbiased estimates



- Ridge (ℓ_2, left), lasso (ℓ_1, centre) and horseshoe (right)

The Horseshoe Prior (2)

- Effective MCMC algorithms for horseshoe exists
- However, the previous EM algorithm cannot be easily applied to find the mode as the required conditional expectations do not have a closed form and must be numerically approximated
- Bhadra et al (2019) introduced the "horseshoe-like" prior which had similar properties (pole at the origin, heavy tails) but for which the conditional expectations exist

$$\pi(\beta_j \,|\, \tau^2) = \frac{1}{2\pi\tau} \log\left(1 + \frac{\tau^2}{\beta_j^2}\right)$$

- They derived an EM updates for both coefficients $\boldsymbol{\beta}$ and global shrinkage parameter τ^2
- However, conditional expectations are unlikely to be available for extensions such as grouped horseshoe or generalized horseshoe
- Additionally, as noted, the previous formulation does not provide a good basis for estimating the global shrinkage parameter τ^2

Why the Horseshoe Mode?

- The sparse EM procedure converts a discrete optimization problem into a continuous one
- In comparison to MCMC, finding modes is usually much faster as each iteration has similar cost to sampling but far fewer iterations are needed
- If τ^2 is estimated effectively, we avoid the need to cross-validate, with potential computational, as well as philosophical benefits
- We may derive new sparse regularizers from Bayesian priors with interesting properties
- Finally, as researchers, it is simple of interest to solve this problem :)

Outline

1 Global-Local Shrinkage Hierarchies

2 EM Algorithm for Sparse Horseshoe Estimation

3 Results and Extensions

Our EM Algorithm (1)

- To address this we proposed an alternative formulation
- We treat the coefficients, β, as latent variables rather than $\boldsymbol{\lambda}$, i.e., we iterate

$$\left\{\hat{\boldsymbol{\lambda}}^2_{(t+1)}, \hat{\sigma}^2_{(t+1)}\right\} = \underset{\boldsymbol{\lambda}^2,\sigma^2}{\arg\max}\, \mathbb{E}_{\boldsymbol{\beta}}\left[\log p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \sigma^2, \tau^2 \,|\, \hat{\boldsymbol{\lambda}}^2_{(t)}, \hat{\sigma}^2_{(t)}, \tau^2, \mathbf{y})\right]$$

- This might seem odd, but we can recover the coefficients as the posterior mode given $\boldsymbol{\lambda}$, τ and σ is uniquely defined by

$$\hat{\boldsymbol{\beta}}_{\mathrm{MAP}}(\tau^2, \sigma^2, \boldsymbol{\lambda}^2) = \mathbb{E}_{\boldsymbol{\beta}}\left[\boldsymbol{\beta} \,|\, \boldsymbol{\lambda}^2, \sigma^2, \tau^2)\right]$$

i.e., the coefficients and the product $\tau^2\sigma^2\boldsymbol{\lambda}^2$ are in one-to-one correspondence via the conditional posterior mean

Our EM Algorithm (2)

- In this formulation the Q-function is

$$Q(\boldsymbol{\lambda}, \sigma^2 \mid \hat{\boldsymbol{\lambda}}^2_{(t)}, \hat{\sigma}^2_{(t)}, \tau^2) = \mathbb{E}_{\boldsymbol{\beta}}\left[-\log p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \tau^2, \sigma^2 \mid \mathbf{y}) \mid \hat{\boldsymbol{\lambda}}^2_{(t)}, \hat{\sigma}^2_{(t)}, \tau^2, \mathbf{y}\right]$$

$$= \left(\frac{n+p}{2}\right)\log\sigma^2 + \frac{\mathbb{E}_{\boldsymbol{\beta}}\left[||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 \mid \hat{\boldsymbol{\lambda}}^2_{(t)}, \hat{\sigma}^2_{(t)}, \tau^2\right]}{2\sigma^2} + \frac{1}{2}\sum_{j=1}^{p}\log\lambda_j^2$$

$$+ \frac{1}{2\sigma^2\tau^2}\sum_{j=1}^{p}\frac{\mathbb{E}_{\boldsymbol{\beta}}\left[\beta_j^2 \mid \hat{\boldsymbol{\lambda}}^2_{(t)}, \hat{\sigma}^2_{(t)}, \tau^2\right]}{\lambda_j^2} - \log\pi(\boldsymbol{\lambda}^2)$$

  where $\pi(\boldsymbol{\lambda}^2)$ is the prior over $\boldsymbol{\lambda}^2$

- The M-step then minimizes this w.r.t. $\boldsymbol{\lambda}^2$ and σ^2

Our EM Algorithm (3)

- The expectations depend on the conditional posterior of $\boldsymbol{\beta}$:

$$\begin{aligned} \boldsymbol{\beta} \,|\, \boldsymbol{\lambda}, \tau, \sigma, \mathbf{y} &\sim N_p(\mathbf{A}^{-1}\mathbf{X}^T\mathbf{y}, \sigma^2\mathbf{A}^{-1}) \\ \mathbf{A} &= (\mathbf{X}^T\mathbf{X} + \tau^{-2}\boldsymbol{\Lambda}^{-1}) \end{aligned}$$

  where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1^2, \cdots, \lambda_{\mathrm{p}}^2)$.
- Crucially, the conditional posterior is independent of the prior on $\boldsymbol{\lambda}$ $\implies$ E-step remains the same regardless of chosen prior
- This also allows for more complex hierarchical prior structures, such as group shrinkage or horseshoe$+$, etc.

Expectations (1)

- The first expectation is given by

$$\mathbb{E}\left[\beta_j^2 \,|\, \boldsymbol{\lambda}^{(t)}, \tau^{(t)}\right] = \mathrm{Var}[\beta_j] + \mathbb{E}\left[\beta_j\right]^2 \tag{1}$$

- As this is Gaussian, $\mathrm{Var}[\beta_j] = (\mathrm{Cov}[\boldsymbol{\beta}])_{j,j}$, and

$$\begin{aligned} \mathrm{Cov}[\boldsymbol{\beta}] &= \sigma^2 \mathbf{A}^{-1}, \\ \mathbb{E}\left[\beta_j\right] &= \left(\mathbf{A}^{-1}\mathbf{X}^T\mathbf{y}\right)_j. \end{aligned}$$

- In practice we do not invert the matrix directly but use variant of Rue or Bhattarchaya's algorithms

Expectations (2)

- The second expectation is given by

$$\mathbb{E}\left[||\mathbf{y}-\mathbf{X}\boldsymbol{\beta}||^2\right] = ||\mathbf{y}-\mathbf{X}\mathbb{E}\left[\boldsymbol{\beta}\right]||^2 + \mathrm{tr}(\mathbf{X}^T\mathbf{X}\cdot\mathrm{Cov}[\boldsymbol{\beta}])$$

  where $\mathrm{tr}(\cdot)$ is the usual trace operator.

- We also considered approximating $\mathbf{A}$ by its diagonal elements, which yields the approximate expectations

$$\mathrm{Var}[\beta_j] \approx \sigma^2\left(||\mathbf{x}_j||^2+\frac{1}{\tau^2\lambda_j^2}\right)^{-1}$$

$$\mathrm{tr}(\mathbf{X}^T\mathbf{X}\cdot\mathrm{Var}[\boldsymbol{\beta}]) \approx \sigma^2\sum_{j=1}^{p}\left(||\mathbf{x}_j||^2+\frac{1}{\tau^2\lambda_j^2}\right)^{-1}||\mathbf{x}_j||^2.$$

  which are exact for orthogonal designs

Expectations (2)

- The second expectation is given by

$$\mathbb{E}\left[||\mathbf{y}-\mathbf{X}\boldsymbol{\beta}||^2\right] = ||\mathbf{y}-\mathbf{X}\mathbb{E}\left[\boldsymbol{\beta}\right]||^2 + \mathrm{tr}(\mathbf{X}^T\mathbf{X}\cdot\mathrm{Cov}[\boldsymbol{\beta}])$$

  where $\mathrm{tr}(\cdot)$ is the usual trace operator.

- We also considered approximating $\mathbf{A}$ by its diagonal elements, which yields the approximate expectations

$$\begin{aligned}\mathrm{Var}[\beta_j] &\approx \sigma^2\left(||\mathbf{x}_j||^2 + \frac{1}{\tau^2\lambda_j^2}\right)^{-1} \\ \mathrm{tr}(\mathbf{X}^T\mathbf{X}\cdot\mathrm{Var}[\boldsymbol{\beta}]) &\approx \sigma^2\sum_{j=1}^{p}\left(||\mathbf{x}_j||^2 + \frac{1}{\tau^2\lambda_j^2}\right)^{-1}||\mathbf{x}_j||^2.\end{aligned}$$

  which are exact for orthogonal designs

Application to Horseshoe (1)

- We now apply this formulation to the horseshoe
- The horseshoe prior on $\boldsymbol{\lambda}^2$ is

$$\pi(\lambda_j^2) \propto \left(\lambda_j^2\right)^{-1/2} (1 + \lambda_j^2)^{-1}$$

- The EM update, holding σ^2 fixed is then

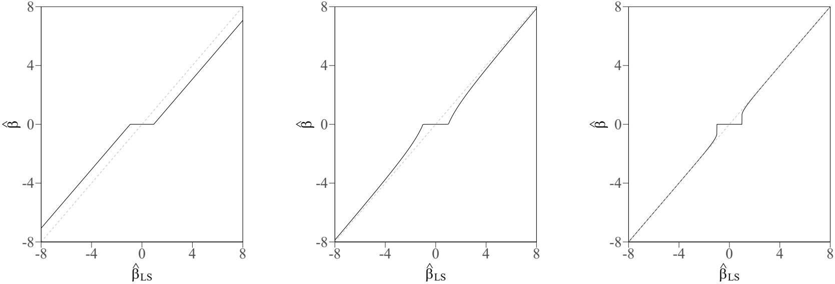
$$\begin{aligned} \hat{\lambda}_j^2 &= \underset{\lambda_j^2}{\operatorname{argmin}} \left\{ \log \lambda_j^2 + \frac{W_j}{\lambda_j^2} + \log(1 + \lambda_j^2) \right\} \\ &= \frac{1}{4} \left(\sqrt{1 + 6W_j + W_j^2} + W_j - 1 \right) \end{aligned}$$

where $W_j = \mathbb{E}\left[\beta_j^2\right]/(2\sigma^2\tau^2)$.

- We then update σ^2 by a numerical search

Application to Horseshoe (2)

- Shrinkage profiles



(a) Lasso **(b)** Horseshoe **(c)** Horseshoe-like

- The posterior mode estimates $\hat{\beta}$ versus $\hat{\beta}_{\mathrm{LS}}$ for the (a) Lasso, (b) Horseshoe and (c) Horseshoe-like estimator. For illustration purposes, τ is chosen such that all three estimators give nearly identical shrinkage within approximately 1 unit of the origin.

Application to Horseshoe (3)

- A strength of this formulation is that it provides a solid basis for also estimating τ^2
- This is because our EM algorithm targets the marginal

$$p(\boldsymbol{\lambda}^2, \sigma^2, \tau^2 \mid \mathbf{y}) = \int p(\boldsymbol{\beta}, \boldsymbol{\lambda}^2, \sigma^2, \tau^2) d\boldsymbol{\beta}$$

- We give τ a half-Cauchy prior and update numerically
- This is an advantage of this EM approach
  - No need to appeal to additional principles such as cross-validation
  - Much faster than sampling and then sparsifying

Outline

1. Global-Local Shrinkage Hierarchies

2. EM Algorithm for Sparse Horseshoe Estimation

3. Results and Extensions

Some Examples

- Does this process actually produce sensible sparse models?
- For purposes of demonstration, I will show a couple of simple examples and some experimental results
- I ran horseshoe MCMC and HS-EM on the (in?)famous diabetes data set from Efron et al (2000)

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6
Mean	-0.009	-18.68	5.769	1.034	-0.223	0.013	-0.592	2.419	48.84	0.179
2.50%	-0.341	-30.93	4.371	0.571	-0.937	-0.342	-1.415	-3.462	32.24	-0.225
97.5%	0.326	-5.144	7.109	1.457	0.098	0.656	0.189	11.36	70.14	0.734
Mode	·	-17.54	5.741	1.021	·	·	-0.909	·	43.58	·

- All coefficients with 95% intervals including zero were omitted; maybe we just made a glorified credible interval thresholder? :)

Smoothing Example (1)

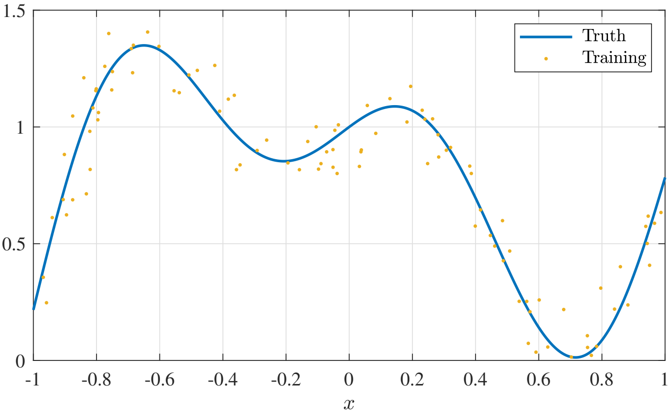
- A simple high dimensional smoothing example
- I generated $n = 100$ noisy observations from the model

$$f(x) = \frac{1}{1 + x^2} + x \cos(5x) + \varepsilon$$

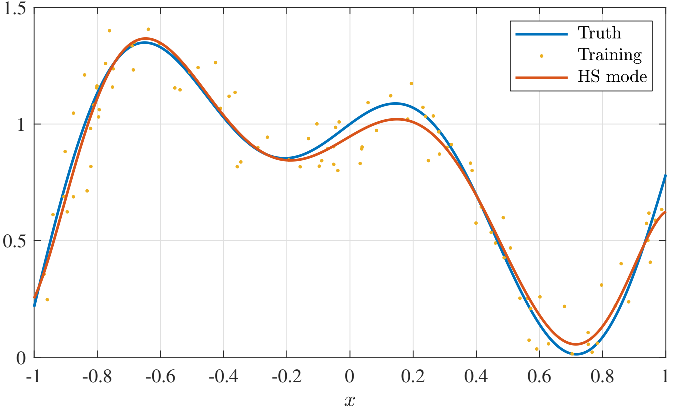
where $\varepsilon \sim N(0, 0.1^2)$

- I then formed a basis matrix of 100 Legendre polynomials and used HS-EM to estimate the coefficients

Smoothing Example (2)

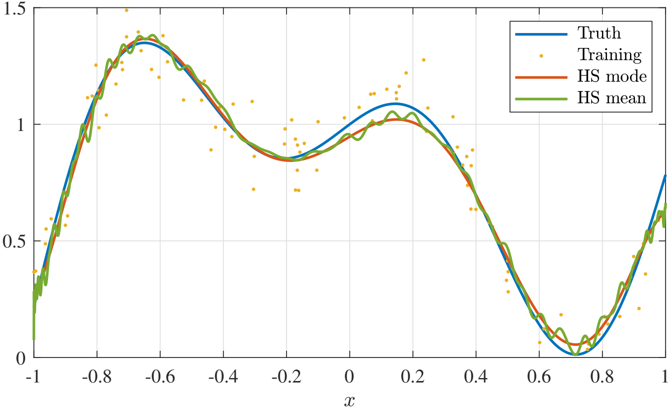


Smoothing Example (3)



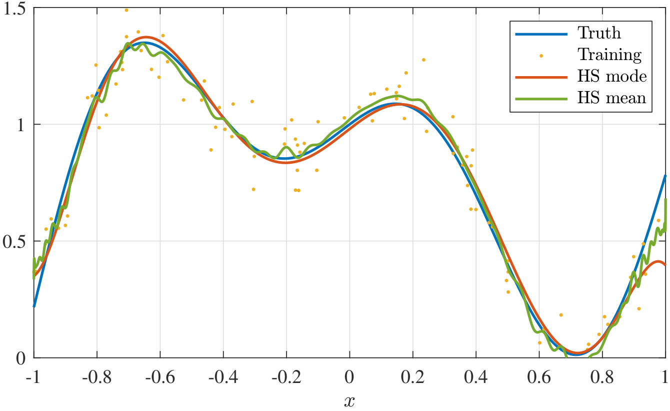
HS-EM retain five Legendre polynomials; $\approx 0.2s$ runtime; MSE ≈ 0.0015

Smoothing Example (4)



Posterior mean from 1,000 samples; $\approx 10s$ runtime; MSE $\approx$ 0.0017

Smoothing Example (5)



Sometimes undersmooths (due to selection); MSE 0.0029 vs 0.0020

CV Experiments (1)

- We ran some other tests versus other non-convex sparsifiers
- Compared HS-EM and HS-EM with the approximate E-step versus
    - The "horseshoe-like" EM
    - Minimax concave penalty (MCP)
    - Smooth clipped absolute deviation (SCAD)
    - Lasso (ℓ_1 penalized regression)
    - Ridge (ℓ_2 penalized regression)
- The latter four all use CV for hyperparameter tuning
- We took some standard datasets, but augmented them with 15 additional correlated noise variables
- We then did CV experiment, using base variables plus all all interactions, squares, cubes and logarithmic transformations to create high dimensional problems

CV Experiments (2)

	HS-EM	HS-apx	HS-like	Lasso	MCP	SCAD	Ridge
Diabetes ($n = 100, p = 385$)							
MSE	**3383.3**(30.5)	3407.2(32.1)	13068(491.1)	3654.4(47.9)	3624.4(66.2)	3667.6(68.5)	4181.6(42.8)
Time	3.01 (0.02)	1.02 (0.02)	0.32 (0.01)	0.29 (0.01)	1.13 (0.01)	1.47 (0.02)	0.74 (0.01)
No.V	1.62 (0.09)	1.54 (0.09)	95.3 (0.29)	4.14 (0.44)	3.68 (0.41)	6.86 (0.74)	385 (0.00)
Boston Housing ($n = 100, p = 473$)							
MSE	26.76(0.71)	**26.72**(0.71)	58.01(3.05)	31.41(0.89)	293.1(24.3)	323.9(29.7)	49.19(1.19)
Time	4.74(0.34)	1.86(0.03)	0.67(0.03)	0.20(0.01)	1.21(0.01)	1.39(0.01)	1.06(0.01)
No.V	2.82(0.16)	2.84(0.16)	47.9(0.78)	4.60(0.52)	4.52(0.41)	10.9(0.87)	473(0.00)
Concrete ($n = 100, p = 327$)							
MSE	**73.71** (3.67)	73.76 (3.66)	222.4 (8.38)	81.67 (1.98)	113.9 (15.9)	108.3 (16.1)	176.2 (2.49)
Time	2.39 (0.17)	0.86 (0.03)	0.36 (0.02)	0.23 (0.01)	0.98 (0.01)	1.14 (0.01)	0.55 (0.01)
No.V	5.46 (0.13)	5.42 (0.14)	67.8 (0.57)	10.7 (0.61)	6.60 (0.44)	13.8 (0.85)	327 (0.00)
Eye ($n = 100, p = 200$)							
MSE	**0.79** (0.06)	**0.79** (0.06)	1.50 (0.23)	0.84 (0.14)	**0.79** (0.06)	0.86 (0.09)	0.85 (0.09)
Time	0.52 (0.02)	0.24 (0.01)	0.15 (0.01)	0.20 (0.01)	0.13 (0.01)	0.19 (0.01)	0.23 (0.01)
No.V	3.84 (0.11)	3.72 (0.09)	0.00 (0.00)	18.0 (0.55)	5.20 (0.27)	10.5 (0.42)	200 (0.00)

CV Experiments (2)

- We undertook other experiments
- General trends observed were:
  - HS-EM and HS-apx tend to produce sparsest models;
  - SCAD and MCP appeared sensitive to correlation in predictors;
  - HS-like appears to overfit; likely due to the way τ^2 is estimated
  - HS-EM is a bit slower than SCAD/MCP, but HS-apx is quite comparable
  - HS-EM and HS-apx do not have much difference in behaviour
- The latter might seem surprising, as the E-step approximation is quite crude if there are a lot of correlated variables
- But, in this case, the HS mode tends to select one of the correlated predictors, shrinking the others to zero; so the diagonal approximation actually becomes quite accurate!

Some Extensions: GLMs

- We adapted the procedure to GLMs
- In this case we used a heteroskedastic normal approximation for the conditional distribution of $\boldsymbol{\beta}$

$$\begin{aligned}\boldsymbol{\beta} \,|\, \boldsymbol{\lambda}, \tau, \boldsymbol{\omega}, \mathbf{z} \;&\sim\; N_p\left(\mathbf{A}_\omega^{-1}\mathbf{X}^T\boldsymbol{\Omega}\mathbf{z},\; \mathbf{A}_\omega^{-1}\right), \\ \mathbf{A}_\omega \;&=\; \mathbf{X}^T\boldsymbol{\Omega}\mathbf{X} + \tau^{-2}\boldsymbol{\Lambda}^{-1}\end{aligned}$$

where $\boldsymbol{\Omega} = \mathrm{diag}(\boldsymbol{\omega})$, $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$ is a vector of weights, and $\mathbf{z}$ is an adjusted version of the targets, $\mathbf{y}$

- For example, for logistic regression we use Polya-Gamma weights:

$$z_i = (y - 1/2)/\omega_i, \;\; \omega_i = \left(\frac{1}{2\eta_i}\right)\tanh\left(\frac{\eta_i}{2}\right),$$

where $\boldsymbol{\eta} = \mathbf{X}\beta + \beta_0\mathbf{1}_n$ is the linear predictor.

Some Extensions: GLMs

- We adapted the procedure to GLMs
- In this case we used a heteroskedastic normal approximation for the conditional distribution of $\boldsymbol{\beta}$

$$\begin{aligned} \boldsymbol{\beta} \,|\, \boldsymbol{\lambda}, \tau, \boldsymbol{\omega}, \mathbf{z} \; &\sim \; N_p\left(\mathbf{A}_{\omega}^{-1}\mathbf{X}^T\boldsymbol{\Omega}\mathbf{z},\; \mathbf{A}_{\omega}^{-1}\right), \\ \mathbf{A}_{\omega} \; &= \; \mathbf{X}^T\boldsymbol{\Omega}\mathbf{X} + \tau^{-2}\boldsymbol{\Lambda}^{-1} \end{aligned}$$

where $\boldsymbol{\Omega} = \mathrm{diag}(\boldsymbol{\omega})$, $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$ is a vector of weights, and $\mathbf{z}$ is an adjusted version of the targets, $\mathbf{y}$

- For example, for logistic regression we use Polya-Gamma weights:

$$z_i = (y - 1/2)/\omega_i, \;\; \omega_i = \left(\frac{1}{2\eta_i}\right)\tanh\left(\frac{\eta_i}{2}\right),$$

where $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \beta_0\mathbf{1}_n$ is the linear predictor.

Some Extensions: Ridge Regression

- We also applied the idea to ridge regression
  - In this case $\boldsymbol{\lambda} = \mathbf{1}_p$, so there are no local shrinkers
- We exploited the properties of the ridge prior and SVD so that the EM procedure which estimates τ^2 and σ^2 has the same complexity as a single LS fit

  $\Longrightarrow$ performance was as good as, or better than, LOOCV ridge regression
- We showed that there is essentially only a single mode for τ^2 and σ^2, and that is easy to find
- In contrast, LOOCV exhibits multiple local minima, and the worst minima can be only marginally better than the worst maxima in terms of risk

Some Extensions: Grouped Horseshoe

- We are now investigating more complex prior structures
- An advantage of the construction is that the E-step is unaffected by these; only the M-step changes
- For example, we investigated grouped half-Cauchy priors:

$$\begin{aligned} \mathbf{y} \,|\, \boldsymbol{\beta} &\sim p(\mathbf{y} \,|\, \boldsymbol{\beta}, \ldots) d\mathbf{y}, \\ \beta_j \,|\, \lambda_j^2, \delta_{g(j)}^2, \tau^2, \sigma^2 &\sim N(0, \lambda_j^2 \delta_{g(j)}^2 \tau^2 \sigma^2) \\ \lambda_j &\sim \pi(\lambda_j) d\lambda_j \\ \delta_{g(j)}^2 &\sim \pi(\delta_{g(j)}^2) \\ \tau &\sim \pi(\tau) d\tau \end{aligned}$$

  where $\delta_{g(j)}^2$ is a "group" shrinker (shared across more than one variable)
- Provides alternative grouped regularizers to grouped lasso, for example

Final Comment: Which Mode?

- We talked about finding posterior modes
- But a good question is: which mode?
- The posterior mode is not invariant under reparamerisation
    - This means EM updates for $\boldsymbol{\lambda}$ instead of $\boldsymbol{\lambda}^2$ lead to different estimators
- We have done some preliminary explorations on different parameterizations, including $\log \boldsymbol{\lambda}$, which yielded interesting results
- Scope for future work ...

Conclusion/Future Work

- If anyone is interested in any of this, feel free to contact me :)
- This presentation was based on material from:
  - "Sparse Horseshoe Estimation via Expectation-Maximisation" (ECML 2022), S.Tew et al. https://arxiv.org/abs/2211.03248 (longer version)
  - "Bayes beats Cross Validation: Efficient and Accurate Ridge Regression via Expectation Maximization", S.Tew et al. (NeuRIPS 2023), https://arxiv.org/abs/2310.18860
  - "Bayesian Shrinkage Methods for Linear Regression", S. Tew (PhD Thesis, Monash University), 2024
- R code for HS-EM available at https://github.com/shuyu-tew/Sparse-Horseshoe-EM
- Python package for fast EM ridge at https://github.com/marioboley/fastridge
- Thank you for your time!