UNIVERSITÀ DEGLI STUDI DI PADOVA

Apprendimento statistico per dati strutturati sparsi

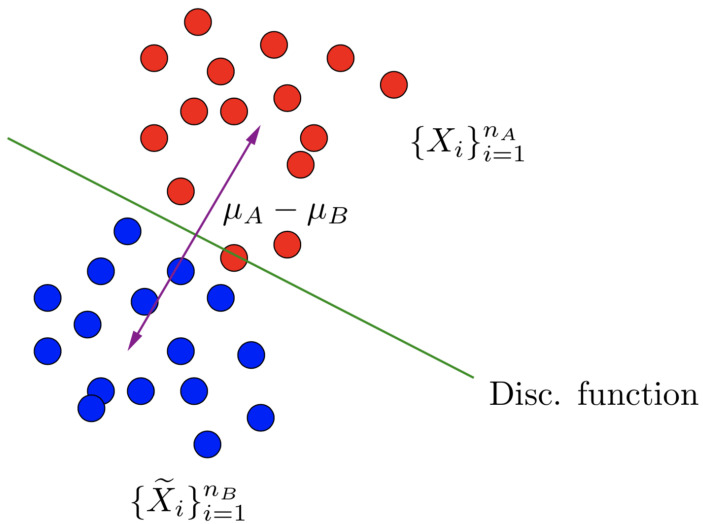
Bruno Scarpa

Stat Data Camp 2021

Classification

Samples $\{X_1, \ldots, X_{n_A}\} \sim \mathbb{P}_A$ from class A and $\{\tilde{X}_1, \ldots, \tilde{X}_{n_B}\} \sim \mathbb{P}_B$ from class B

- Natural decision rule is based on thresholding the log-likelihood ratio $\Psi(x) = \log \frac{\mathbb{P}_B\{x\}}{\mathbb{P}_A\{x\}}$.
- Varying the threshold allows for a trade-off between the two types of errors – namely, deciding $\mathbb{P}_A$ when the true distribution is $\mathbb{P}_B$, and vice versa



Linear discriminant analysis

- Optimal decision boundary in Gaussian case, i.e. $\mathbb{P}_A = \mathcal{N}(\mu_A, \Sigma)$ and $\mathbb{P}_B = \mathcal{N}(\mu_B, \Sigma)$:

$$\Psi(x) = \left\langle \mu_A - \mu_B, \Sigma^{-1} \left(x - \frac{\mu_A + \mu_B}{2} \right) \right\rangle$$

  where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product in $\mathbb{R}^p$.
  Variance Σ is known and shared; means μ_A and μ_B are known.
- If the two classes are equally likely, and x' and x'' are random vectors drawn from the distributions $\mathbb{P}_A$ and $\mathbb{P}_B$, therefore

$$P(\text{classification error}) = \frac{1}{2}\mathbb{P}_A\{\Psi(x') \leq 0\} + \frac{1}{2}\mathbb{P}_B\{\Psi(x'') > 0\}$$

- Given Gaussian assumptions

$$P(\text{classification error}) = \Phi\left(-\frac{\gamma}{2}\right)$$

  where $\gamma = \sqrt{(\mu_A - \mu_B)^\top \Sigma^{-1} (\mu_A - \mu_B)}$ and $\Phi(t) = P(Z \leq t)$ is the cumulative distribution function of a standard normal variable.

Linear discriminant analysis

"Plug-in" principle: substitute estimates $\{\mu_A, \mu_B, \Sigma\}$ from given sample (Fisher linear discriminant function):

$$\hat{\Psi}(x) = \left\langle \hat{\mu}_A - \hat{\mu}_B, \hat{\Sigma}^{-1} \left(x - \frac{\hat{\mu}_A + \hat{\mu}_B}{2} \right) \right\rangle$$

[given that $\hat{\Sigma}$ is invertible, i.e., $n_A > p$ and $n_B > p$]

Classical analysis (simple case in which $\Sigma = I_{p \times p}$):

$$P(\text{classification error}) \xrightarrow{n \to +\infty} \underbrace{\Phi\left(-\frac{\|\mu_A - \mu_B\|_2}{2} \right)}_{\text{Tail function of standard normal}}$$

Linear discriminant analysis

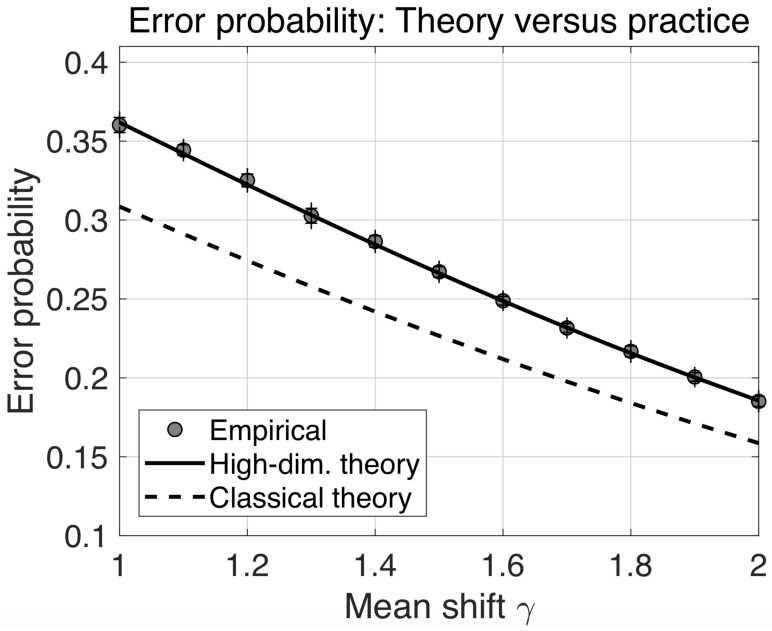
What happens if $(n_A, n_B, p) \to +\infty$ with

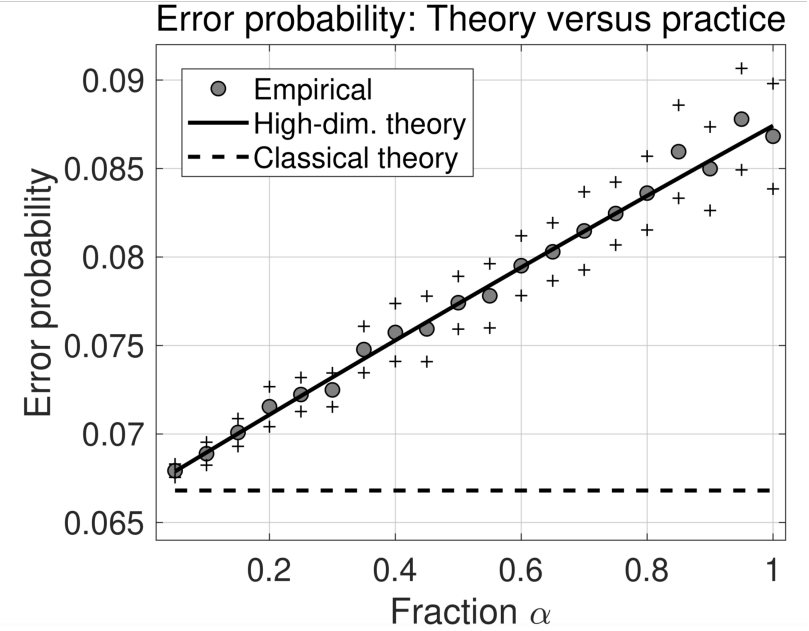
$$\frac{p}{n_A} \to \alpha, \quad \frac{p}{n_B} \to \alpha \qquad \alpha > 0$$

$$P(\text{classification error}) \xrightarrow{n \to +\infty} \Phi\left(-\frac{\gamma^2}{2\sqrt{\gamma^2 + 2\alpha}}\right)$$

- if $\frac{p}{n_i} \to 0$, then the asymptotic error probability is simply $\Phi(-\frac{\gamma}{2})$,as is predicted by classical scaling
- when $\frac{p}{n_i} \to \alpha > 0$, the asymptotic error probability is strictly larger than the classical prediction: $\frac{\gamma^2}{2\sqrt{\gamma^2+2\alpha}}$ shifted towards zero.

Error probability vs mean shift $\gamma = \|\mu_A - \mu_B\|_2$





$$p = 400, \quad \alpha = 0.5 \Rightarrow n_A = n_B = 800, \quad 50 \text{ trials}$$

Kolmogorov prediction: $\Phi\left(-\frac{\gamma^2}{2\sqrt{\gamma^2 + 2\alpha}}\right)$

Classical prediction: $\Phi\left(-\frac{\gamma}{2}\right)$

High-dimensional phenomena are unavoidable

For the classification problem

If the ratio $\frac{p}{n}$ stays bounded strictly above zero, then it is not possible to achieve the optimal classification rate.

Any help?

What can help in the high-dimensional setting?

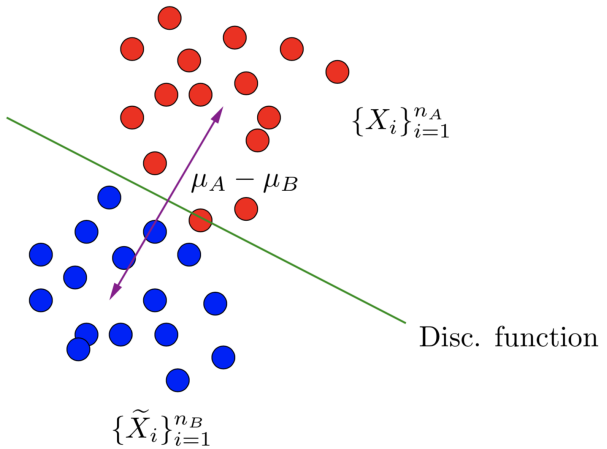
Our only hope is that the data is endowed with some form of low-dimensional structure

- Much of high-dimensional statistics proposes the construction of models of high-dimensional phenomena involving some implicit form of low-dimensional structure
- and then studying the statistical and computational gains afforded by exploiting this structure.

Let us see how the behavior can change dramatically when low-dimensional structure is present.

Structures in parametric classification

- Recall the simple classification problem



- Setting $n = n_A = n_B$, let us recall the scaling in which the ratios $\frac{p}{n_j}$ ($j \in \{A, B\}$) are fixed to some number $\alpha \in (0, \infty)$.
- What is the underlying cause of the inaccuracy of the classical prediction discussed above?
- Denote $\hat{\mu}_j$ the sample mean of the n_j samples
- the squared Euclidean error $\|\hat{\mu}_j - \mu_j\|_2^2$ concentrate sharply around $\frac{p}{n_j} = \alpha$: there is a constant level of error, for which reason the classical prediction $\Phi\left(-\frac{\gamma}{2}\right)$ of the error rate is overly optimistic.

Sparsity in vectors

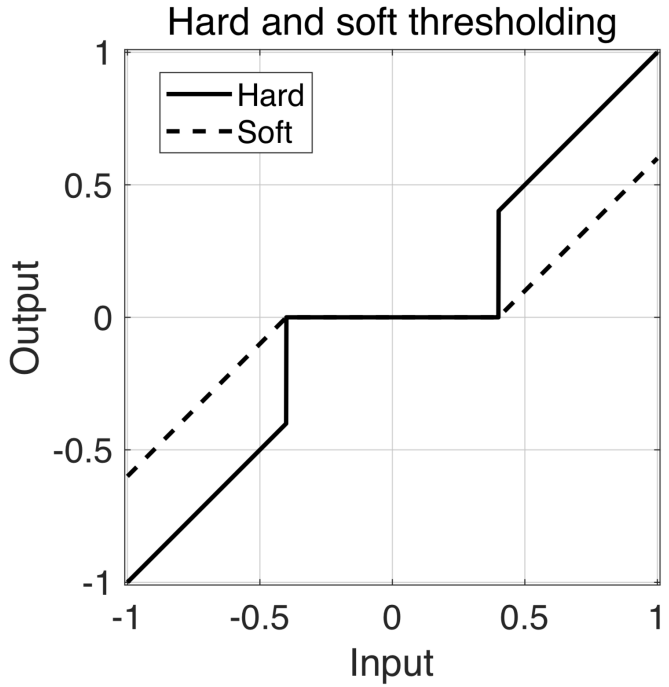
. . . but the sample mean is not the only possible estimate of the true mean

When the true mean vector is equipped with some type of low-dimensional structure, there can be much better estimators

- Perhaps the simplest form of structure is sparsity
- suppose that we knew that each mean vector μ_j were relatively sparse, with only s of its p entries being non-zero, for some sparsity parameter $s \ll p$.
- In this case, we can obtain a substantially better estimator by applying some form of thresholding to the sample means, $\tilde{\mu} = H_\lambda(\hat{\mu})$
- For a given threshold level $\lambda > 0$, the hard-thresholding and the soft-thresholding estimatord are given by

$$H_\lambda(x) = \begin{cases} x & \text{if } |x| > \lambda, \\ 0 & \text{otherwise} \end{cases} \qquad T_\lambda(x) = \begin{cases} x - \lambda \text{ sign}(x) & \text{if } |x| > \lambda, \\ 0 & \text{otherwise} \end{cases}$$

Sparsity in vectors



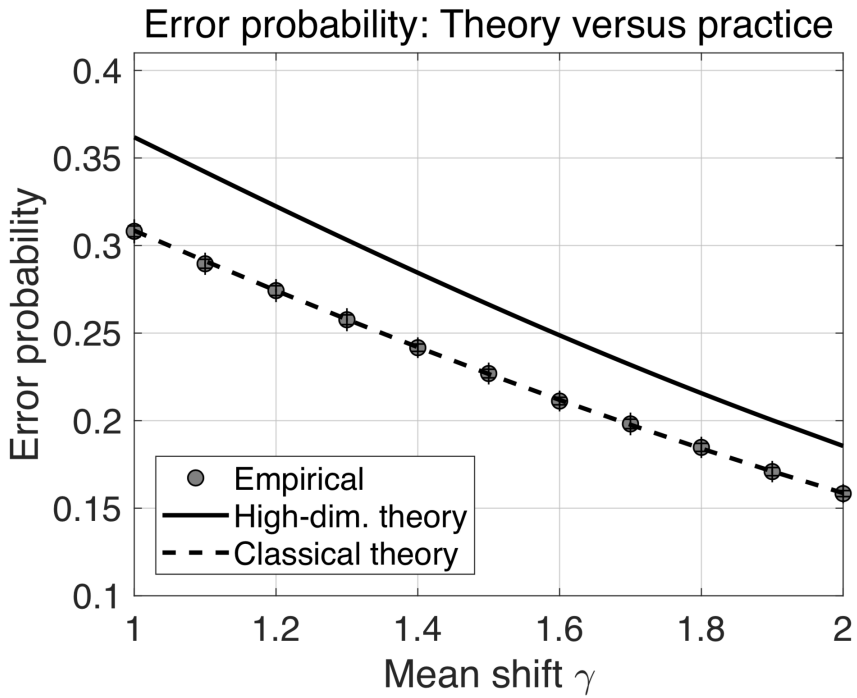
Using the thresholded estimates $\tilde{\mu}$, we can then implement a classifier based on the linear discriminant

$$\tilde{\Psi}(x) = \left\langle \tilde{\mu}_A - \tilde{\mu}_B, x - \frac{\tilde{\mu}_A + \tilde{\mu}_B}{2} \right\rangle$$

Sparsity in vectors

$$H_{\lambda_n}(x) = \begin{cases} x & \text{if } |x| > \lambda_n, \\ 0 & \text{otherwise} \end{cases}$$

- hard-threshold ('optimal' choice) $\lambda_n = \sqrt{\frac{2 \log p}{n}}$
- $p = 400$
- $\alpha = 0.5 \Rightarrow n_A = n_B = 800$
- 50 trials
- $s = 5$ non-zero entries for each means vector



Classical limit prediction is exact whenever the ratio $\frac{\log \binom{p}{s}}{n} \to 0$

The lasso

The lasso

The lasso is a shrinkage method that acts in a nonlinear manner on the outcome y.
The lasso is the solution (β_0, β) to the optimization problem

$$\min_{\beta_0,\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s$$

or in matrix form

$$\min_{\beta_0,\beta} \left\{ \frac{1}{2n} (\mathbf{y} - \beta_0\mathbf{1} - \mathbf{X}\beta)^\top (\mathbf{y} - \beta_0\mathbf{1} - \mathbf{X}\beta) \right\} \quad \text{subject to} \quad \|\beta\|_1 \leq s$$

The Lagrangian form is

$$\min_{\beta_0,\beta} \left\{ \frac{1}{2n} (\mathbf{y} - \beta_0\mathbf{1} - \mathbf{X}\beta)^\top (\mathbf{y} - \beta_0\mathbf{1} - \mathbf{X}\beta) + \lambda\|\beta\|_1 \right\}$$

The lasso

- For convenience, we assume that the outcome values y_i have been centered $\Rightarrow \frac{1}{n}\sum_{i=1}^{n} y_i = 0$.
- Thus, we can omit the intercept term β_0
- $\hat{\beta}$ is the same, and the intercept $\hat{\beta}_0$ is given by $\hat{\beta}_0 = \bar{y} - \sum_{j=1}^{p} \bar{x}_j \hat{\beta}_j$, where $\bar{y}$ and $\bar{x}_j$, $j = 1, \ldots, p$ are the original means.
- The Lagrangian may be rewritten

$$\min_{\beta} \left\{ \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_1 \right\}$$

The lasso

From theory of convex analysis, necessary and sufficient conditions for a solution to problem of minimum in Lagrange form is

$$-\frac{1}{n}\langle \mathbf{x}_j, \mathbf{y} - \mathbf{X}\beta \rangle + \lambda t_j = 0, \quad j = 1, \ldots, p$$

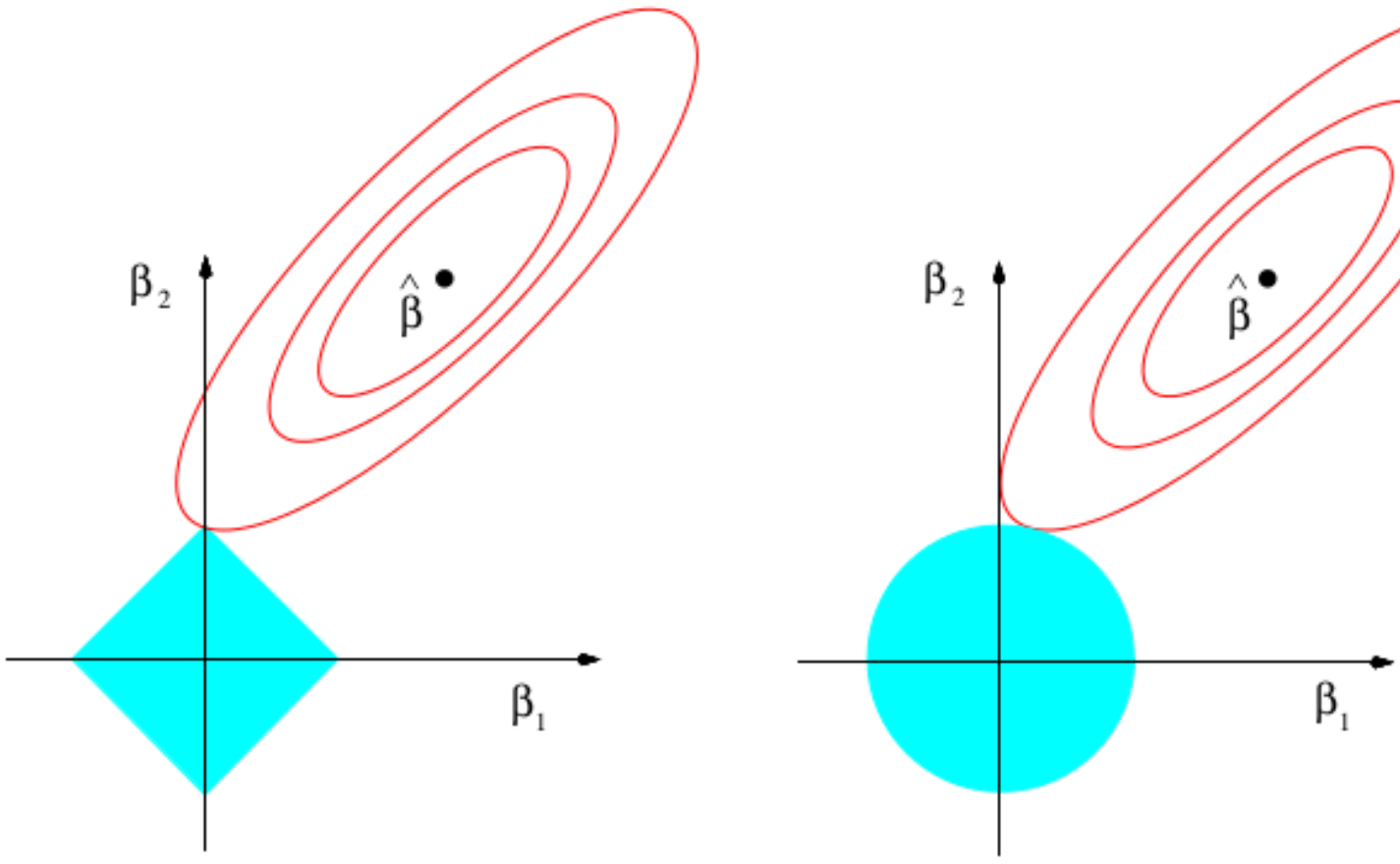
where t_j is an unknown quantity equal to $\operatorname{sign}(\beta_j)$ if $\beta_j \neq 0$ and some value lying in $[-1, 1]$ otherwise (the subgradient for the absolute value function).

Sparsity

A model is sparse when it has few nonzero coefficients.

- A key property of the ℓ_1-constraint is its ability to yield sparse solutions.
- This idea can be applied in many different statistical models,

Picture of lasso and ridge regression



Degrees of freedom for lasso

- Suppose we have p predictors and we fit a linear regression model using only a subset of k of these predictors.
- If these k predictors were chosen without regard to the response variable, the fitting procedure "spends" k degrees of freedom (i.e., error deviance has a χ^2 distribution with k degrees of freedom, with the error variance σ^2 assumed to be known)
- If the k predictors were chosen using knowledge of the response variable, e.g., to yield the smallest training error among all subsets of size k, then we would expect that the fitting procedure *spends* more than k degrees of freedom.
- best subset regression, forward and backward stepwise and *lasso* are example of this type (sometimes called *adaptive* procedure).
- Therefore, in general, degrees of freedom are not the number of nonzero coefficients in the fitted model

Degrees of freedom for lasso

- However, it turns out that for the lasso, one can count degrees of freedom by the number of nonzero coefficients.
- If the n sample predictions are denoted by $\hat{y}$, we define degrees of freedom

$$\mathrm{df}(\hat{y}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{cov}\{\hat{y}_i, y_i\}$$

- i.e., degrees of freedom corresponds to the total amount of *self-influence* that each response measurement has on its prediction.
- The more the model fits (adapts) to the data, the larger the degrees of freedom
- Under *adaptive fitting*, it is typically the case that the degrees of freedom is larger than k.

Degrees of freedom result

For the lasso, with a fixed penalty parameter λ

$$\mathsf{df}(\hat{y}) = \frac{1}{\sigma^2}\sum_{i=1}^{n}\mathsf{cov}\{\hat{y}_i, y_i\} = k$$

In orthogonal case, it is proved by using the fact that the lasso estimates are soft-thresholded versions of the univariate regression coefficients.

Proof is an application of the Stein's unbiased risk estimate (SURE)

- Suppose that $g : \mathbb{R}^n \to \mathbb{R}^n$ is almost differentiable and set

$$\nabla \cdot g = \sum_{i=1}^{n}\frac{\partial g_i}{\partial x_i}$$

- If $y \sim \mathcal{N}_n(\mu, \sigma^2 I)$, then Stein's formula states that

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}\mathsf{cov}\{g_i, y_i\} = \mathbb{E}\{\nabla \cdot g(y)\}$$

- Left hand side is degrees of freedom. Set $g(\cdot)$ equal to the *lasso* estimate.
- In orthogonal case, $\frac{\partial g_i}{\partial x_i}$ is 1 if predictor is in model and 0 otherwise. Hence right hand side equals number of predictors in model ($= k$).
- Non orthogonal case is much harder.

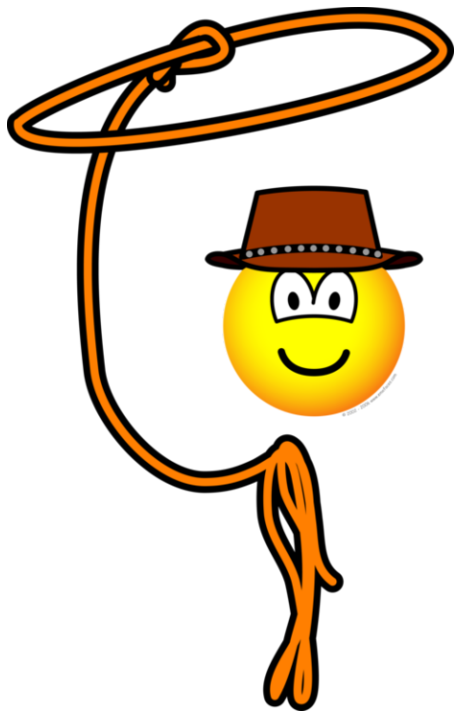
Degrees of freedom – comments

Why the lasso have this simple degrees of freedom property?

- The lasso not only selects predictors, which inflates the degrees of freedom,
- . . . but also shrinks their coefficients toward zero, relative to the usual least-squares estimates.
  This shrinkage turns out to be just the right amount to bring the degrees of freedom down to k.

$\Rightarrow$ a qualitative measure of the 'amount of fitting' that we have done at any point along the lasso path.

Lasso for Generalized linear models



Generalized linear model

- Response variable member of the exponential family
- Goal: to introduce a way to express the relationship between x and the conditional mean $\mathbb{E}\{Y|x\}$.
- Link function: there is regular monotone function $g : \mathbb{R} \to \mathbb{R}$

$$g(\mathbb{E}\{Y|x\}) = \underbrace{\beta_0 + \beta^\top x}_{\eta(x)}, \qquad \mathbb{E}\{Y|x\} = g^{-1}(\eta) = g^{-1}(\beta_0 + \beta^\top x)$$

- The estimate of the β_r's are obtained via maximum likelihood.

$$\min_{\beta_0,\beta} \left\{ -\frac{1}{n}\ell(\beta_0, \beta; y, x) \right\}$$

  where $\ell(\cdot, \cdot; y, x)$ is the log-likelihood function.
- In general no explicit form is available for the estimate, and a numerically iterative procedure is needed: the Fisher scoring algorithm (variants of the Newton-Raphson algorithm)
- When $p > n$, any linear model is over-parametrized, and regularization is needed to achieve a stable fit.

Logistic regression

- Binary response $Y \in \{0, 1\}$
- Goal: to introduce a way to express the relationship between x and the probability of success $\pi = P(Y = 1|x)$
- Link function: there is regular monotone function $g(\cdot)$ such that

$$g(\pi) = \underbrace{\beta_0 + \beta^\top x}_{\eta}, \qquad \pi = g^{-1}(\eta) = g^{-1}(\beta_0 + \beta^\top x)$$

- A common choice is

$$g(\pi) = \log \frac{\pi}{1-\pi}, \qquad \pi = \frac{e^\eta}{1+e^\eta} = \frac{\exp(\beta_0 + \beta^\top x)}{1 + \exp(\beta_0 + \beta^\top x)}$$

that are the *logit* and logistic function, respectively;

Lasso: regularize log-likelihood

To introduce a lasso estimate we penalize the log-likelihood

$$-\frac{1}{n}\sum_{i=1}^{n}\{y_i \log P(Y=1|x_i) + (1-y_i)\log P(Y=0|x_i)\} + \lambda\|\beta\|_1$$

$$= -\frac{1}{n}\sum_{i=1}^{n}\left\{y_i(\beta_0+\beta^\top x_i) - \log\left(1+e^{\beta_0+\beta^\top x_i}\right)\right\} + \lambda\|\beta\|_1$$

Optimization

- The objective function is convex and the log-likelihood part is differentiable, so in principle finding a solution is a standard task in convex optimization
- Coordinate descent is attractive and efficient for this problem
- Typical packages uses a proximal-Newton iterative approach, which repeatedly approximates the negative log-likelihood by a quadratic function
- With the current estimate $\tilde{\beta}_0, \tilde{\beta}$, form the quadratic function

$$Q(\beta_0, \beta) = \frac{1}{2n} \sum_{i=1}^{n} (z_i - \beta_0 - \beta^\top x_i)^2 w_i + C(\tilde{\beta}_0, \tilde{\beta})$$

where C denotes a term independent of (β_0, β), and

$$z_i = \tilde{\beta}_0 + \tilde{\beta}^\top x_i + \frac{y_i - \tilde{p}(x_i)}{\tilde{p}(x_i)(1 - \tilde{p}(x_i))}, \qquad \text{and} \qquad w_i = \tilde{p}(x_i)(1 - \tilde{p}(x_i))$$

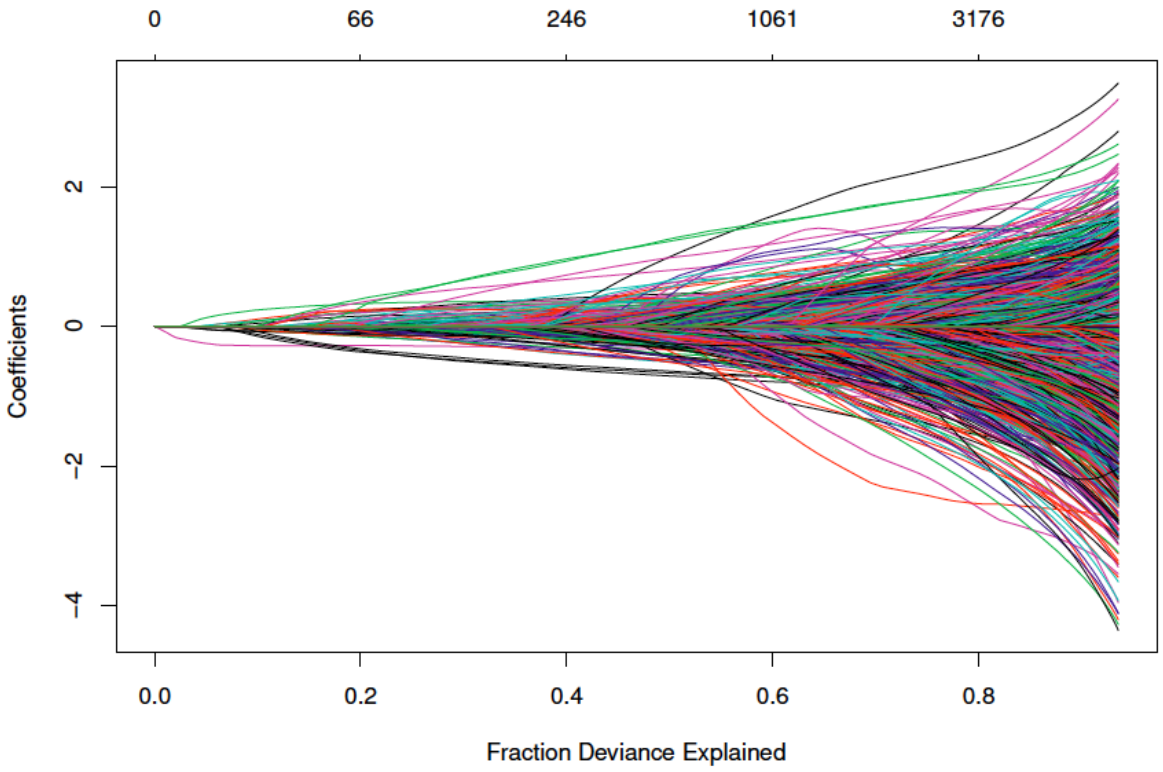
with $\tilde{p}(x_i)$ being the current estimate for $Pr(Y = 1|x_i)$.

- Each outer loop then amounts to a weighted lasso regression.

Document classification

- 20-Newsgroups corpus (Lang 1995).
- positive class: the 10 groups with names of the form `sci.*`, `comp.*` and `misc.forsale`,
- the rest are the negative class.
- The feature set consists of trigrams, with message headers skipped, no stoplist, and features with less than two documents omitted.
- $n = 11314$ documents and $p = 777811$ features, with 52% in the positive class
- Only 0.05% of the features are nonzero for any given document.

Document classification



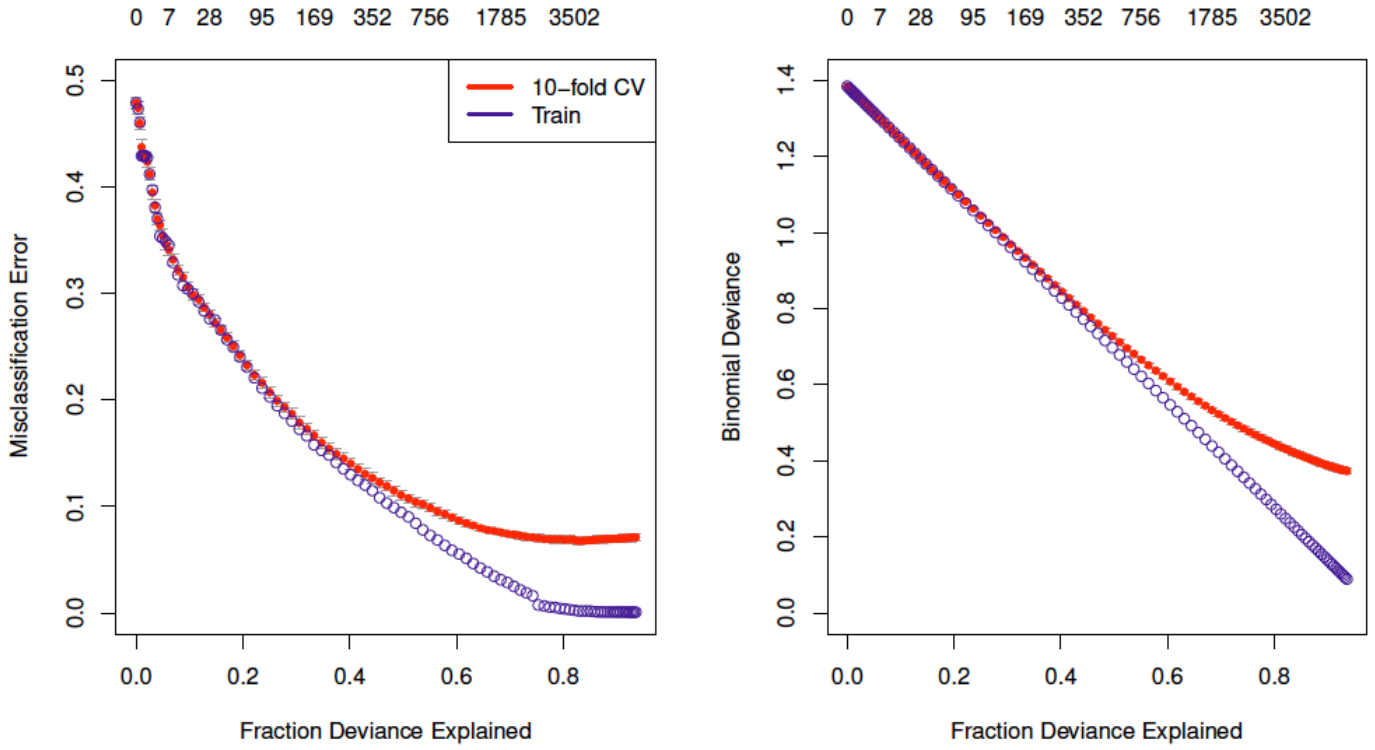
fraction of deviance explained on the training data:

$$D^2_\lambda = \frac{Dev_{null} - Dev_\lambda}{Dev_{null}}$$

- Dev_λ is defined as minus twice the difference in the log-likelihood for a model fit with parameter λ and the "saturated" model (having $\hat{y} = y_i$).
- Dev_{null} is the null deviance computed at the constant (mean) model.

Document classification

- The maximum number of nonzero coefficients in any of these models can be shown to be $\min(n, p)$, which is equal $n = 11314$ in this case.
- Results of tenfold cross-validation for these data, as well as training error.



- The number of nonzero coefficients in each model is shown along the top of each plot

Multivariate logistic regression

- Suppose, instead of a binary response, we have an outcome with multiple possible categories (i.e., that follows a multinomial response, $y_i \in \{1, ..., K\}$)
- Response categorical with K classes (from 1 to K)

$$\pi_k(x) = P(Y = k|x) = \frac{e^{\beta_{0k}+\beta_k^\top x}}{\sum_{r=1}^{K} e^{\beta_{0r}+\beta_r^\top x}}, \qquad \text{for } k = 1, \ldots, K$$

- This model is over specified, since we can add the linear term $\gamma_0 + \gamma^\top x$ to the linear model for each class, and the probabilities are unchanged.
- Set one of the class models to zero - often the first or last class - leading to a model with $K - 1$ linear functions to estimate (each a contrast with the last class)
- Data follow a multinomial distribution
- $\pi_0(x), \ldots, \pi_K(x)$ are the parameters of the multinomial distribution (probability of different way to allocate n observations in K classes).

Lasso: regularize log-likelihood

- Consider the over-specified (redundant) model
- Penalize the log-likelihood

$$-\frac{1}{n}\sum_{i=1}^{n}\log P\left(Y=y_i|x_i;\{\beta_{0k},\beta_k\}_{k=1}^{K}\right)+\lambda\sum_{k=1}^{K}\|\beta_k\|_1$$

- Denote by R the $n\times K$ indicator response matrix with elements $r_{ik}=I(y_i=k)$.
- The penalized log-likelihood become

$$-\frac{1}{n}\sum_{i=1}^{n}\left[\sum_{k=1}^{K}r_{ik}(\beta_{0k}+\beta_k^{\top}x_i)-\log\left\{\sum_{k=1}^{K}e^{\beta_{0k}+\beta_k^{\top}x_i}\right\}\right]+\lambda\sum_{k=1}^{K}\|\beta_k\|_1$$

- Algorithm: proximal-Newton approach. Here we hold all but one class parameters fixed when making the quadratic approximation.

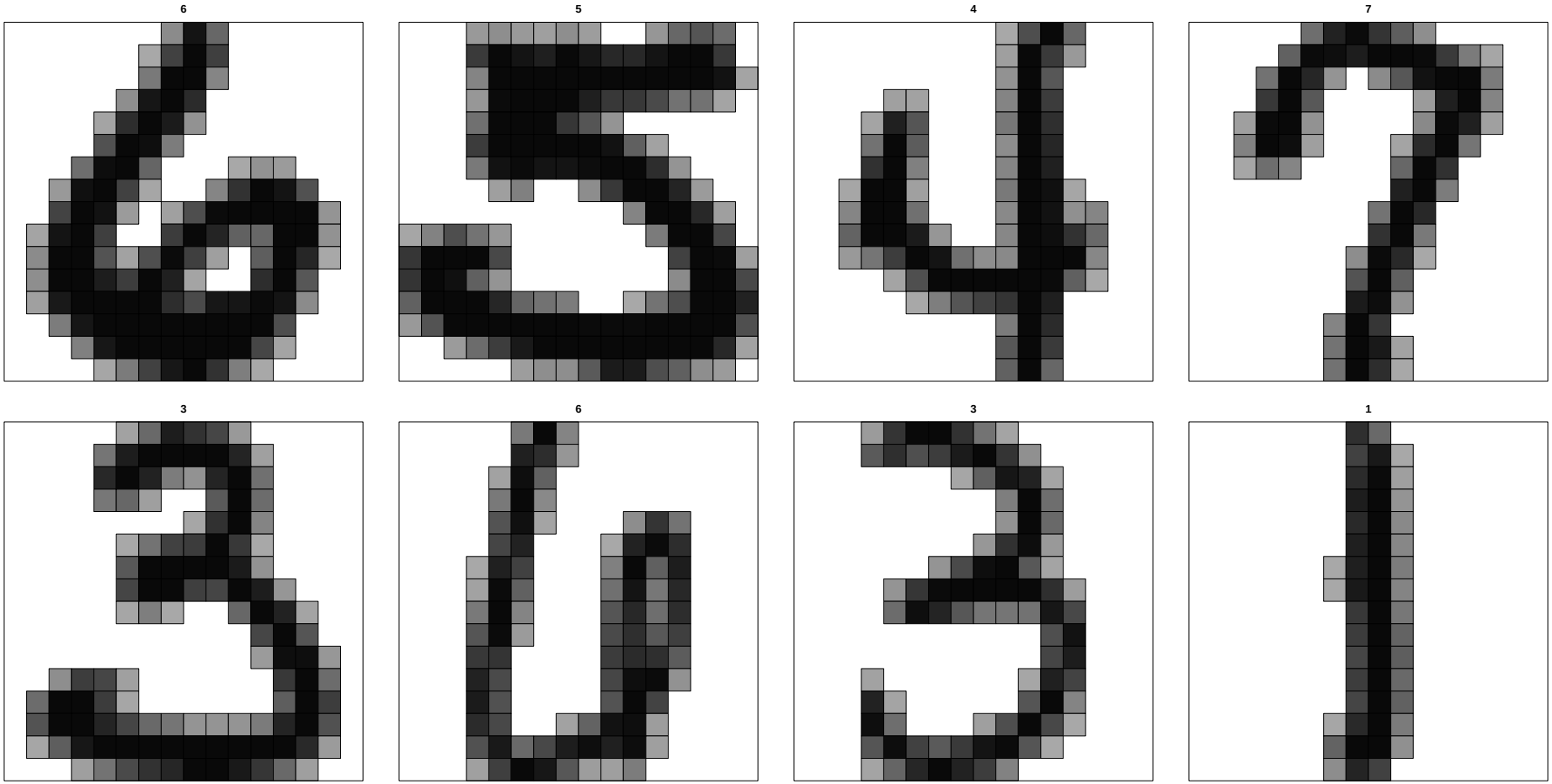
Lasso: penalization and identifiability

- With penalized regression, the restriction is not necessary
- For example, suppose $K = 2$; then $\beta_{1j} = 1, \beta_{2j} = -1$ produces the exact same $\{\pi_{ik}\}$ as $\beta_{1j} = 2, \beta_{2j} = 0$
- As it is impossible to tell the two models apart (and an infinite range of other models), we cannot estimate $\{\beta_k\}$
- With, say, a ridge penalty, this is no longer the case, as $\sum_k \beta_{ij}^2 = 2$ in the first situation and 4 in the second; the proper estimate is clear
- A similar phenomenon occurs for the lasso penalty, although of course there is now the possibility of sparsity, perhaps with respect to multiple classes

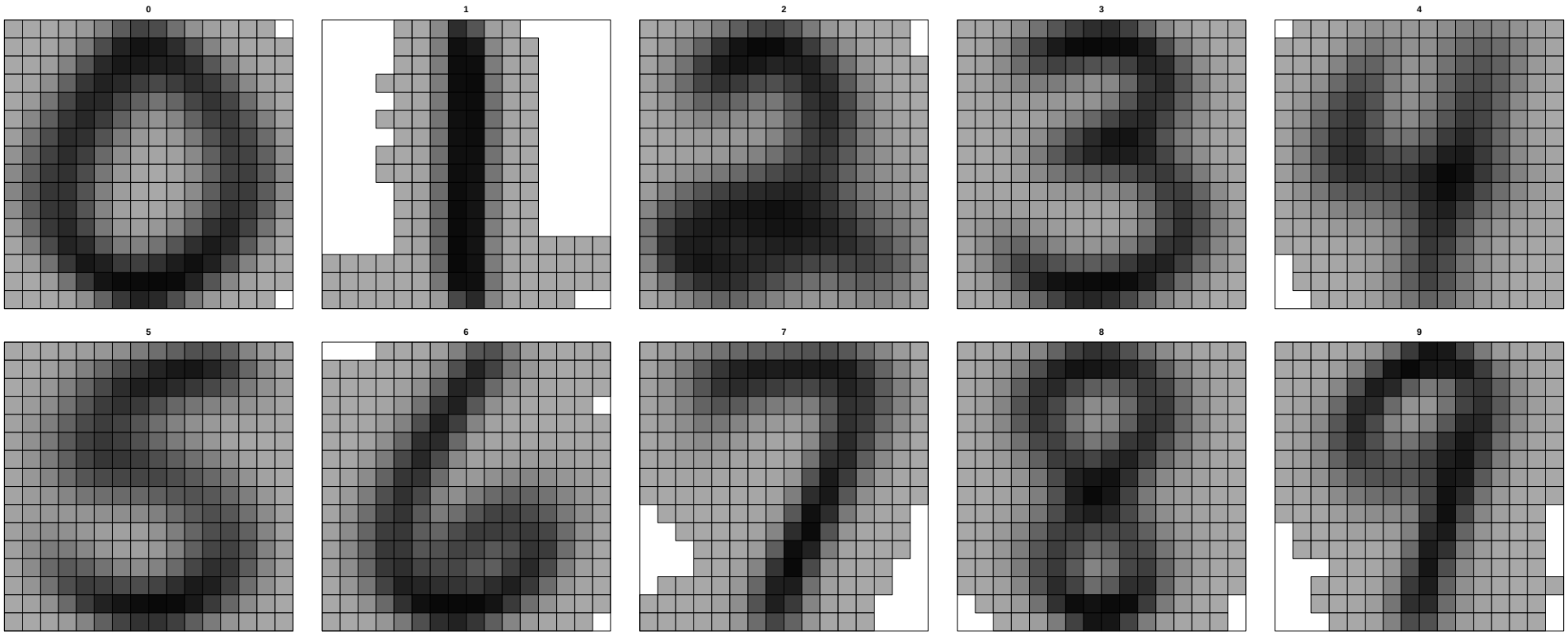
Handwritten digits

- US post-office handwritten digits data (Le Cun et al., 1990)
- Training set: $n = 7291$ training images of the digits 0, 1, . . . , 9
- Test set: $n = 2007$ test images of the digits 0, 1, . . . , 9
- digitized to a 16×16 gray-scale image $\Rightarrow$ features are $p = 256$ pixels
- Fit a 10-class lasso multinomial model

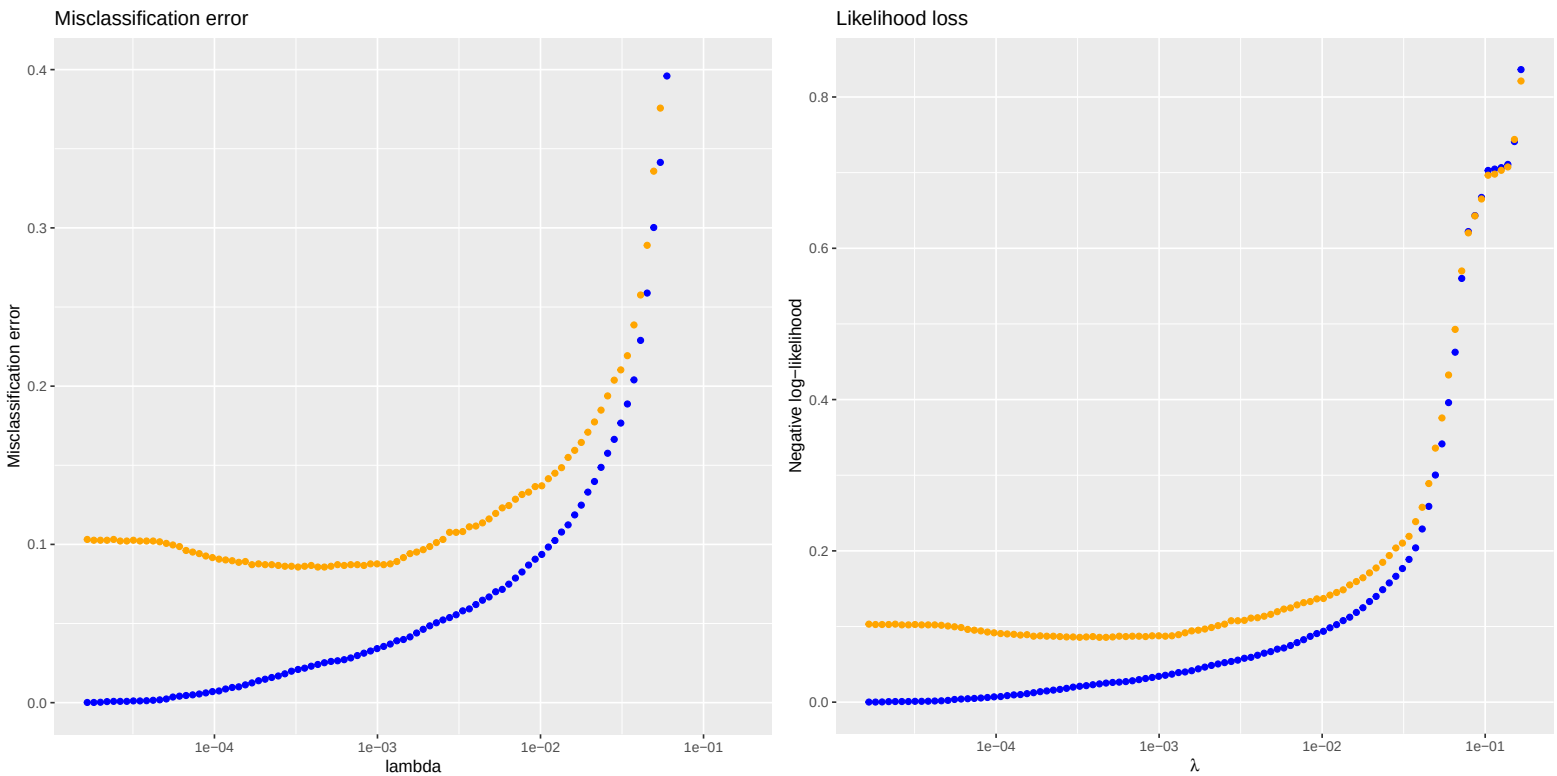
Handwritten digits: some digit



Handwritten digits: averages



Handwritten digits: lasso fit

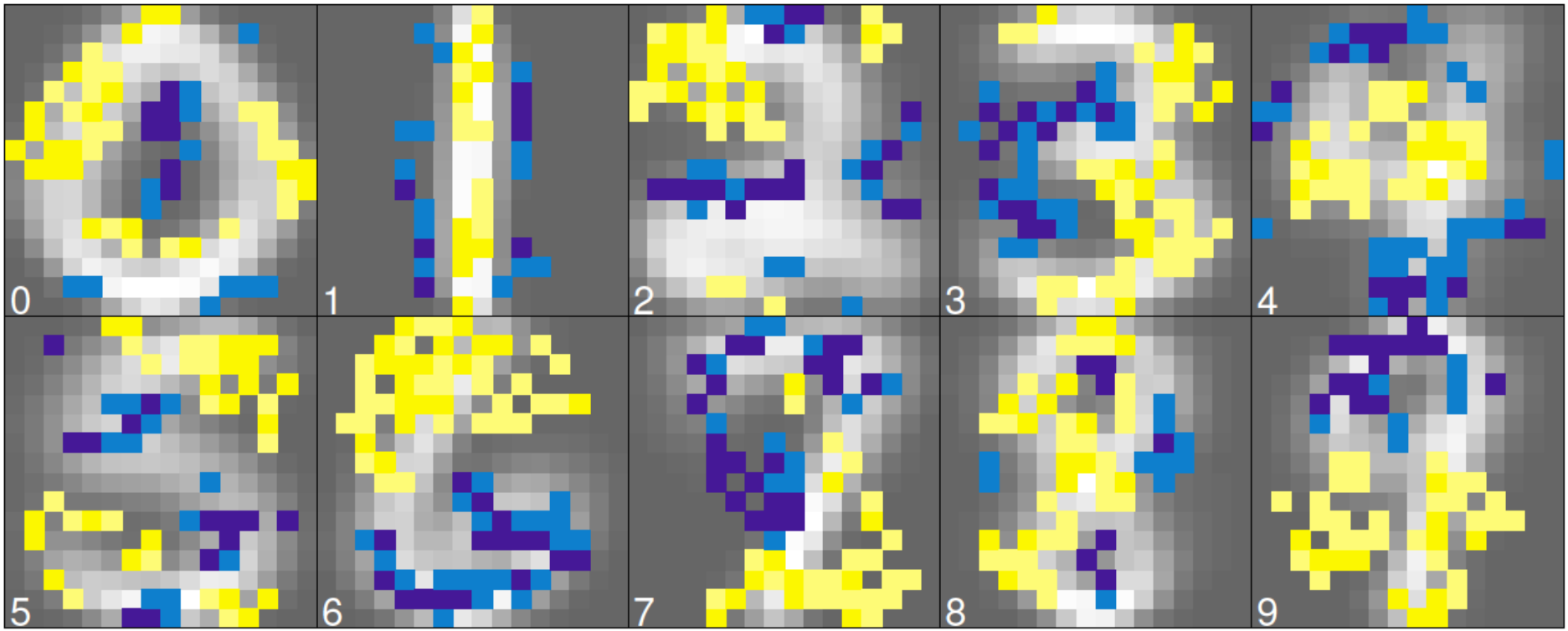


Handwritten digits

	actual									
predicted	0	1	2	3	4	5	6	7	8	9
0	349	0	3	1	2	4	2	0	6	0
1	0	252	0	0	3	0	0	1	0	0
2	0	0	170	5	7	0	2	1	5	1
3	3	3	4	145	0	8	0	1	4	0
4	3	3	8	2	177	3	4	6	2	2
5	0	0	2	7	1	141	4	0	6	1
6	1	4	3	0	3	1	157	0	1	0
7	0	0	1	1	1	0	0	132	0	2
8	2	1	6	3	0	0	1	1	140	2
9	1	1	1	2	6	3	0	5	2	169

Overall classification accuracy: 91% (random accuracy: 10%)

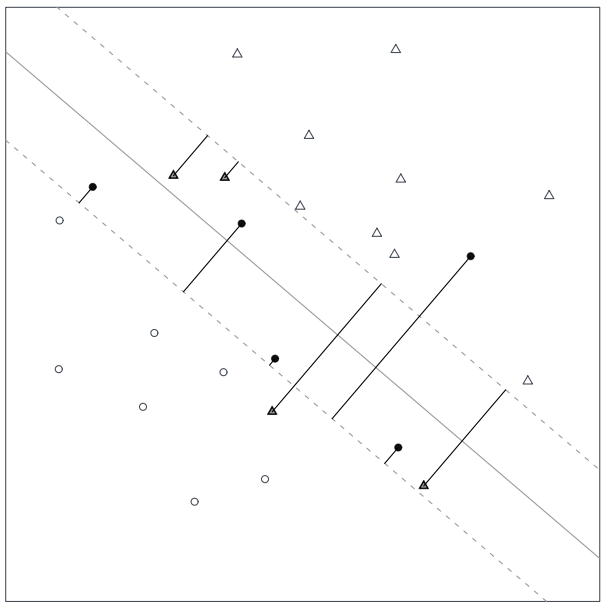
Handwritten digits



- Coefficients as images (on average about 25% are nonzero).
- Gray background image: average training example for that class.
- Nonzero coefficients for each class: yellow positive, blue negative.
- We notice that they are nonzero in different places, and create discriminant scores for each class (overall 81% of the variables are used).
- Some of these can be identified as appropriate contrast functionals for highlighting each digit. Not all of these are interpretable

Support vector machine

Goal: predict a two class response $Y \in \{-1, +1\}$ using a linear (simplest case) classification boundary of the form $f(x) = \beta_0 + \beta^\top x$, with the predicted class given by $\text{sign}(f(x))$



- The decision boundary is the solid line in the middle of the inner area.
- The margin is the half-width of the inner slab.
- Ideally, all of the triangle data points should lie above the slab on the right, and the circle points should lie below it on the left.

- In the picture, two circle points and two triangle points lie on the wrong side of their margin. These correspond to the "errors" ξ_i.
- The SVM decision boundary is chosen to maximize the margin, subject to a fixed budget on the total error $\sum_{i=1}^{n} \xi_i$.
- Idea: decision boundary achieving the largest margin has more space between the classes and will generalize better to test data.

Support vector machine

Optimization problem

$$\max_{\beta_0,\beta,\{\xi_i\}_1^n} M \quad \text{subject to } y_i \underbrace{(\beta_0 + \beta^\top x_i)}_{f(x_i;\beta_0,\beta)} \geq M(1-\xi_i) \quad \forall i$$

$$\text{and} \quad \xi_i \geq 0 \ \forall i, \quad \sum_{i=1}^n \xi_i \leq C_i, \ \text{and } \|\beta\|_2 = 1$$

- Linear cost function subject to convex constraints
- Many efficient algorithms have been designed for its solution
- It can be shown to be equivalent to the form

$$\min_{\beta_0,\beta} \left\{ \frac{1}{n} \sum_{i=1}^n [1 - y_i f(x_i;\beta_0,\beta)]_+ + \lambda \|\beta\|_2^2 \right\}$$

  where, for linear classification boundary, $f(x) = \beta_0 + \beta^\top x$
  (here decreasing λ has a similar effect to decreasing C; $\lambda = \frac{1}{C}$)
- The correctness of a given decision can be determined by checking whether or not the 'margin' $yf(x;\cdot,\cdot)$ is positive.
- Generalizations using *kernel* functions to create nonlinear boundaries; replace the squared L_2-norm of x by the squared Hilbert norm defined by a symmetric bivariate kernel $K(x,x') = \langle h(x), h(x') \rangle$, $f(x) = \beta_0 + \beta^\top h(x)$

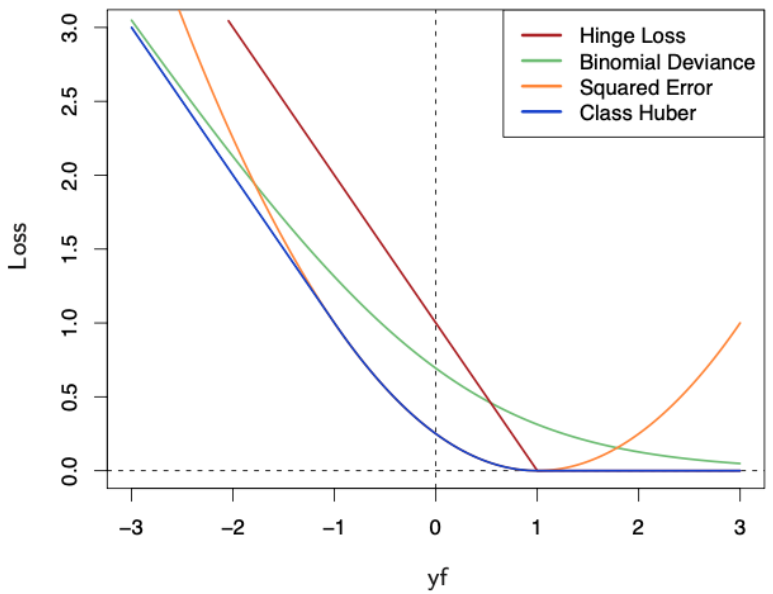
Support vector machine

- The traditional soft-margin linear SVM is fit by solving the optimization problem:

$$\min_{\beta_0,\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} [1 - y_i f(x_i)]_+ + \lambda \|\beta\|_2^2 \right\}$$

$$\min_{\beta_0,\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} \underbrace{[1 - y_i f(x_i)]_+}_{\phi(y_i f(x_i))} + \lambda \|\beta\|_2^2 \right\}$$

- $\phi(\cdot)$ is called *hinge loss* and penalize the negative margins that represent incorrect classifications.
- Written the optimization function in this form, the term $\lambda \|\beta\|_2^2$ is a penalization for the *hinge loss*.



'Sparsity' in support vector machine

- SVM are not sparse in the features.
- However, the *hinge loss function* $\phi(y_i f(x_i)) = [1 - y_i f(x_i; \beta_0, \beta)]_+$ is piecewise linear $\Rightarrow$ introduces a different kind of sparsity (in the observations).
- The solution $\hat{\beta}$ has the form

$$\hat{\beta} = \sum_{i=1}^{n} \hat{\alpha}_i y_i x_i$$

  each observation $i \in \{1, \ldots, n\}$ is associated with a nonnegative weight $\hat{\alpha}_i$, and only a subset of observations, referred to as the support set, will be associated with nonzero weights.
- Popular in high-dimensional classification problems with $p \gg n$: computations are based only on the support vector and are $O(pn^2)$ for both linear and nonlinear kernels.

Sparsity in support vector machine

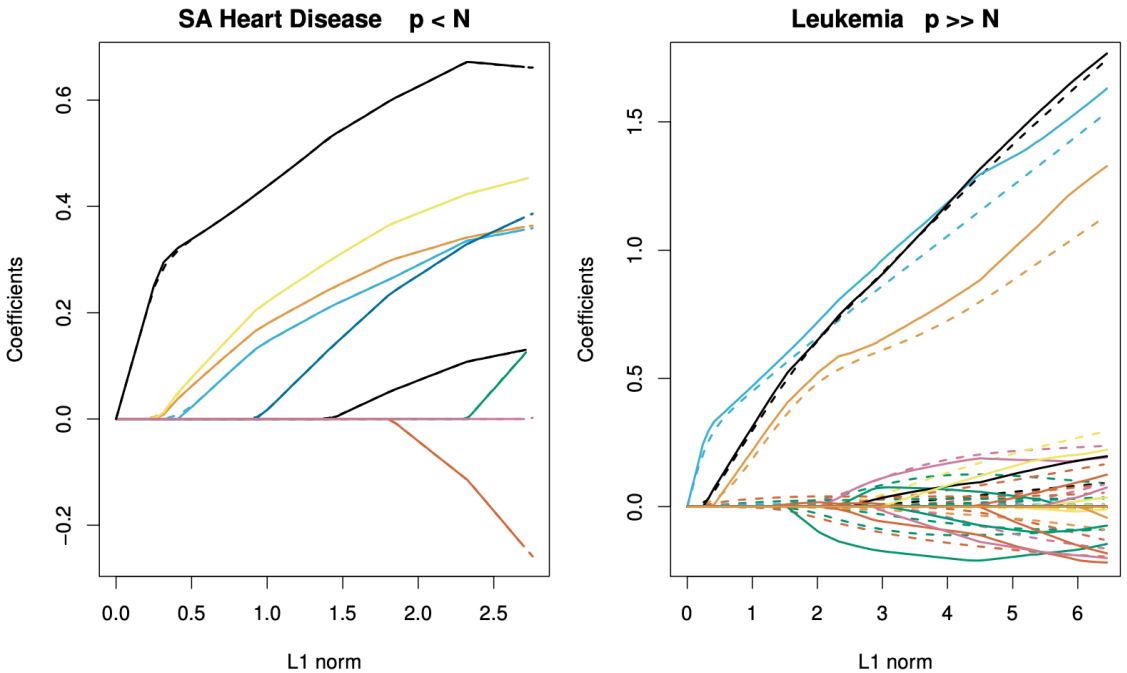
How to include sparsity in the features?
Idea: replace the L_2 penalty in the objective function with an L_1 penalty.
This yields the L_1-regularized linear SVM:

$$\min_{\beta_0,\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} [1 - y_i f(x_i; \beta_0, \beta)]_+ + \lambda \|\beta\|_1 \right\}$$

- This optimization problem is a linear program with many constraints. Efficient algorithms are complex.
- The solution paths can have many jumps, and show many discontinuities.
- Solution: replace the hinge loss $\phi(t) = [1 - t]_+$ with a squared hinge loss $\phi_{sh} = [1 - t]_+^2$, which is differentiable everywhere.

SVM lasso and logistic lasso

Results of regularized SVM are very similar to regularized logistic regression (simpler to be estimated and with results more stable)



Comparison of coefficient paths for L_1-regularized SVM vs logistic regression

Dashed lines: SVM coeff
Solid lines: logistic coeff.

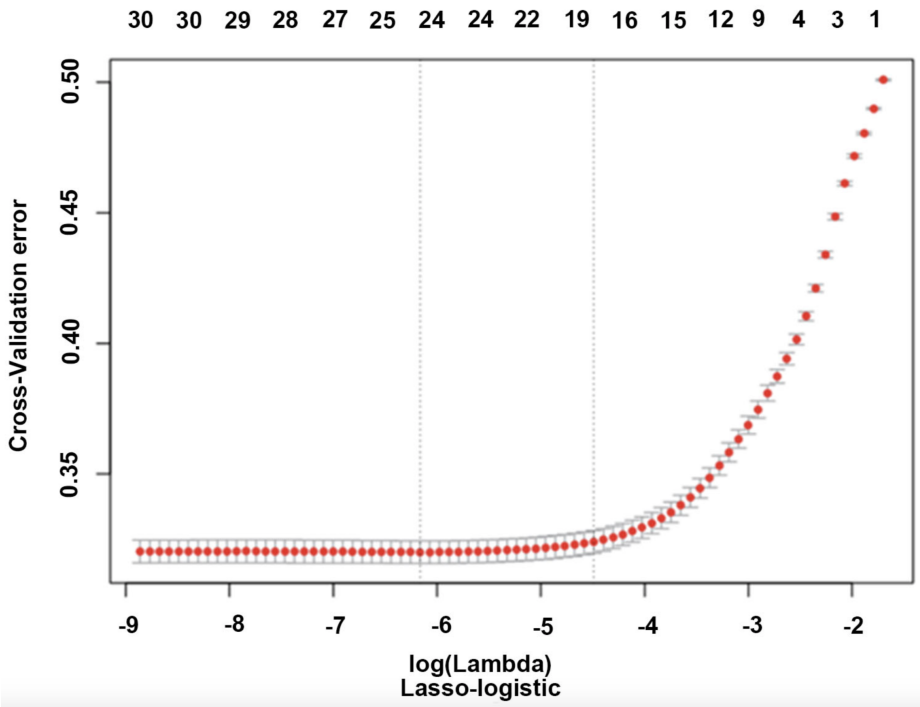
- South African heart disease data ($n = 462$ and $p = 9$): coronary heart disease (response) and risk factors (systolic blood pressure, tobacco, cholesterol, adiposity, family history of heart disease, type-A behavior, obesity, alcohol, age)
- Leukemia data ($n = 38$ and $p = 6087$): gene expression (predictors) of disease subtype (response: *acute lymphobastic leukemia* or *acute myeloid leukemia*).
- Similarity: striking in the left example, strong in the right.

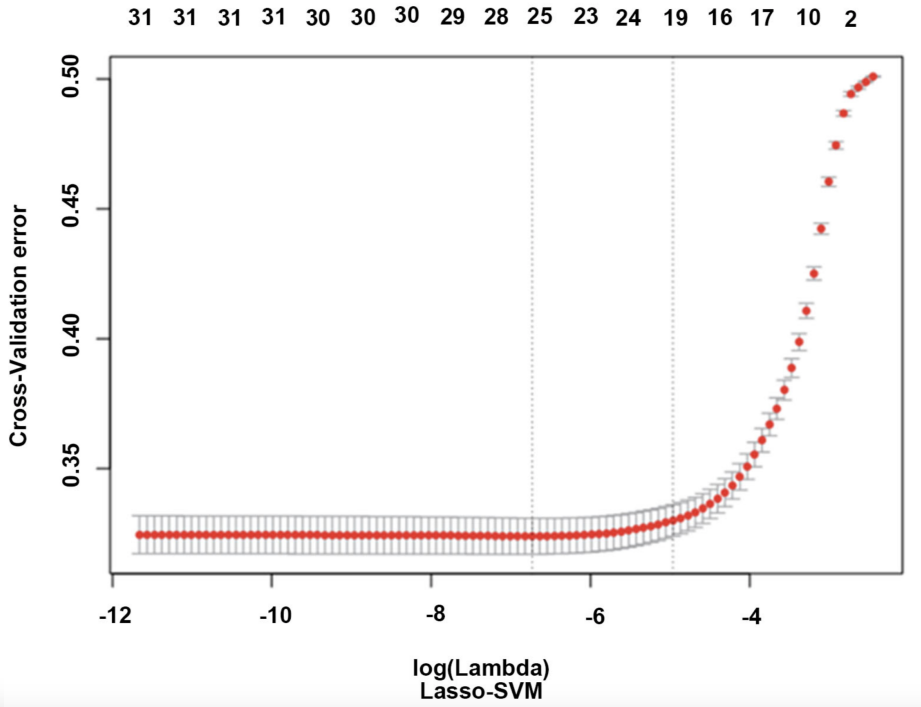
Credit scoring

- $n = 3521$ records, $p = 24$ predictors.
- Response:
  '0' performance customer;
  '1' default customer.
- training set: 3002 obs
  (1500 compliance, 1502 default),
  test set: 519 obs
  (258 compliance data, 261 default)
- Predictors. *Basic personal identity information*: domicile, gender, local work, education level and marital status; *Personal economic ability*: whether there is a CPF salary level; *Personal debt and debt repayment record*: frequency of personal housing loan, personal commercial housing loan pen number and frequency of other loan credit card account number, number, frequency of delinquent loans, loans overdue month loan highest monthly overdue amount, maximum length, loan account number of the contract amount, loan balance has been used lines, the average individual loan maximum contract value, the average individual loans minimum contract amount, the last six months on average use; *Total number of times of individual approval query and loan number*
- Fit the logistic lasso and SVM - lasso,
- Selection of λ: 10-fold cross-validation

ID	Variable Name	The Values
y	default	$y=0$ Not default; $y=1$ Default
x_1	registered area	$x_{1,1}$ = Northeast; $x_{1,2}$ = North China Plain; $x_{1,3}$ = Central China; $x_{1,4}$ = Eastern China
x_2	gender	$x_{2,1}$ = Male; $x_{2,2}$ = Female
x_3	Whether work is local	$x_{3,1}$ = Local; $x_{3,2}$ = Not local
x_4	edu level	$x_{4,1}$ = Junior high school/Senior high school/others; $x_{4,2}$ = Junior college/Junior college and below; $x_{4,3}$ = Undergraduate; $x_{4,4}$ = Master/Doctor
x_5	marital status	$x_{5,1}$ = Maid; $x_{5,2}$ = Married; $x_{5,3}$ = Others (divorced/widowed)
x_6	Whether accumulation fund	$x_{6,1}$ = Not; $x_{6,2}$ = Yes
x_7	pay grades	$x_{7,1}$ = 0 - 3500; $x_{7,2}$ = 3501 - 8000; $x_{7,3}$ = 8000 above
x_8	The number of individual housing loans	$x_8 \in N$
x_9	The number of individual commercial housing loans	$x_9 \in N$
x_{10}	Other loans	$x_{10} \in N$
x_{11}	Debit card account number	$x_{11} \in N$
x_{12}	The number of overdue loans	$x_{12} \in N$
x_{13}	Months of overdue loans	$x_{13} \in N$
x_{14}	The maximum amount of overdue loans per month	$x_{14} \in [0,+\infty]$
x_{15}	Maximum length of loan (year)	$x_{15} \in N$
x_{16}	Total number of approval inquiries	$x_{16} \in N$
x_{17}	Loan number	$x_{17} \in N$
x_{18}	Loan account number	$x_{18} \in N$
x_{19}	contract amount	$x_{19} \in [0,+\infty]$
x_{20}	loan balance	$x_{20} \in [0,+\infty]$
x_{21}	Have used limit	$x_{21} \in [0,+\infty]$
x_{22}	Average maximum contract amount for a single lender	$x_{22} \in [0,+\infty]$
x_{23}	Average minimum contract amount for a single lender	$x_{23} \in [0,+\infty]$
x_{24}	Average usage in the last 6 months	$x_{24} \in [0,+\infty]$

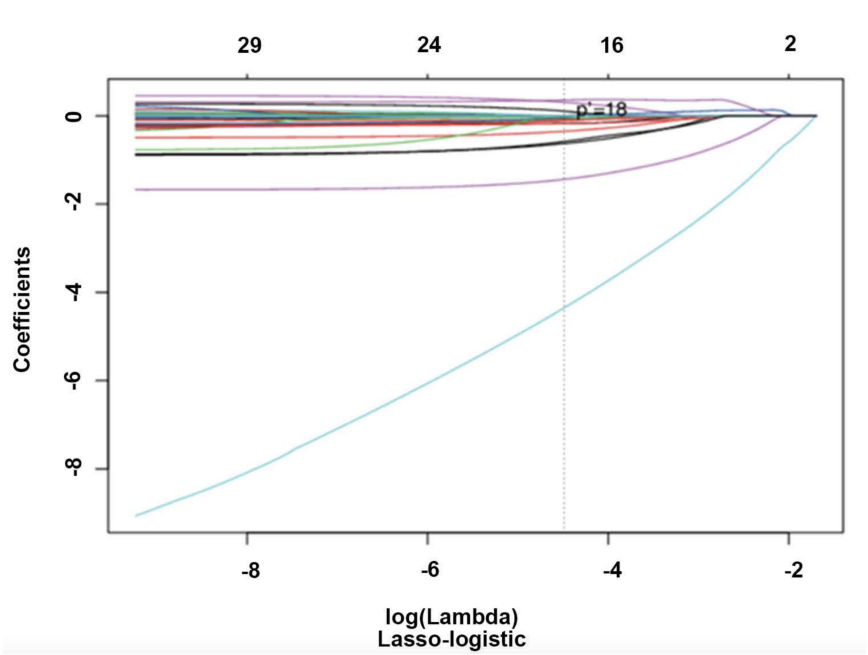
Credit scoring - cross validation

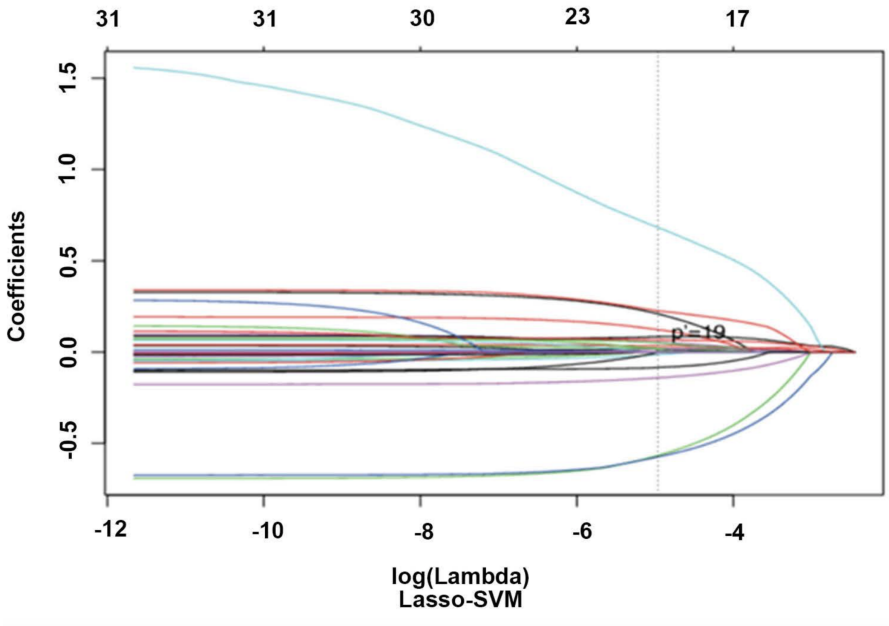




- Lasso-logistic, $\lambda = 0.01122485$
- Lasso-SVM, $\lambda = 0.00699683$

Credit scoring - coefficient estimates





Variate	Full variables	Forward	Backwards	Lasso-logistic	Lasso-SVM
X_{1_1}	1.715	0	1.721	0	−0.555
X_{1_2}	1.674	0	1.695	0.010	−0.567
X_{1_3}	−4.562	−4.247	−4.561	−4.354	0.667
X_{1_4}	−1.534	−1.681	0	−1.440	0.016
X_2	−0.868	−0.854	−0.858	−0.630	0.209
X_3	−0.497	−0.504	−0.502	−0.352	0.122
X_{4_1}	1.408	0.341	0.501	0.105	−0.003
X_{4_2}	0.780	0.301	0.218	0	0
X_{4_3}	0.575	0	0	−0.111	0.015
X_{4_4}	0.761	0	0	0	0
X_{5_1}	−0.025	−0.002	0	0	0
X_{5_2}	−0.179	0	−0.201	−0.037	0
X_{5_3}	−0.034	0	0	0	0
X_6	0.072	0	0	0	0
X_{7_1}	0.306	0.393	0.311	0.355	−0.085
X_{7_2}	−0.893	0.205	−1.111	−0.582	0.226
X_{7_3}	0.934	0	0	0	0
X_8	0.123	0.101	0.098	0	0
X_9	0.058	0	0	0	0
X_{10}	−0.265	−0.213	−0.212	−0.104	0.053
X_{11}	−0.078	0	0	−0.003	0.013
X_{12}	−0.100	−0.111	−0.109	−0.084	0.033
X_{13}	−0.190	−0.206	−0.197	−0.064	0.050
X_{14}	−0.033	0	0	0	0
X_{15}	0.130	0.121	0.119	0	−0.016
X_{16}	0.455	0.459	0.457	0.306	−0.142
X_{17}	0.222	−0.218	−0.209	−0.173	0.084
X_{18}	0.164	0	0	0	0
X_{19}	−0.405	0	0	−0.021	0
X_{20}	0.276	0	0	0	0
X_{21}	0.136	0	0	0	0
X_{22}	0.065	0	0	0	0
X_{23}	0.041	0	0	0	0
X_{24}	−0.325	−0.249	−0.263	−0.178	0.067
Intercept term	1.026	1.275	1.836	2.575	−0.382

Credit scoring - correct classifications

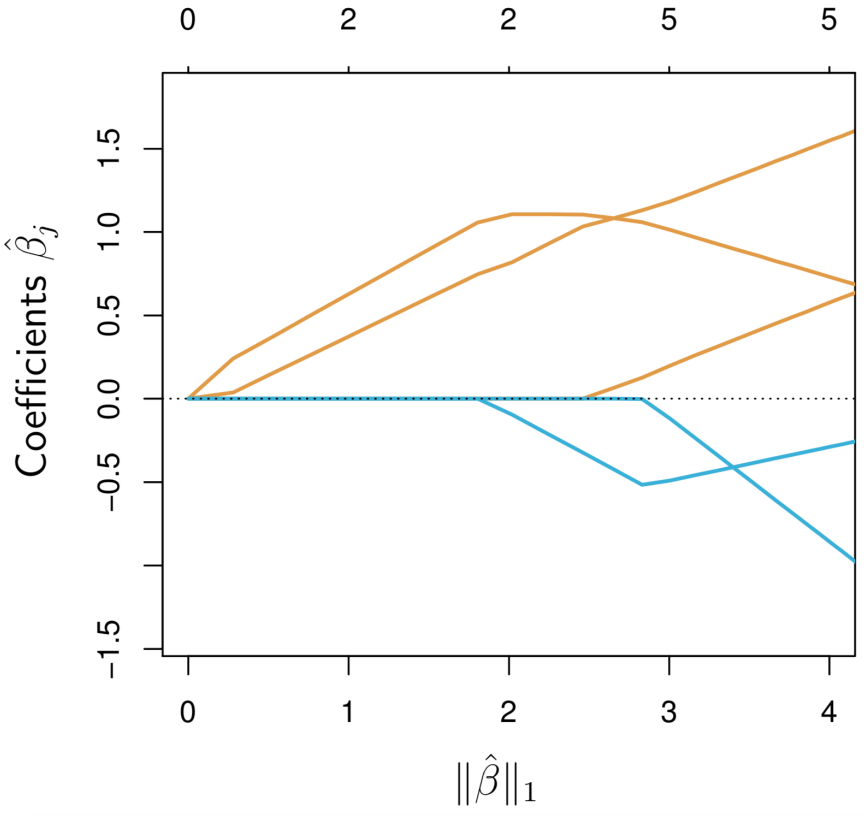
Model	Training set			Test set		
	Good	Bad	Total	Good	Bad	Total
Full variables	71.3	75.6	73.4	67.0	71.7	69.3
Forward selection	70.7	75.6	73.1	65.5	72.1	68.9
Backward selection	72.3	73.8	73.1	69.3	67.1	68.2
Lasso-logistic	74.5	80.0	77.2	74.7	79.5	77.1
Lasso-SVM	73.7	80.2	76.9	73.6	80.2	76.8

- In credit risk assessment, misclassification of default users into non-defaulting users is more of a potential loss to banks or society.
- The model is more important for to correctly classify the default users than to take non-defaulting users into consideration.
- Test set
  SVM-lasso model number of default users will be up to 80.2%
  $\Rightarrow$ +8.5% wrt full-variable model; +8.1% wrt stepwise forward; +13.1% wrt stepwise backward; +0.70% wrt logistic-lasso
- classification of non-defaulting users. Logistic-lasso is the best rate in both the training set and the test set.

Generalizations of the lasso

Highly correlated variables

- The lasso does not handle highly correlated variables very well;
- the coefficient paths tend to be erratic and can sometimes show wild behavior
- e.g., simulate $n = 100$ units from six variables highly correlated in groups of three
- Pairwise correlations around 0.97 in each group.



- little identifiability between parameters
- erratic behavior as the regularization parameter λ is varied.
- lasso coefficients do not reflect the relative importance of individual variables.

The elastic net

Compromise between the ridge and the lasso penalties

Elastic net is the solution (β_0, β) to the optimization problem

$$\min_{\beta_0,\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \right\}$$

- The squared norm tends to average the coefficients of predictors that are correlated, while the L_1 norm chooses among the averaged groups.
- When $p > n$, the number of non-zero coefficients can exceed n – unlike the lasso.
- Alternative form of penalty

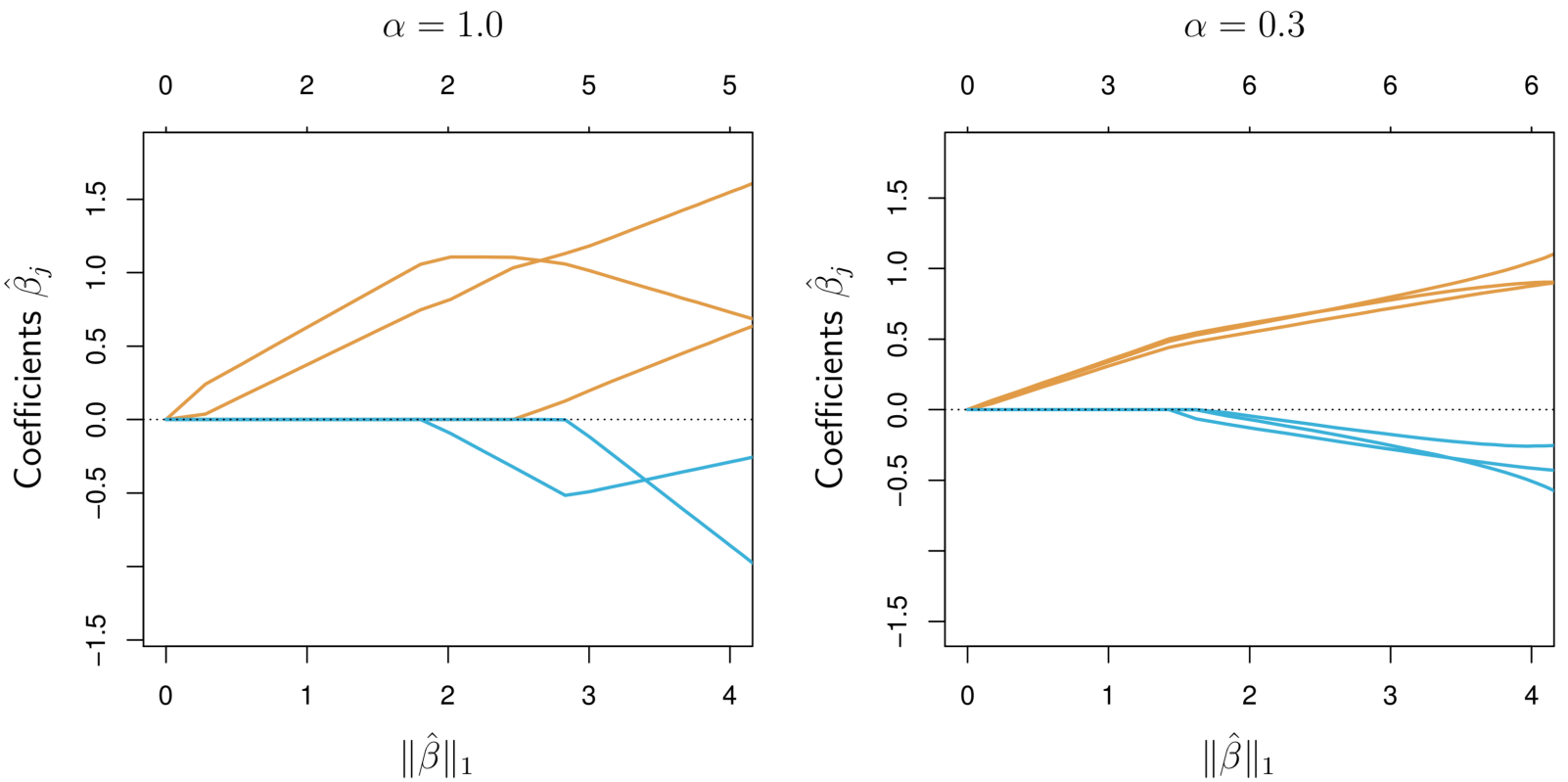
$$\lambda \sum_{j=1}^{p} \left[\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j| \right]$$

The elastic net

$$\min_{\beta_0,\beta}\left\{\frac{1}{2n}\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}x_{ij}\beta_j\right)^2+\lambda\sum_{j=1}^{p}\left[\frac{1}{2}(1-\alpha)\beta_j^2+\alpha|\beta_j|\right]\right\}$$

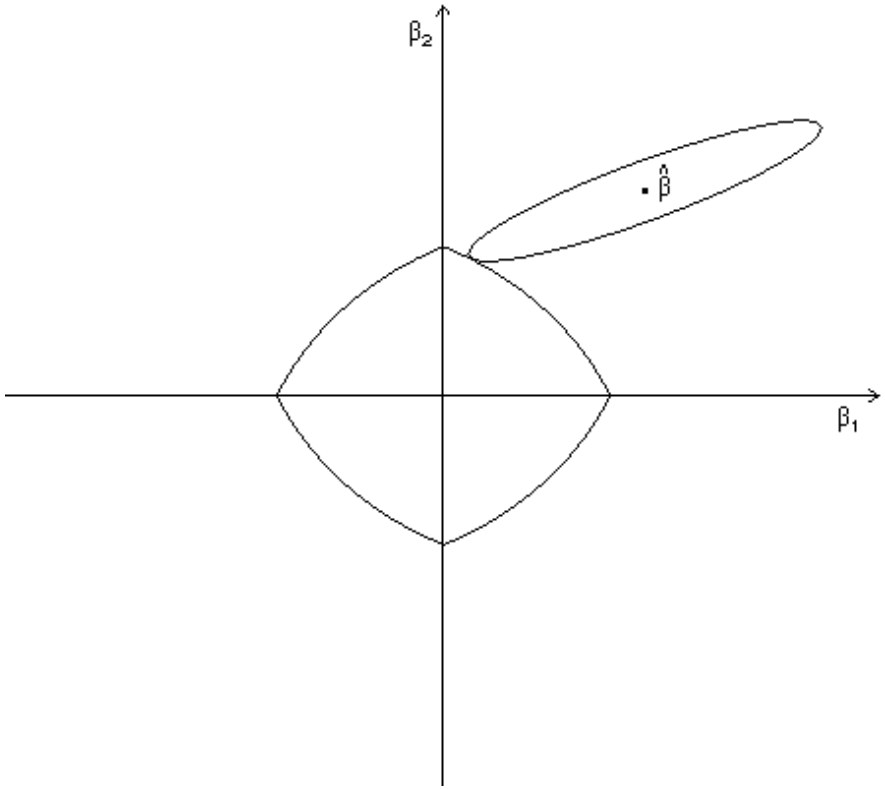
- When $\alpha = 1$, it reduces to the L_1-norm or lasso penalty, and with $\alpha = 0$, it reduces to the squared L_2-norm, corresponding to the ridge penalty.
- Elastic net automatically controls for strong within-group correlations.
- For any $\alpha < 1$ and $\lambda > 0$, the elastic-net problem is strictly convex: a unique solution exists irrespective of the correlations or duplications in the s_j.

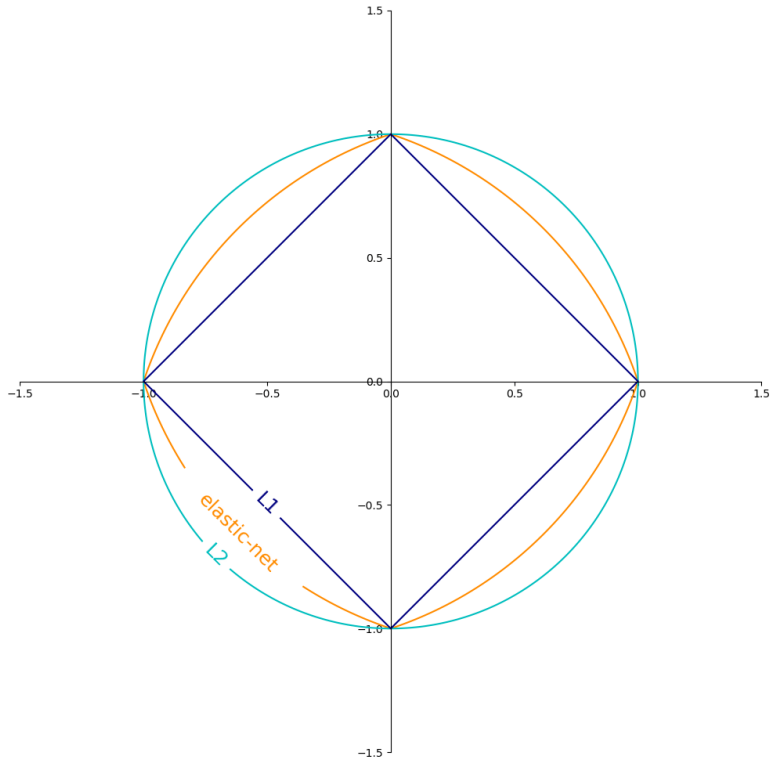
Elastic net



- In contrast to the lasso paths, the coefficients are selected approximately together in their groups, and also approximately share their values equally.

Elastic net





- $p = 2$, $\alpha = 1$ and $t = 0.5$.
- There are singularities at the vertexes and the edges are strictly convex.
- The sharp corners and edges encourage selection
- The curved contours encourage sharing coefficients (of strongly correlated variables)
- The strength of convexity varies with α.

Elastic net

- An additional tuning parameter α to be determined.
- It can be viewed as a higher-level parameter, and set on subjective grounds.
- Or, selected by cross-validation
- Optimization algorithm: elastic-net problem is convex in the pair (β_0, β).
- Coordinate descent is particularly effective, with updates simple extension of those for the lasso

Ames (Iowa) housing data

- Property sales that had occurred in Ames (Iowa) between 2006 and 2010.
- The data set contains 2930 observations and a large number of explanatory variables (23 nominal, 23 ordinal, 14 discrete, and 20 continuous) involved in assessing home values.
- Training set: 2053 units
  Test set: 877 units
- Response is `log(Sale_Price)`

Ames (Iowa) housing data

- Explanatory variables:

Order: **Observation number**
PID: **Parcel identification number - can be used with city web site for parcel review.**
MS SubClass: **Identifies the type of dwelling involved in the sale.**
MS Zoning: **Identifies the general zoning classification of the sale.**
Lot Frontage: **Linear feet of street connected to property**
Lot Area: **Lot size in square feet**
Street: **Type of road access to property**
Alley: **Type of alley access to property**
Lot Shape: **General shape of property**
Land Contour: **Flatness of the property**
Utilities: **Type of utilities available**
Lot Config: **Lot configuration**
Land Slope: **Slope of property**
Neighborhood: **Physical locations within Ames city limits (map available)**
Condition 1: **Proximity to various conditions**
Condition 2: **Proximity to various conditions (if more than one is present)**
Bldg Type: **Type of dwelling**
House Style: **Style of dwelling**
Overall Qual: **Rates the overall material and finish of the house**
Overall Cond: **Rates the overall condition of the house**
Year Built: **Original construction date**
Year Remod/Add: **Remodel date (same as construction date if no remodeling or additions)**
Roof Style: **Type of roof**
Roof Matl: **Roof material**
Exterior 1: **Exterior covering on house**
Exterior 2: **Exterior covering on house (if more than one material)**
Mas Vnr Type: **Masonry veneer type**
Mas Vnr Area: **Masonry veneer area in square feet**
Exter Qual: **Evaluates the quality of the material on the exterior**
Exter Cond: **Evaluates the present condition of the material on the exterior**
Foundation: **Type of foundation**
Bsmt Qual: **Evaluates the height of the basement**
Bsmt Cond: **Evaluates the general condition of the basement**
Bsmt Exposure: **Refers to walkout or garden level walls**
BsmtFin Type 1: **Rating of basement finished area**
BsmtFin SF 1: **Type 1 finished square feet**
BsmtFinType 2: **Rating of basement finished area (if multiple types)**
BsmtFin SF 2: **Type 2 finished square feet**
Bsmt Unf SF: **Unfinished square feet of basement area**
Total Bsmt SF: **Total square feet of basement area**
Heating: **Type of heating**
HeatingQC: **Heating quality and condition**
Central Air: **Central air conditioning**
Electrical: **Electrical system**
1st Flr SF: **First Floor square feet**
2nd Flr SF: **Second floor square feet**
Low Qual Fin SF: **Low quality finished square feet (all floors)**
Gr Liv Area: **Above grade (ground) living area square feet**
Bsmt Full Bath: **Basement full bathrooms**
Bsmt Half Bath: **Basement half bathrooms**
Full Bath: **Full bathrooms above grade**
Half Bath: **Half baths above grade**
Bedroom: **Bedrooms above grade (does NOT include basement bedrooms)**
Kitchen: **Kitchens above grade**
KitchenQual: **Kitchen quality**
TotRmsAbvGrd: **Total rooms above grade (does not include bathrooms)**
Functional: **Home functionality (Assume typical unless deductions are warranted)**
Fireplaces: **Number of fireplaces**
FireplaceQu: **Fireplace quality**
Garage Type: **Garage location**
Garage Yr Blt: **Year garage was built**
Garage Finish: **Interior finish of the garage**
Garage Cars: **Size of garage in car capacity**
Garage Area: **Size of garage in square feet**
Garage Qual: **Garage quality**
Garage Cond: **Garage condition**
Paved Drive: **Paved driveway**
Wood Deck SF: **Wood deck area in square feet**
Open Porch SF: **Open porch area in square feet**
Enclosed Porch: **Enclosed porch area in square feet**
3-Ssn Porch: **Three season porch area in square feet**
Screen Porch: **Screen porch area in square feet**
Pool Area: **Pool area in square feet**
Pool QC: **Pool quality**
Fence: **Fence quality**
Misc Feature: **Miscellaneous feature not covered in other categories**
Misc Val: **$Value of miscellaneous feature**
Mo Sold: **Month Sold**
Yr Sold: **Year Sold**
Sale Type: **Type of sale**
Sale Condition: **Condition of sale**

Ames (Iowa) housing data

- Explanatory variables:

Order: **Observation number**
PID: **Parcel identification number - can be used with city web site for parcel review.**
MS SubClass: **Identifies the type of dwelling involved in the sale.**
MS Zoning: **Identifies the general zoning classification of the sale.**
Lot Frontage: **Linear feet of street connected to property**
Lot Area: **Lot size in square feet**
Street: **Type of road access to property**
Alley: **Type of alley access to property**
Lot Shape: **General shape of property**
Land Contour: **Flatness of the property**
Utilities: **Type of utilities available**
Lot Config: **Lot configuration**
Land Slope: **Slope of property**
Neighborhood: **Physical locations within Ames city limits (map available)**
Condition 1: **Proximity to various conditions**
Condition 2: **Proximity to various conditions (if more than one is present)**
Bldg Type: **Type of dwelling**
House Style: **Style of dwelling**
Overall Qual: **Rates the overall material and finish of the house**
Overall Cond: **Rates the overall condition of the house**
Year Built: **Original construction date**
Year Remod/Add: **Remodel date (same as construction date if no remodeling or additions)**
Roof Style: **Type of roof**
Roof Matl: **Roof material**
Exterior 1: **Exterior covering on house**
Exterior 2: **Exterior covering on house (if more than one material)**
Mas Vnr Type: **Masonry veneer type**
Mas Vnr Area: **Masonry veneer area in square feet**
Exter Qual: **Evaluates the quality of the material on the exterior**
Exter Cond: **Evaluates the present condition of the material on the exterior**
Foundation: **Type of foundation**
Bsmt Qual: **Evaluates the height of the basement**
Bsmt Cond: **Evaluates the general condition of the basement**
Bsmt Exposure: **Refers to walkout or garden level walls**
BsmtFin Type 1: **Rating of basement finished area**
BsmtFin SF 1: **Type 1 finished square feet**
BsmtFinType 2: **Rating of basement finished area (if multiple types)**
BsmtFin SF 2: **Type 2 finished square feet**
Bsmt Unf SF: **Unfinished square feet of basement area**
Total Bsmt SF: **Total square feet of basement area**
Heating: **Type of heating**
HeatingQC: **Heating quality and condition**
Central Air: **Central air conditioning**
Electrical: **Electrical system**
1st Flr SF: **First Floor square feet**
2nd Flr SF: **Second floor square feet**
Low Qual Fin SF: **Low quality finished square feet (all floors)**
Gr Liv Area: **Above grade (ground) living area square feet**
Bsmt Full Bath: **Basement full bathrooms**
Bsmt Half Bath: **Basement half bathrooms**
Full Bath: **Full bathrooms above grade**
Half Bath: **Half baths above grade**
Bedroom: **Bedrooms above grade (does NOT include basement bedrooms)**
Kitchen: **Kitchens above grade**
KitchenQual: **Kitchen quality**
TotRmsAbvGrd: **Total rooms above grade (does not include bathrooms)**
Functional: **Home functionality (Assume typical unless deductions are warranted)**
Fireplaces: **Number of fireplaces**
FireplaceQu: **Fireplace quality**
Garage Type: **Garage location**
Garage Yr Blt: **Year garage was built**
Garage Finish: **Interior finish of the garage**
Garage Cars: **Size of garage in car capacity**
Garage Area: **Size of garage in square feet**
Garage Qual: **Garage quality**
Garage Cond: **Garage condition**
Paved Drive: **Paved driveway**
Wood Deck SF: **Wood deck area in square feet**
Open Porch SF: **Open porch area in square feet**
Enclosed Porch: **Enclosed porch area in square feet**
3-Ssn Porch: **Three season porch area in square feet**
Screen Porch: **Screen porch area in square feet**
Pool Area: **Pool area in square feet**
Pool QC: **Pool quality**
Fence: **Fence quality**
Misc Feature: **Miscellaneous feature not covered in other categories**
Misc Val: **$Value of miscellaneous feature**
Mo Sold: **Month Sold**
Yr Sold: **Year Sold**
Sale Type: **Type of sale**
Sale Condition: **Condition of sale**

Ames (Iowa) housing data - multicollinearity

- Multicollinearity is present in the data
- e.g., `Gr_Liv_Area` and `TotRmsAbvGrd` have a correlation of 0.801
- Both variables strongly correlated to response variable (`log(Sale_Price)`)
- Fit a model with both these variables

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	11.2876	0.0272	414.83	0.0000
Gr_Liv_Area	0.0007	0.0000	32.55	0.0000
TotRms_AbvGrd	-0.0503	0.0069	-7.25	0.0000

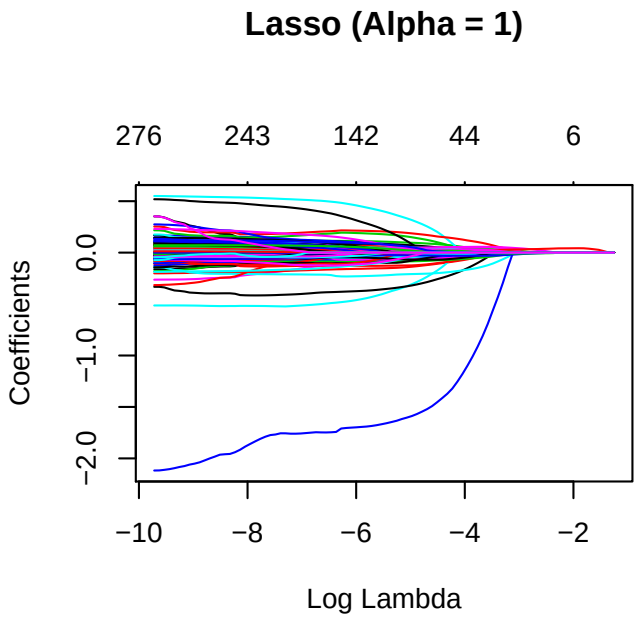
- Fit one model for each variable independently:

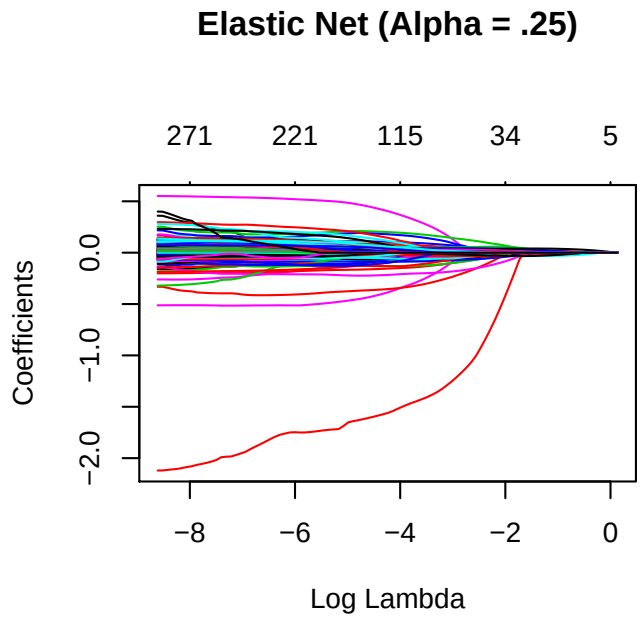
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	11.1552	0.0204	546.13	0.0000
Gr_Liv_Area	0.0006	0.0000	44.68	0.0000

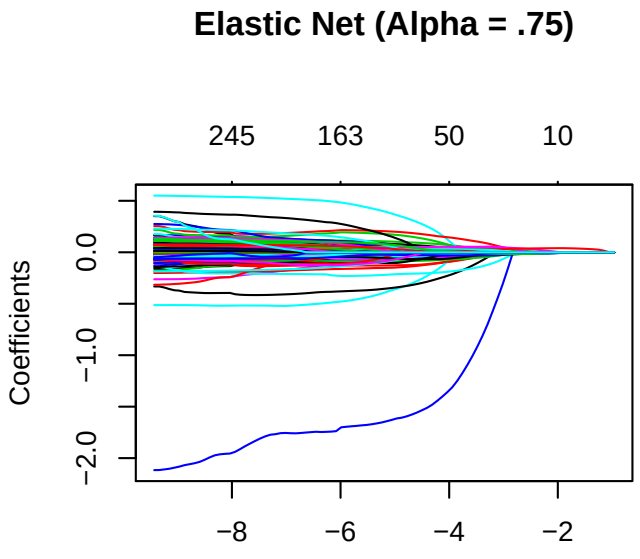
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	11.1758	0.0332	336.27	0.0000
TotRms_AbvGrd	0.1319	0.0050	26.19	0.0000

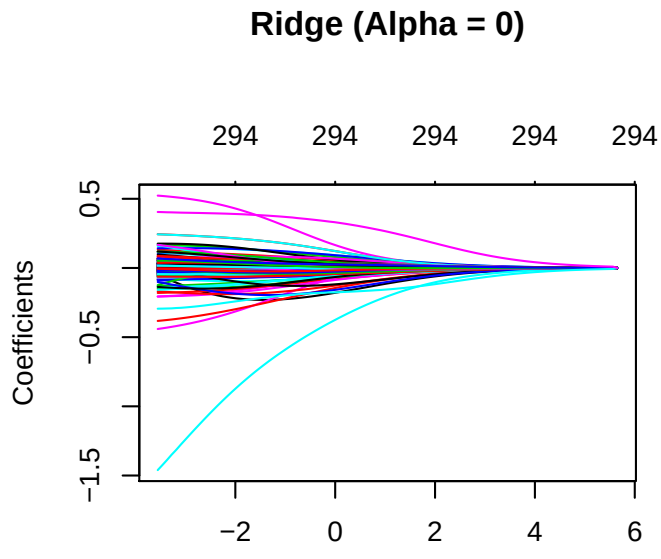
- They both show a positive impact.
- `Gr_Liv_Area` effect is smaller and the TotRmsAbvGrd is positive with a much larger magnitude.

Ames (Iowa) housing data





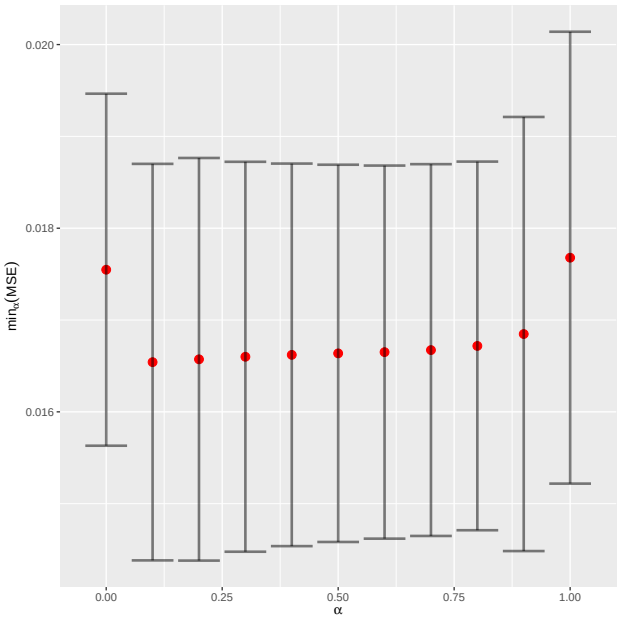




Ames (Iowa) housing data

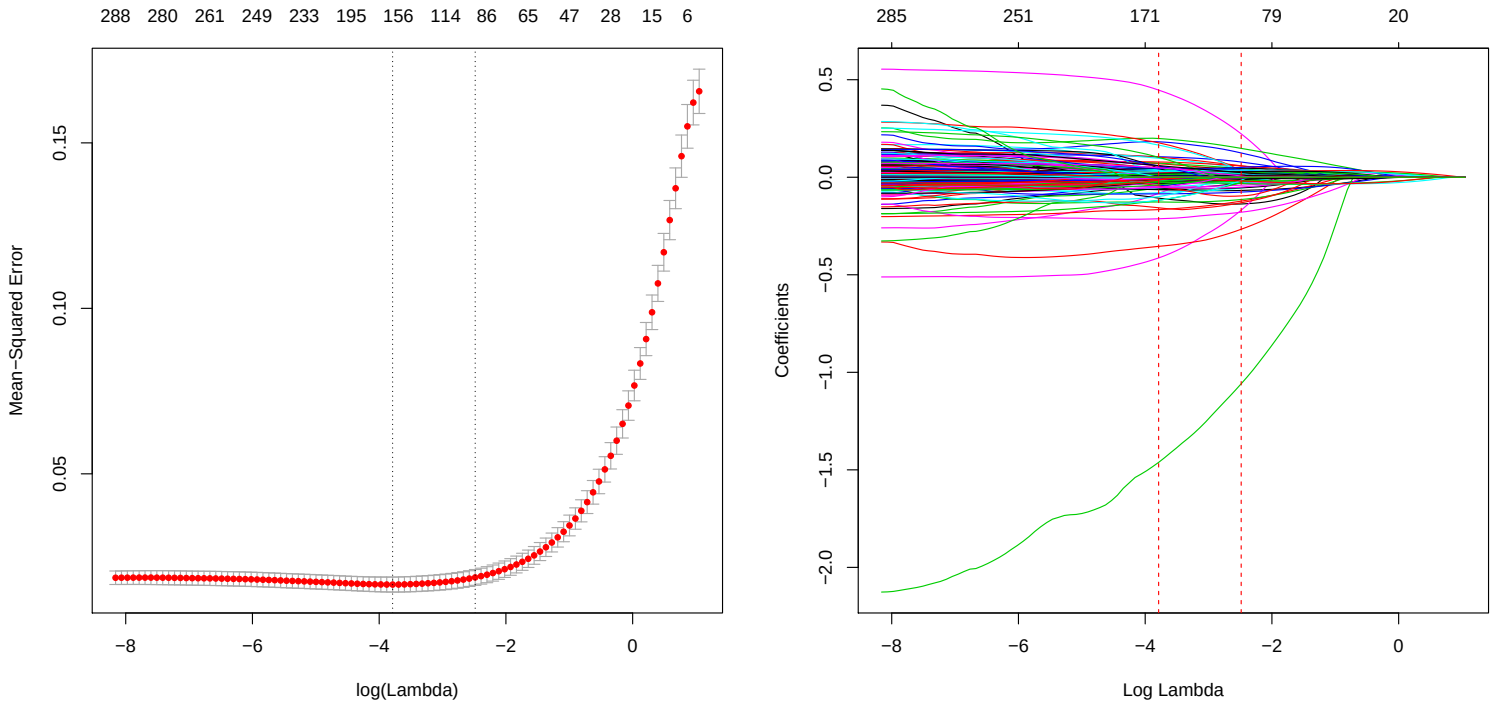
- We need to tune both λ and α
- Cross validation: for a grid of 10 values of α $(0.0, 0.1, 0.2, \ldots, 1.0)$ apply elastic net and extract the minimum and one standard error MSE values and their respective λ values.

	alpha	mse_min	mse_1se	lambda_min	lambda_1se
1	0.0000	0.0175	0.0195	0.1051	0.4244
2	0.1000	0.0165	0.0187	0.0226	0.0833
3	0.2000	0.0166	0.0188	0.0113	0.0457
4	0.3000	0.0166	0.0187	0.0075	0.0305
5	0.4000	0.0166	0.0187	0.0057	0.0229
6	0.5000	0.0166	0.0187	0.0045	0.0183
7	0.6000	0.0167	0.0187	0.0038	0.0152
8	0.7000	0.0167	0.0187	0.0032	0.0131
9	0.8000	0.0167	0.0187	0.0028	0.0114
10	0.9000	0.0168	0.0192	0.0025	0.0112
11	1.0000	0.0177	0.0201	0.0025	0.0110



- Select $\alpha = 0.1$ with $\lambda_{min} = 0.0226$, or $\lambda_{1se} = 0.0833$

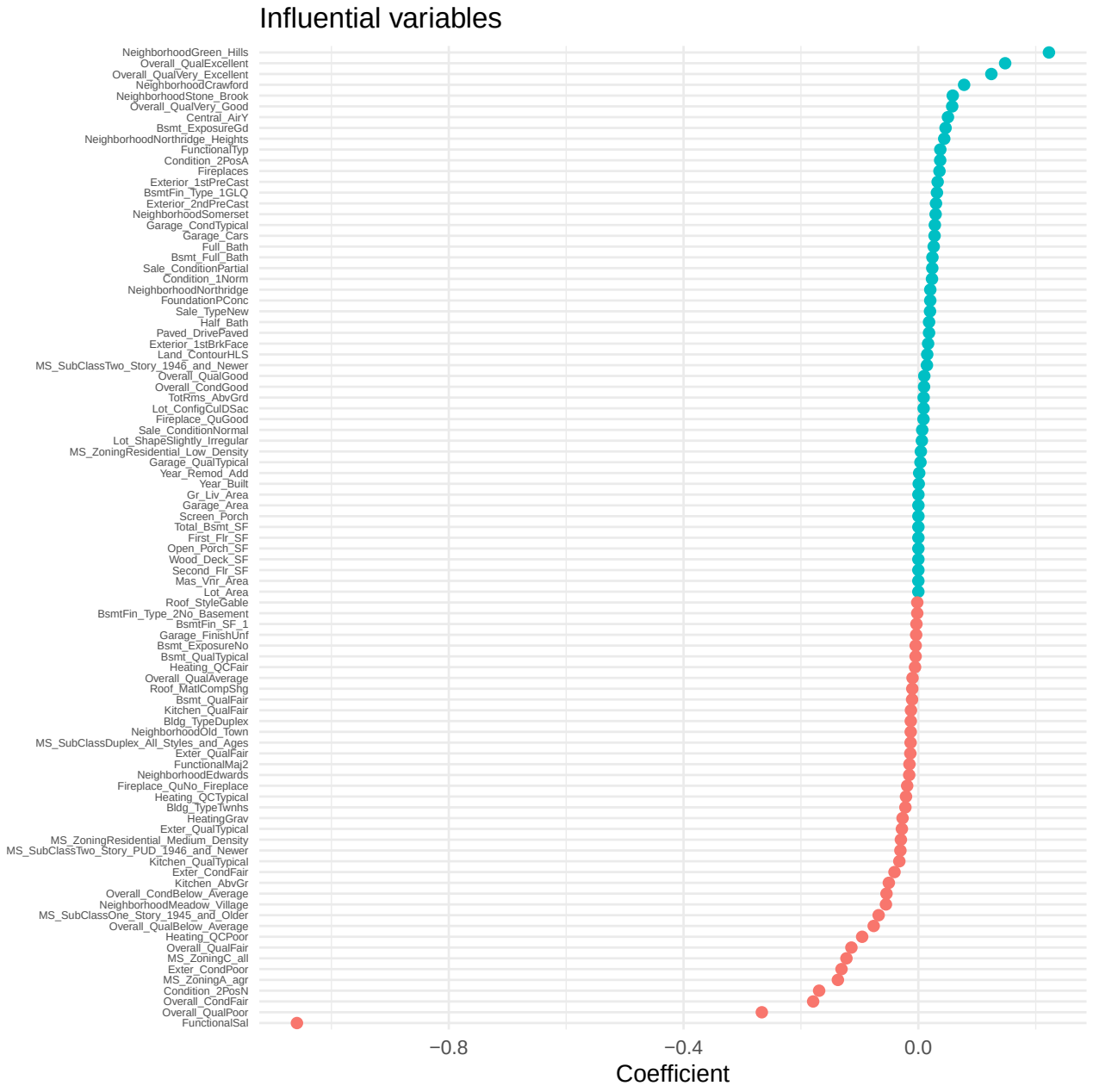
Ames (Iowa) housing data



The coefficients for the two correlated variables are now

- for $\lambda_{min} = 0.0226$:
  `Gr_Liv_Area`: 0.00014
  `TotRmsAbvGrd`: 0.00707
- for $\lambda_{1se} = 0.0833$:
  `Gr_Liv_Area`: 0.00013
  `TotRmsAbvGrd`: 0.00905

Ames (Iowa) housing data



Ames (Iowa) housing data

- Once selected the 'best' model we can do prediction on the test set
- and we can obtain measures of quality of prediction
  Mean squared error elastic-net on the test set $= 0.0250$
  Mean squared error lasso on the test set $= 0.0268$

Group structure in covariates

- In many regression problems, predictors are not distinct but arise from common underlying factors
- Sometimes covariates have a natural group structure
- $\Rightarrow$ all coefficients within a group become nonzero (or zero) simultaneously.
- Examples
  - categorical factors coded as a set of indicator variables and want to include or exclude this group of variables together.
  - Continuous features may be represented by a group of basis functions (e.g., splines)
  - Groups of measurements may be taken in the hopes of capturing unobservable latent variables

Potential advantages of grouping

- We look at these cases where features can be organized into related groups, and focus on methods for selecting important groups and estimating their effects
- One could, of course, still use methods like the lasso in these cases
- However, if there is indeed information contained in the grouping structure, methods that ignore it, will likely be inefficient
- Furthermore, by selecting important groups of variables, we should obtain models that are more sensible and interpretable

Notation

Let us extend our usual notation as follows:

- Consider a linear regression model
- Suppose we have J $(j = 1, ..., J)$ groups of covariates, with p_j denoting the size of group j, i.e., $\sum_j p_j = p$
- Suppose $Z_j \in \mathbb{R}^{p_j}$ is an $n \times p_j$ matrix representing the jth group of p_j variables, and γ_j the corresponding coefficient vector of the jth group.
- Our goal is to predict $y \in \mathbb{R}$ based on covariates $(Z_1, \ldots, Z_J)$.
- As usual, we are interested in estimating the vector of coefficients $\gamma = [\gamma_1, \ldots, \gamma_J]^\top$ using a loss function $L(\gamma|Z, y)$ (in the linear case this is the least squares loss function) which quantifies the discrepancy between the observations y and the linear predictors $\eta = Z\gamma = \sum_j Z_j\gamma_j$
- Covariates that do not belong to a group may be thought of as a group of one

Group lasso

The grouped lasso solves

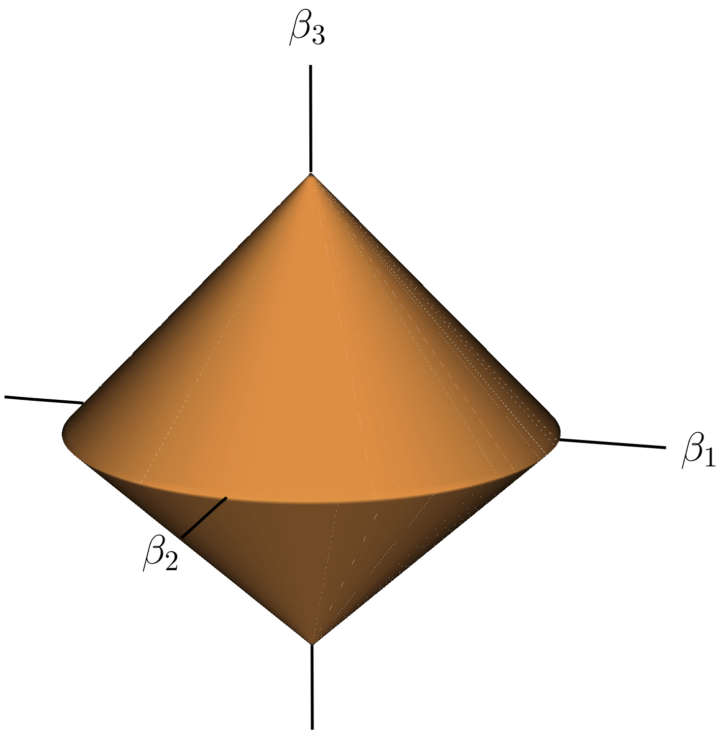
$$\min_{\beta_0 \in \mathbb{R}, \gamma \in \mathbb{R}^{p_j}} \left\{ \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{J} z_{ij}^{\top} \gamma_j \right)^2 + \lambda \sum_{j=1}^{J} \|\gamma_j\|_2 \right\}$$

where $\|\gamma_j\|_2$ is the Euclidean norm of γ_j

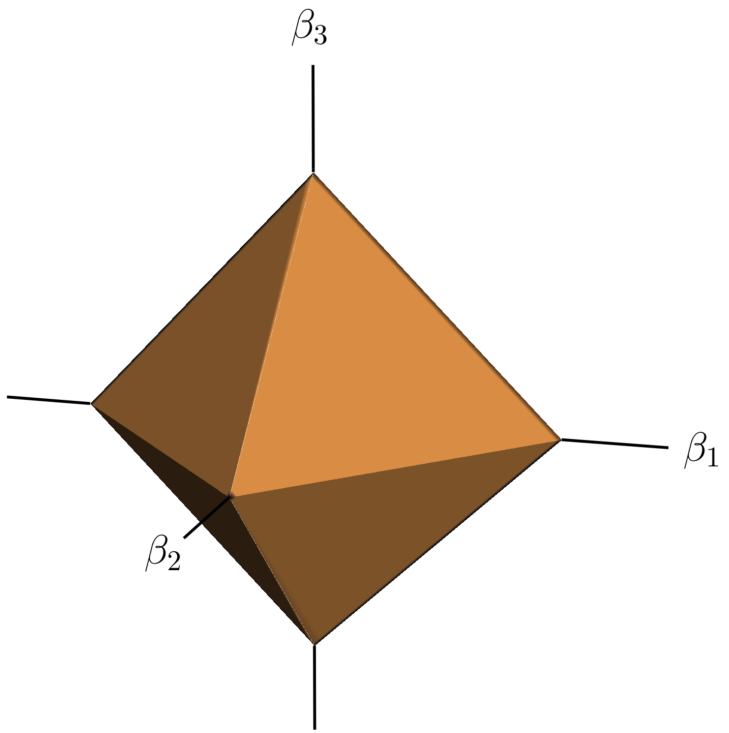
- This is a natural extension of the lasso to the grouped variable setting: instead of penalizing the magnitude ($|\gamma_j|$) of individual coefficients, we penalize the magnitude ($\|\gamma_j\|_2$) of groups of coefficients
- depending on $\lambda \geq 0$, either the entire vector $\hat{\gamma}_j = 0$, or all its elements will be nonzero
- when $p_j = 1$, then we have $\|\gamma_j\|_2 = |\gamma_j|$, so if all the groups are singletons, the optimization problem reduces to the ordinary lasso.

Group lasso

Group lasso ball in $\mathbb{R}^3$



L_1 ball in $\mathbb{R}^3$



Two groups with coefficients $\gamma_1 = (\beta_1, \beta_2) \in \mathbb{R}^2$ and $\gamma_2 = \beta_3 \in \mathbb{R}^1$.

Computation for the Group Lasso

- We can apply the coordinate descent idea in a groupwise fashion; this algorithm is known as *group descent*, *blockwise coordinate descent*, or the "shooting algorithm"
- We want to minimize

$$\min_{\gamma_1,\ldots,\gamma_J} \left\{ \frac{1}{2}\|y - \sum_{j=1}^{J} Z_j\gamma_j\|_2^2 + \lambda \left(\sum_{j=1}^{J} \|\gamma_j\|_2 \right) \right\}$$

- For simplicity we ignore the intercept β_0, we can center all the variables and the response and $\beta_0 = 0$

Computation for the Group Lasso

- The subgradient equations are

$$-Z_j^\top (y - \sum_{k=1}^{J} Z_k \hat{\gamma}_k) + \lambda \hat{s}_j = 0, \qquad \text{for } j = 1, \ldots, J$$

  where $\hat{s}_j \in \mathbb{R}^{p_j}$ is an element of the subdifferential of the norm $\|\cdot\|_2$ evaluated at $\hat{\gamma}_j$:

$$\hat{s}_j = \begin{cases} \hat{\gamma}_j / \|\hat{\gamma}_j\|_2 & \text{if } \hat{\gamma}_j \neq 0 \\ \text{is any vector } \|\hat{s}_j\|_2 < 1 & \text{if } \hat{\gamma}_j = 0 \end{cases}$$

- The penalty is block separable and, with all $\{\hat{\gamma}_k, k \neq j\}$ fixed except the jth, we may write

$$-Z_j^\top (r_j - Z_j \hat{\gamma}_j) + \lambda \hat{s}_j = 0$$

  where $r_i = y - \sum_{k \neq j} Z_k \hat{\gamma}_k$ is the jth *partial residual*

- Therefore,

$$\hat{\gamma}_j = \begin{cases} 0 & \text{if } \|Z_j^\top r_j\|_2 < \lambda \\ \left(Z_j^\top Z_j + \frac{\lambda}{\|\hat{\gamma}_j\|_2} I \right)^{-1} Z_j^\top r_j & \text{if } \|Z_j^\top r_j\|_2 \geq \lambda \end{cases}$$

Computation for the Group Lasso

- This equation does not have a closed-form solution for $\hat{\gamma}_j$ unless Z_j is orthonormal, with

$$\left(1 - \frac{\lambda}{\|Z_j^\top r_j\|_2} I\right)_+ Z_j^\top r_j$$

  where $(t)_+ = \max\{0, t\}$
- To solve for γ, one can apply this closed-form solution to each group sequentially, as we did with coordinate descent
- however, the closed form solution assumed orthonormality $\frac{1}{n} Z_j^\top Z_j = I$, which is not the case in general
- We can always make this assumption hold by transforming to the orthonormal case and then transforming back to obtain solutions on the original scale
  Block coordinate descent

Composite gradient methods

- Although there is an initial cost in terms of computing the *singular value decomposition* for each group, once this is done the cost per iteration for the group descent algorithm is simply $O(np)$
- Because the penalty is separable in terms of the groups γ_j, and because we are updating whole groups at once, the algorithm is guaranteed to decrease the objective function with every step and to converge to a minimum, as it was for the ordinary lasso
- Extensions to other loss functions (GLMs, etc.) are available similarly at the lasso case

Composite gradient methods

- Algorithm that is also iterative within each block.
- At each iteration the block-optimization problem is approximated by an easier problem, for which an easy update is possible
- The updates take the form

$$\begin{aligned} w &\leftarrow \gamma_j + \nu \cdot Z_j^\top (r_j - Z_j \hat{\theta}_j) \\ \hat{\theta}_j &\leftarrow \left(1 - \frac{\nu\lambda}{\|w\|_2}\right)_+ w \end{aligned}$$

where ν is a step-size parameter.

Multilevel categorical variable

- Consider the simple case of one continuous predictor x and a three-level factor G with levels g_1, g_2, and g_3.
- The linear model

$$\mathbb{E}\{y|x, G\} = x\beta + \sum_{k=1}^{3} \gamma_k I_k(G)$$

  where $I_k(G)$ is a 0-1 valued indicator function for the event $\{G = g_k\}$
- Linear regression in x with different intercepts γ_k depending on the level of G.
- Introduce a vector $Z = (z_1, z_2, z_3)$ of three indicator variables with $Z_k = I_k(G)$ the model is

$$\mathbb{E}\{y|x, G\} = \mathbb{E}\{y|x, Z\} = x\beta + Z^\top \gamma,$$

  where $\gamma = [\gamma_1, \gamma_2, \gamma_3]^\top$
- If G (or equally Z) has no predictive power $\Rightarrow$ the full $\gamma = 0$. When G is useful for prediction $\Rightarrow$ all coefficients of γ are nonzero.

Multilevel categorical variable

- When many of such 'single' and 'group' variables, the linear model is

$$\mathbb{E}\{y|X, G_1, \ldots, G_J\} = \beta_0 + X^\top \beta + \sum_{j=1}^{J} Z_j^\top \gamma_j$$

- With factors, usually, one has to worry about aliasing, i.e., the dummy variables in a set add to one, which is aliased with the intercept term
- we need to use contrasts to code factors that enforce, for example, that coefficients in a group sum to zero.
- With the group lasso this is not a concern, because of the L_2 penalties.
- The symmetric full representation can be used: the penalty term ensures that the coefficients in a group sum to zero

Back to Ames (Iowa) housing data

- Explanatory variables:

Order: **Observation number**
PID: **Parcel identification number - can be used with city web site for parcel review.**
MS SubClass: **Identifies the type of dwelling involved in the sale.**
MS Zoning: **Identifies the general zoning classification of the sale.**
Lot Frontage: **Linear feet of street connected to property**
Lot Area: **Lot size in square feet**
Street: **Type of road access to property**
Alley: **Type of alley access to property**
Lot Shape: **General shape of property**
Land Contour: **Flatness of the property**
Utilities: **Type of utilities available**
Lot Config: **Lot configuration**
Land Slope: **Slope of property**
Neighborhood: **Physical locations within Ames city limits (map available)**
Condition 1: **Proximity to various conditions**
Condition 2: **Proximity to various conditions (if more than one is present)**
Bldg Type: **Type of dwelling**
House Style: **Style of dwelling**
Overall Qual: **Rates the overall material and finish of the house**
Overall Cond: **Rates the overall condition of the house**
Year Built: **Original construction date**
Year Remod/Add: **Remodel date (same as construction date if no remodeling or additions)**
Roof Style: **Type of roof**
Roof Matl: **Roof material**
Exterior 1: **Exterior covering on house**
Exterior 2: **Exterior covering on house (if more than one material)**
Mas Vnr Type: **Masonry veneer type**
Mas Vnr Area: **Masonry veneer area in square feet**
Exter Qual: **Evaluates the quality of the material on the exterior**
Exter Cond: **Evaluates the present condition of the material on the exterior**
Foundation: **Type of foundation**
Bsmt Qual: **Evaluates the height of the basement**
Bsmt Cond: **Evaluates the general condition of the basement**
Bsmt Exposure: **Refers to walkout or garden level walls**
BsmtFin Type 1: **Rating of basement finished area**
BsmtFin SF 1: **Type 1 finished square feet**
BsmtFinType 2: **Rating of basement finished area (if multiple types)**
BsmtFin SF 2: **Type 2 finished square feet**
Bsmt Unf SF: **Unfinished square feet of basement area**
Total Bsmt SF: **Total square feet of basement area**
Heating: **Type of heating**
HeatingQC: **Heating quality and condition**
Central Air: **Central air conditioning**
Electrical: **Electrical system**
1st Flr SF: **First Floor square feet**
2nd Flr SF: **Second floor square feet**
Low Qual Fin SF: **Low quality finished square feet (all floors)**
Gr Liv Area: **Above grade (ground) living area square feet**
Bsmt Full Bath: **Basement full bathrooms**
Bsmt Half Bath: **Basement half bathrooms**
Full Bath: **Full bathrooms above grade**
Half Bath: **Half baths above grade**
Bedroom: **Bedrooms above grade (does NOT include basement bedrooms)**
Kitchen: **Kitchens above grade**
KitchenQual: **Kitchen quality**
TotRmsAbvGrd: **Total rooms above grade (does not include bathrooms)**
Functional: **Home functionality (Assume typical unless deductions are warranted)**
Fireplaces: **Number of fireplaces**
FireplaceQu: **Fireplace quality**
Garage Type: **Garage location**
Garage Yr Blt: **Year garage was built**
Garage Finish: **Interior finish of the garage**
Garage Cars: **Size of garage in car capacity**
Garage Area: **Size of garage in square feet**
Garage Qual: **Garage quality**
Garage Cond: **Garage condition**
Paved Drive: **Paved driveway**
Wood Deck SF: **Wood deck area in square feet**
Open Porch SF: **Open porch area in square feet**
Enclosed Porch: **Enclosed porch area in square feet**
3-Ssn Porch: **Three season porch area in square feet**
Screen Porch: **Screen porch area in square feet**
Pool Area: **Pool area in square feet**
Pool QC: **Pool quality**
Fence: **Fence quality**
Misc Feature: **Miscellaneous feature not covered in other categories**
Misc Val: **$Value of miscellaneous feature**
Mo Sold: **Month Sold**
Yr Sold: **Year Sold**
Sale Type: **Type of sale**
Sale Condition: **Condition of sale**

Back to Ames (Iowa) housing data

- Many categorical variables. For example,

`Lot Shape`: *General shape of property*
(3 levels)
Slightly_Irregular
Moderately_Irregular
Irregular

`Bldg Type`: *Type of dwelling*
(4 levels)
TwoFmCon
Duplex
Twnhs
TwnhsE

`House Style`: *Style of dwelling*
(6 levels)
One_and_Half_Unf
One_Story
SFoyer
SLvl
Two_and_Half_Fin
Two_and_Half_Unf
Two_Story

`Overall Qual`: *Rates the overall material and finish of the house*
(9 levels)
Poor
Fair
Below_Average
Average
Above_Average
Good
Very_Good
Excellent
Very_Excellent

`Mas Vnr Type`: *Masonry veneer type*
(4 levels)
BrkFace
CBlock
None
Stone

`Bsmt Qual`: *Evaluates the height of the basement*
(5 levels)
Fair
Good
No_Basement
Poor
Typical

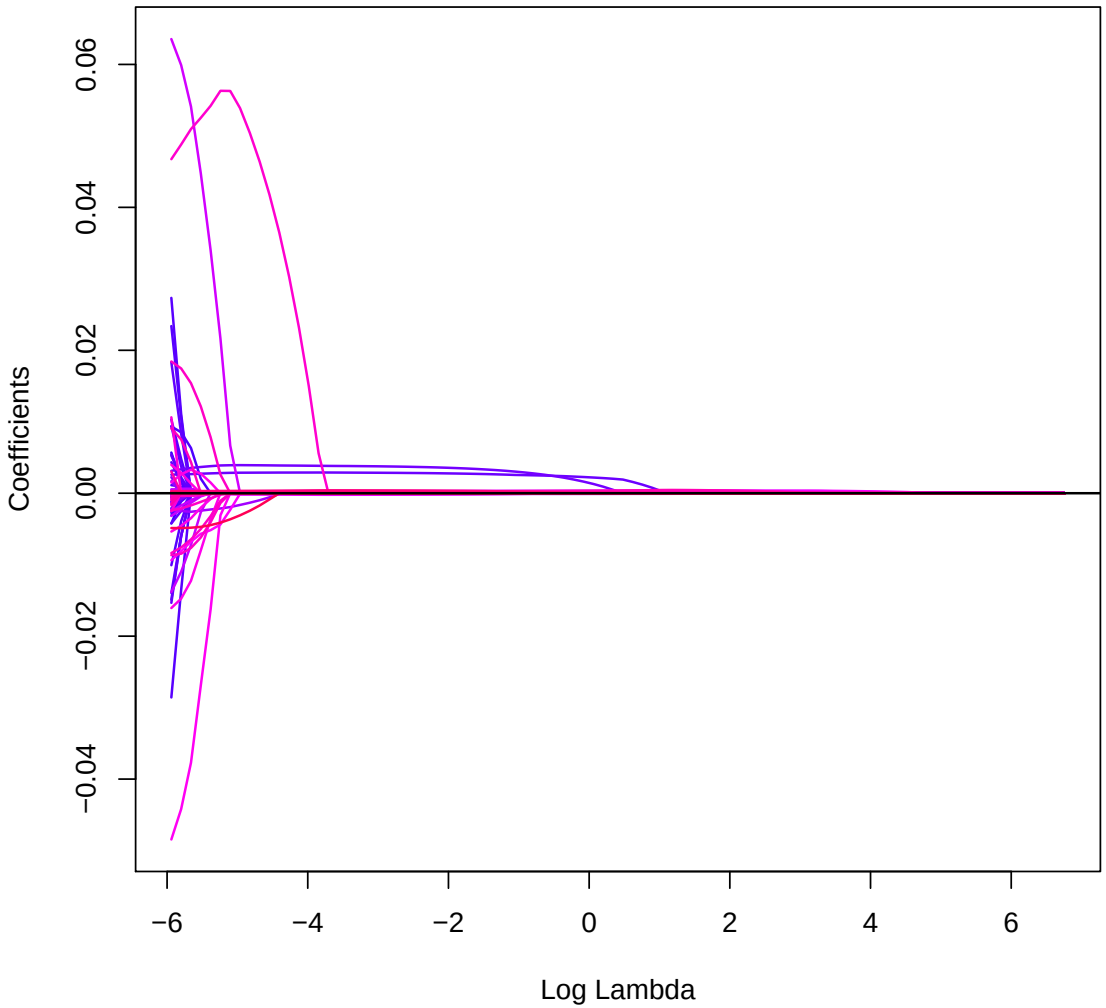
`Garage Type`: *Garage location*
(6 levels)
Basment
BuiltIn
CarPort
Detchd
More_Than_Two_Types
No_Garage

`Pool QC`: *Pool quality*
(4 levels)
Fair
Good
No_Pool
Typical

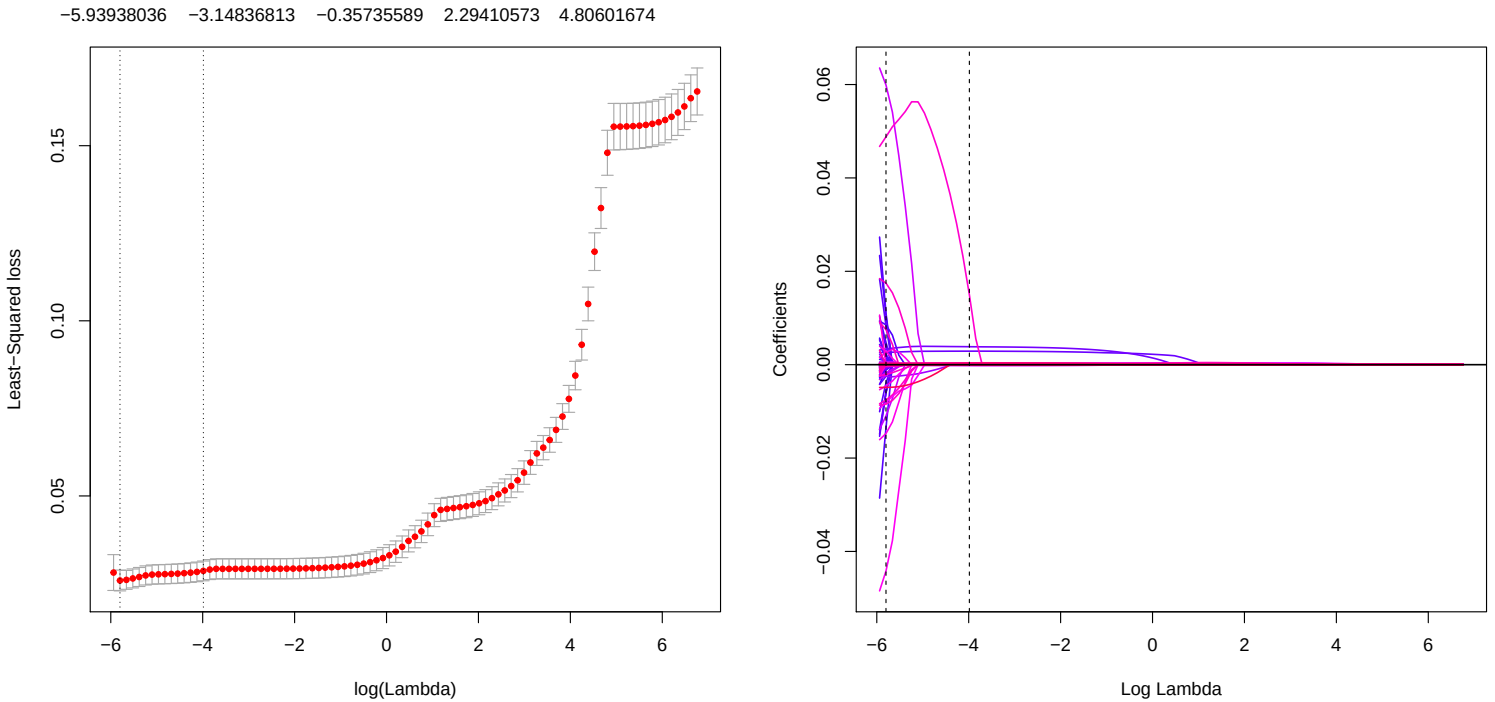
`Fence`: *Fence quality*
(4 levels)
Good_Wood
Minimum_Privacy
Minimum_Wood_Wire
No_Fence

`Sale Condition`: *Condition of sales*
(5 levels)
AdjLand
Alloca
Family
Normal
Partial

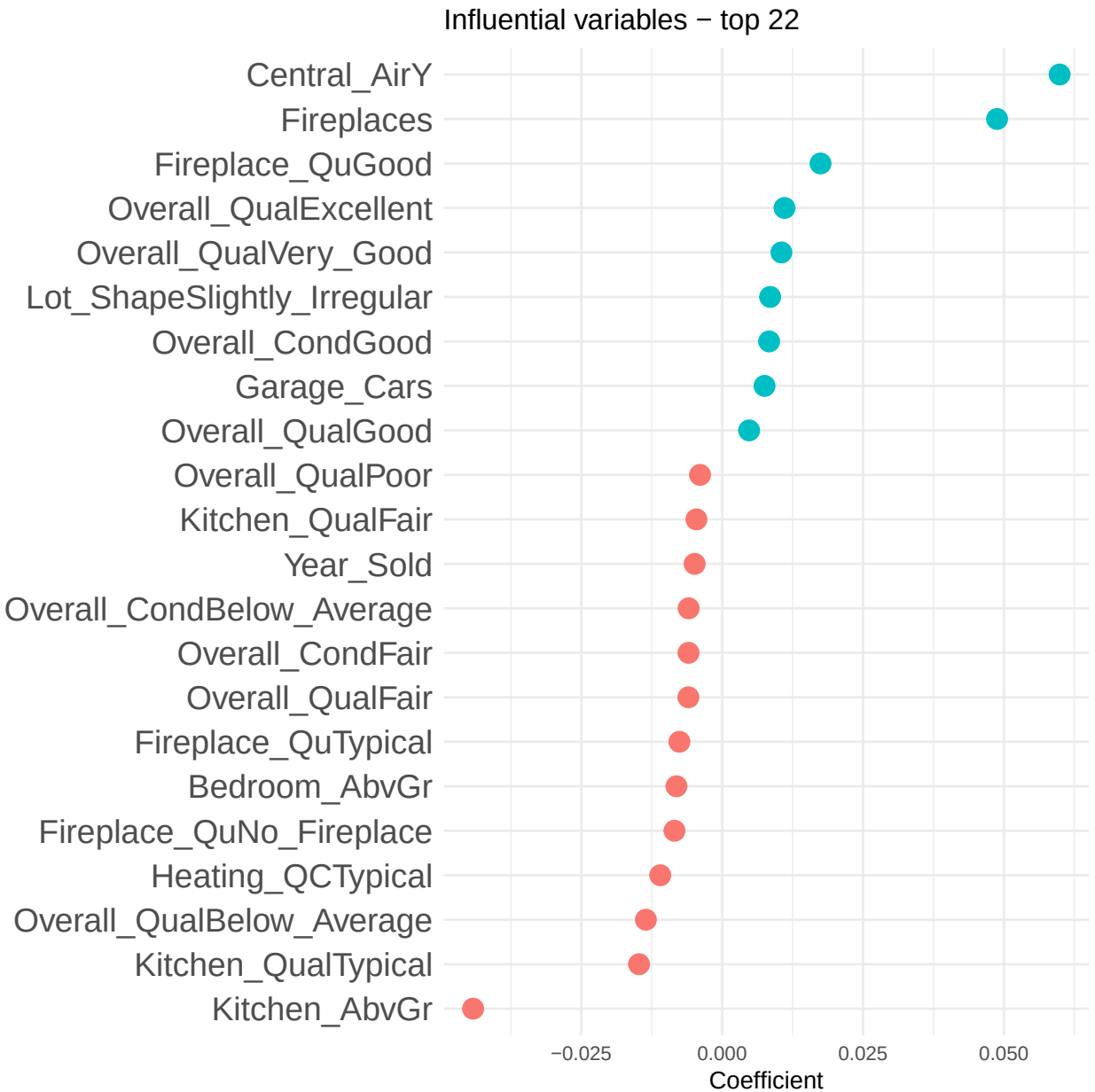
Ames (Iowa) housing data



Ames (Iowa) housing data



Ames (Iowa) housing data

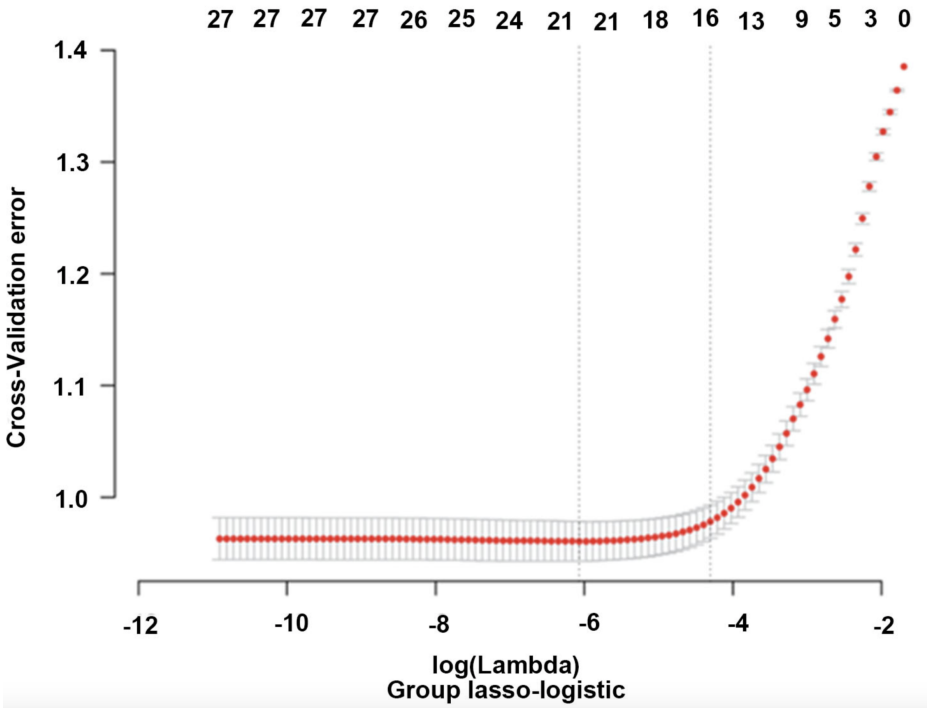


Back to credit scoring example

- $n = 3521$ records, $p = 25$ variables.
- training set: 3002 data (1500 compliance, 1502 default),
  test set: 519 data (258 compliance data, 261 default)
- Response: '0' performance customer; '1' default customer.
- Explanatory variables. Basic personal identity information: domicile, gender, local work, education level and marital status; Personal economic ability: whether there is a CPF salary level; Personal debt and debt repayment record: frequency of personal housing loan, personal commercial housing loan pen number and frequency of other loan credit card account number, number, frequency of delinquent loans, loans overdue month loan highest monthly overdue amount, maximum length, loan account number of the contract amount, loan balance has been used lines, the average individual loan maximum contract value, the average individual loans minimum contract amount, the last six months on average use; Total number of times of individual approval query and loan number
- Fit the logistic lasso and SVM - lasso,
- 10-fold cross-validation for selecting λ

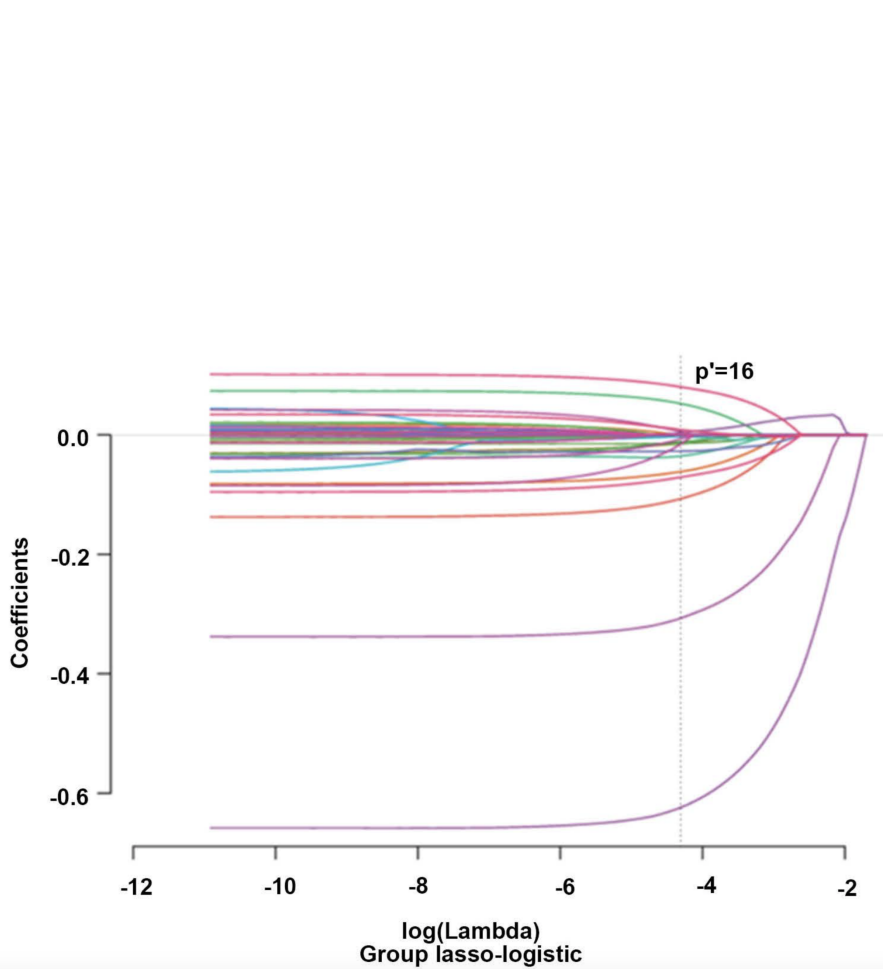
ID	Variable Name	The Values
y	default	$y=0$ Not default; $y=1$ Default
x_1	registered area	$x_{1,1}$ = Northeast; $x_{1,2}$ = North China Plain; $x_{1,3}$ = Central China; $x_{1,4}$ = Eastern China
x_2	gender	$x_{2,1}$ = Male; $x_{2,2}$ = Female
x_3	Whether work is local	$x_{3,1}$ = Local; $x_{3,2}$ = Not local
x_4	edu level	$x_{4,1}$ = Junior high school/Senior high school/others; $x_{4,2}$ = Junior college/Junior college and below; $x_{4,3}$ = Undergraduate; $x_{4,4}$ = Master/Doctor
x_5	marital status	$x_{5,1}$ = Maid; $x_{5,2}$ = Married; $x_{5,3}$ = Others (divorced/widowed)
x_6	Whether accumulation fund	$x_{6,1}$ = Not; $x_{6,2}$ = Yes
x_7	pay grades	$x_{7,1}$ = 0 - 3500; $x_{7,2}$ = 3501 - 8000; $x_{7,3}$ = 8000 above
x_8	The number of individual housing loans	$x_8 \in N$
x_9	The number of individual commercial housing loans	$x_9 \in N$
x_{10}	Other loans	$x_{10} \in N$
x_{11}	Debit card account number	$x_{11} \in N$
x_{12}	The number of overdue loans	$x_{12} \in N$
x_{13}	Months of overdue loans	$x_{13} \in N$
x_{14}	The maximum amount of overdue loans per month	$x_{14} \in [0,+\infty]$
x_{15}	Maximum length of loan (year)	$x_{15} \in N$
x_{16}	Total number of approval inquiries	$x_{16} \in N$
x_{17}	Loan number	$x_{17} \in N$
x_{18}	Loan account number	$x_{18} \in N$
x_{19}	contract amount	$x_{19} \in [0,+\infty]$
x_{20}	loan balance	$x_{20} \in [0,+\infty]$
x_{21}	Have used limit	$x_{21} \in [0,+\infty]$
x_{22}	Average maximum contract amount for a single lender	$x_{22} \in [0,+\infty]$
x_{23}	Average minimum contract amount for a single lender	$x_{23} \in [0,+\infty]$
x_{24}	Average usage in the last 6 months	$x_{24} \in [0,+\infty]$

Credit scoring - cross validation



- Group lasso-logistic, $\lambda = 0.0163$.

Credit scoring - coefficient estimates



Variate	Full variables	Forward	Backwards	Lasso-logistic	Lasso-SVM	Group-lasso
X_{1_1}	1.715	0	1.721	0	−0.555	0.025
X_{1_2}	1.674	0	1.695	0.010	−0.567	0.001
X_{1_3}	−4.562	−4.247	−4.561	−4.354	0.667	−6.520
X_{1_4}	−1.534	−1.681	0	−1.440	0.016	−1.637
X_2	−0.868	−0.854	−0.858	−0.630	0.209	−0.824
X_3	−0.497	−0.504	−0.502	−0.352	0.122	−0.471
X_{4_1}	1.408	0.341	0.501	0.105	−0.003	0.207
X_{4_2}	0.780	0.301	0.218	0	0	−0.007
X_{4_3}	0.575	0	0	−0.111	0.015	−0.195
X_{4_4}	0.761	0	0	0	0	−0.682
X_{5_1}	−0.025	−0.002	0	0	0	0.062
X_{5_2}	−0.179	0	−0.201	−0.037	0	−0.072
X_{5_3}	−0.034	0	0	0	0	0.073
X_6	0.072	0	0	0	0	0.065
X_{7_1}	0.306	0.393	0.311	0.355	−0.085	0.570
X_{7_2}	−0.893	0.205	−1.111	−0.582	0.226	−0.554
X_{7_3}	0.934	0	0	0	0	0.251
X_8	0.123	0.101	0.098	0	0	0
X_9	0.058	0	0	0	0	0.039
X_{10}	−0.265	−0.213	−0.212	−0.104	0.053	−0.194
X_{11}	−0.078	0	0	−0.003	0.013	−0.016
X_{12}	−0.100	−0.111	−0.109	−0.084	0.033	−0.108
X_{13}	−0.190	−0.206	−0.197	−0.064	0.050	−0.170
X_{14}	−0.033	0	0	0	0	−0.021
X_{15}	0.130	0.121	0.119	0	−0.016	0.101
X_{16}	0.455	0.459	0.457	0.306	−0.142	0.431
X_{17}	0.222	−0.218	−0.209	−0.173	0.084	−0.179
X_{18}	0.164	0	0	0	0	0
X_{19}	−0.405	0	0	−0.021	0	−0.062
X_{20}	0.276	0	0	0	0	0
X_{21}	0.136	0	0	0	0	0
X_{22}	0.065	0	0	0	0	0
X_{23}	0.041	0	0	0	0	0.035
X_{24}	−0.325	−0.249	−0.263	−0.178	0.067	−0.212
Intercept term	1.026	1.275	1.836	2.575	−0.382	2.980

Credit scoring - correct classifications

Model	Training set			Test set		
	Good	Bad	Total	Good	Bad	Total
Full variables	71.3	75.6	73.4	67.0	71.7	69.3
Forward selection	70.7	75.6	73.1	65.5	72.1	68.9
Backward selection	72.3	73.8	73.1	69.3	67.1	68.2
Lasso-logistic	74.5	80.0	77.2	74.7	79.5	77.1
Lasso-SVM	73.7	80.2	76.9	73.6	80.2	76.8
Group lasso	74.4	79.4	76.9	75.1	78.0	76.5

- In credit risk assessment, misclassification of default users into non-defaulting users is more of a potential loss to banks or society.
- The model is more important for to correctly classify the default users than to take non-defaulting users into consideration.

Multivariate regression

- Consider q response variables $Y \in \mathbb{R}^q$
- Goal: contemporary model Y on the basis of $X \in \mathbb{R}^p$

$$Y = XB + E$$

here Y and E have q columns and n rows $(\tilde{Y}_i, \tilde{E}_i)$,
B is the *matrix* of parameters $(p \times q)$
$\text{var}\left\{\tilde{E}_i\right\} = \Sigma$

- Least squares $\hat{Y} = X\hat{B}, \qquad \hat{B} = (X^\top X)^{-1} X^\top Y$
- $\hat{\Sigma} = \frac{1}{n-p} Y^\top (I - P)\, Y \qquad$ where $P = X(X^\top X)^{-1} X^\top$
- Can be seen as a collection of q standard regression problems in $\mathbb{R}^p$
- Lasso version: we could fit a separate regression coefficient vector β_q for each of the q different problems, using the lasso in the case of a sparse linear model.

Multivariate regression

- In the setting of sparsity, sometimes, we may look for an unknown subset of the covariates that are relevant for prediction, and this same subset is preserved across all q components of the response variable.
- Consider a group lasso penalty, in which the p groups are defined by the rows $\{\tilde{\beta}_j \in \mathbb{R}^q, j = 1, \ldots, p\}$ of the full coefficient matrix $B \in \mathbb{R}^{p \times q}$
- The group lasso regularized least squares problem

$$\min_{B \in \mathbb{R}^{p \times q}} \left\{ \frac{1}{2} \|Y - XB\|_F^2 + \lambda \left(\sum_{j=1}^{p} \|\tilde{\beta}_j\|_2 \right) \right\}$$

  where $\|\cdot\|_F$ is the 'Frobenius norm', which is the L_2-norm applied to the entries of the matrix.
- This is a group lasso with $J = p$ and $p_j = q, \forall j$

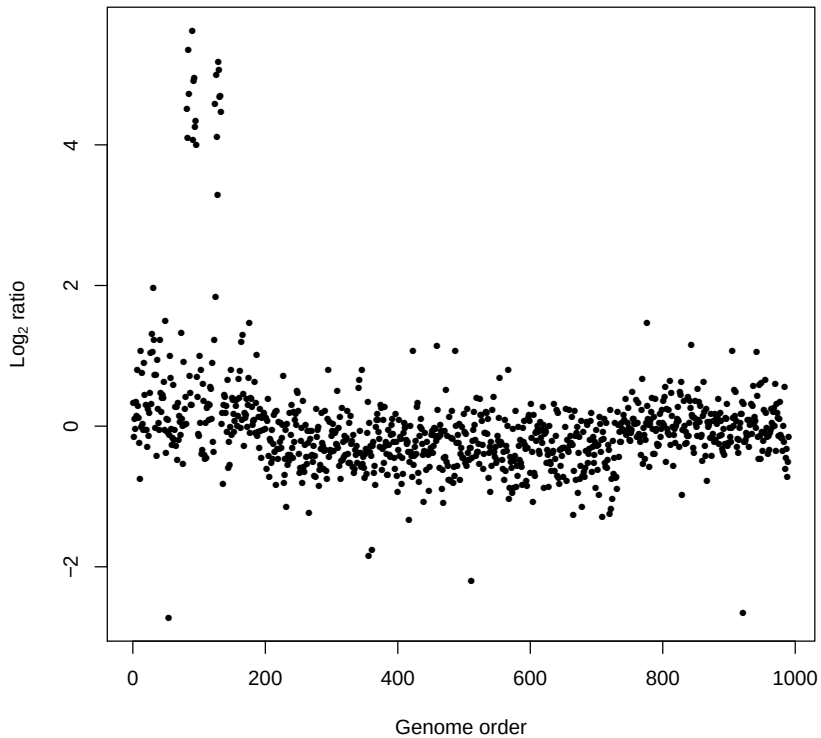
GCH

- Broadly speaking, humans have two copies of their genome
- Occasionally however, a region of the genome is duplicated or destroyed; this is known as copy number variation (CNV) and it occurs in all humans
- Copy number variation tends to be more extreme in cancer: gains or losses of large regions of the genome often trigger uncontrolled cell growth
- There are a variety of methods for measuring copy number variation in a genome-wide fashion
- the data comes from a method known as comparative genomic hybridization (CGH)

GCH data

- CGH data from two glioblastoma tumors (chromosome 7 in one patient, chromosome 13 in another) spliced together in order to create a challenging data set for CNV detection:
  - Both gains and losses are present
  - The copy number changes occur over both short and large scales
- CGH data is typically reported on the $\log_2$ ratio scale, so that 0 means 2 copies (i.e., a normal number of copies), $\log_2 \frac{3}{2} = 1$ means a gain of a copy, and $\log_2 \frac{1}{2} = -1$ means the loss of a copy
- Biological considerations dictate that it is typically segments of a chromosome – rather than individual genes – that are replicated

glioma data



- Each spike represents the copy number of a gene in a tumor sample, relative to that of a control (on the log base-2 scale).
- the copy numbers are plotted against the chromosome order of the gene.
- Data are very noisy, so that some kind of smoothing is essential.
- We might expect that the underlying vector of true copy numbers to be piecewise-constant over contiguous regions of a chromosome.

Fused lasso

- The fused lasso estimates $\hat{\beta}$ are the values minimizing the following objective function:

$$\min_{\beta_0,\beta}\left\{\frac{1}{2}\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}x_{ij}\beta_j\right)^2+\lambda_1\sum_{j=1}^{p}|\beta_j|+\lambda_2\sum_{j=2}^{p}|\beta_j-\beta_{j-1}|\right\}$$

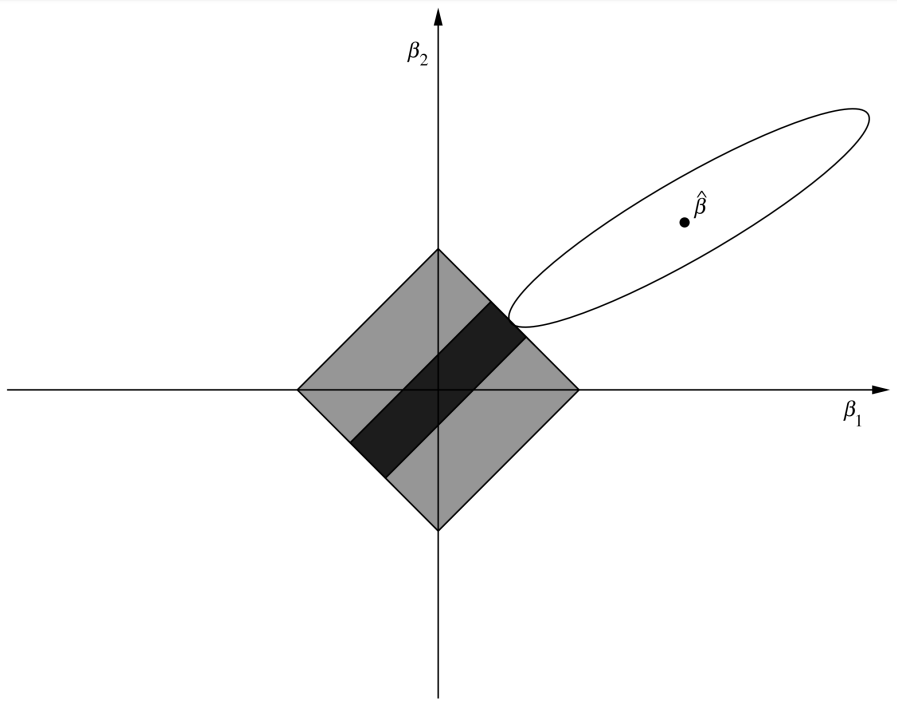
- here covariates x_{ij} and their coefficients β_j are indexed along some sequence j for which neighborhood clumping makes sense
- The penalty consists of two pieces:
  - A lasso penalty that encourages $\beta_j = 0$
  - A fusion penalty that encourages β_j to be equal to β_{j-1} and β_{j+1}

Fused lasso

$$\min_{\beta_0,\beta}\left\{\frac{1}{2}\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}x_{ij}\beta_j\right)^2\right\} \quad \text{subject to} \quad \sum_{j=1}^{p}|\beta_j|\le s_1$$

$$\text{and} \quad \sum_{j=2}^{p}|\beta_j-\beta_{j-1}|\le s_2$$

- Schematic diagram of the fused lasso
- $n > p = 2$
- we seek the first time that the contours of the sum-of-squares loss function satisfy $\sum_j |\beta_j| = |\beta_1| + |\beta_2| = s_1$ (gray) and $\sum_j |\beta_j - \beta_{j-1}| = |\beta_2 - \beta_1| = s_2$ (black)



Fused lasso signal approximator

- A special case of the fused lasso is the situation where $X = I$, i.e., only the "index" variables are present
- Let use $\hat{\theta}$ to denote the solutions to this problem of minimizing

$$\min_{\theta \in \mathbb{R}^n} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \theta_i)^2 + \lambda_1 \sum_{i=1}^{n} |\theta_i| + \lambda_2 \sum_{i=2}^{n} |\theta_i - \theta_{i-1}| \right\}$$

- This version of the problem is sometimes called the fused lasso signal approximator, in the sense that it amounts to approximating a one-dimensional signal with a series of zeroes and piecewise constant functions
- The second penalty encourages neighboring coefficients θ_i to be similar, and will cause some to be identical (also known as total-variation denoising). Now we obtain sparsity in adjacent differences $\hat{\theta}_i - \hat{\theta}_{i-1}$, i.e., we obtain $\hat{\theta}_i = \hat{\theta}_{i-1}$ at many locations i
- A constant term θ_0 is not included; the coefficient θ_i represents the response y_i directly, and for these kinds of problems zero is a natural origin.
- Hence, plotted in order of the locations $i = 1, \ldots, n$, the solution $\hat{\theta}$ appears piecewise constant

Two dimensional fused lasso

- We can generalize the notion of neighbors, for examples adjacent pixels in an image. This leads to a penalty of the form

$$\lambda_2 \sum_{i,i'} |\theta_i - \theta_{i'}|$$

- Therefore, the two-dimensional fused lasso (in *signal approximator* form) minimizes

$$\min_{\theta \in \mathbb{R}^n} \left\{ \frac{1}{2} \sum_{i=1}^{n} \sum_{i'=1}^{n} \left(y_{i,i'} - \theta_{i,i'}\right)^2 + \right.$$

$$\left. \lambda_1 \sum_{i=1}^{n} \sum_{i'=1}^{n} |\theta_{i,i'}| + \lambda_2 \sum_{i,i'} \left(|\theta_{i,i'} - \theta_{i-1,i'}| + |\theta_{i,i'} - \theta_{i,i'-1}| \right) \right\}$$

  the first term is the Frobenius norm: $\|A\|_F = \sum_{i,j} a_{ij}^2$

- Can be generalized for non discrete neighbors (sums over $|i - i'| < \ell$)

Coordinate descent: Unsuitable?

- Solving this optimization problem, however, introduces some new challenges that we have not yet encountered
- The difference penalty is not a separable function of the coordinates
- Coordinate descent can become "stuck" at a nonoptimal point
- As we will see, coordinate descent does not work well at all for solving the fused lasso problem; new tools are needed

Toy example

- Consider a toy data set:

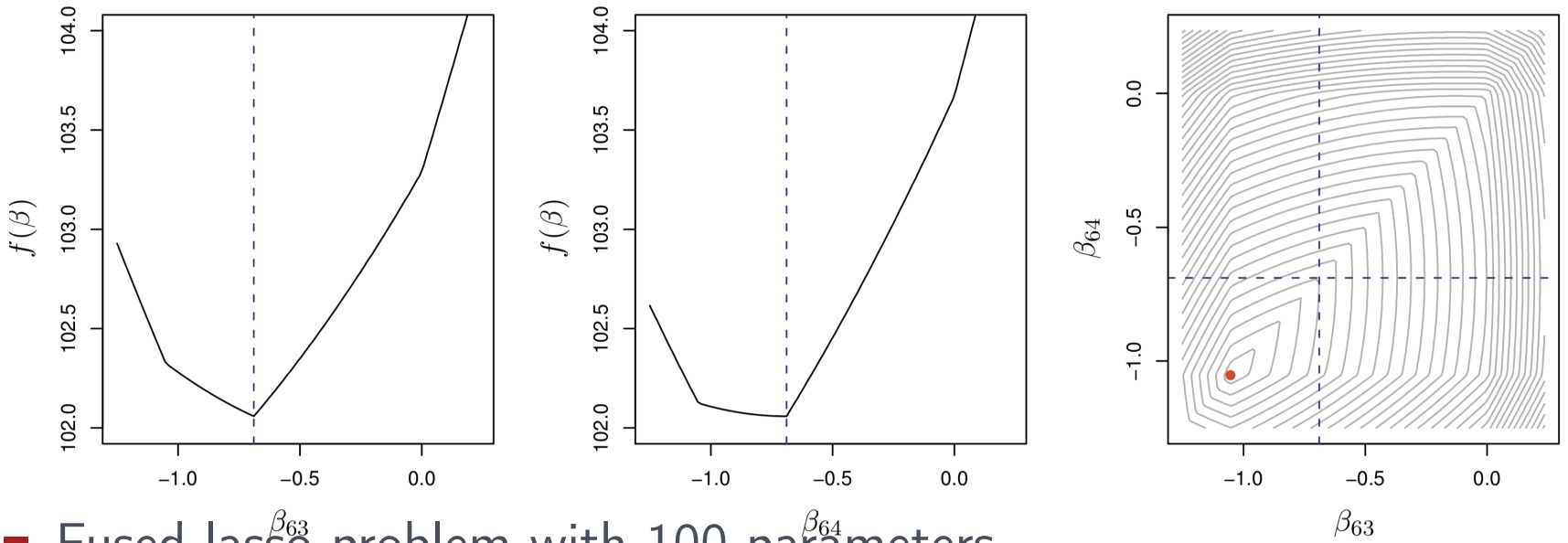
$$y = [0, 0, 0, 1, 1, 1, 0, 0, 0]$$

- For the purposes of illustration, let $\lambda_1 = 0$ and $\lambda_2 = \frac{1}{2}$
- Call the loss function

$$Q(\theta, y) = \min_{\theta \in \mathbb{R}^n} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \theta_i)^2 + \lambda_1 \sum_{i=1}^{n} |\theta_i| + \lambda_2 \sum_{i=2}^{n} |\theta_i - \theta_{i-1}| \right\}$$

  we can see that
  $Q(y, y) = 0 + 0 + \frac{1}{2}(0 + 0 + 1 + 0 + 0 + 1 + 0 + 0) = 1$ while
  $Q(0, y) = 1.5$, so $Q(y, y) < Q(0, y)$
- Nevertheless, if we start at the initial value $\theta = 0$, the coordinate descent algorithm can never escape zero
- In fact whatever move of ε of only one θ_i from zero gives a loss of $\frac{1}{2} \cdot (3 + \varepsilon^2)$ which is larger than 1.5.
- By only considering one-coordinate-at-a-time transitions, the Coordinate Descent algorithm misses the fact that we could simultaneously move $\{\theta_4, \theta_5, \theta_6\}$ and obtain a better solution

Failure of coordinate descent



- Fused lasso problem with 100 parameters. Solutions for two of the parameters, $\beta_{63} = \beta_{64} = -1.05$
- Left and middle panels show slices of the objective function f as a function of β_{63} and β_{64}, with the other parameters set to the global minimizers.
- The coordinate-wise minimizer over β_{63} and β_{64} (separately) is -0.69
- Coordinate-descent algorithm is stuck at the point $(-0.69, -0.69)$.
- Despite being strictly convex, the surface has corners, in which the coordinate-wise procedure can get stuck.
- We have to move both β_{63} and β_{64} together.

Reframing the problem ($\lambda_1 = 0$ for simplicity)

- There are a variety of alternative algorithms we could use
- One simple approach is to reparametrize the fused optimization problem so that the penalty is additive.
- Consider a linear transformation of the form $\gamma = M\theta$ such that

$$\gamma_1 = \theta_1, \qquad \text{and } \gamma_i = \theta_i - \theta_{i-1}, \text{ for } i = 2, \ldots, n$$

- We may reframe the problem as

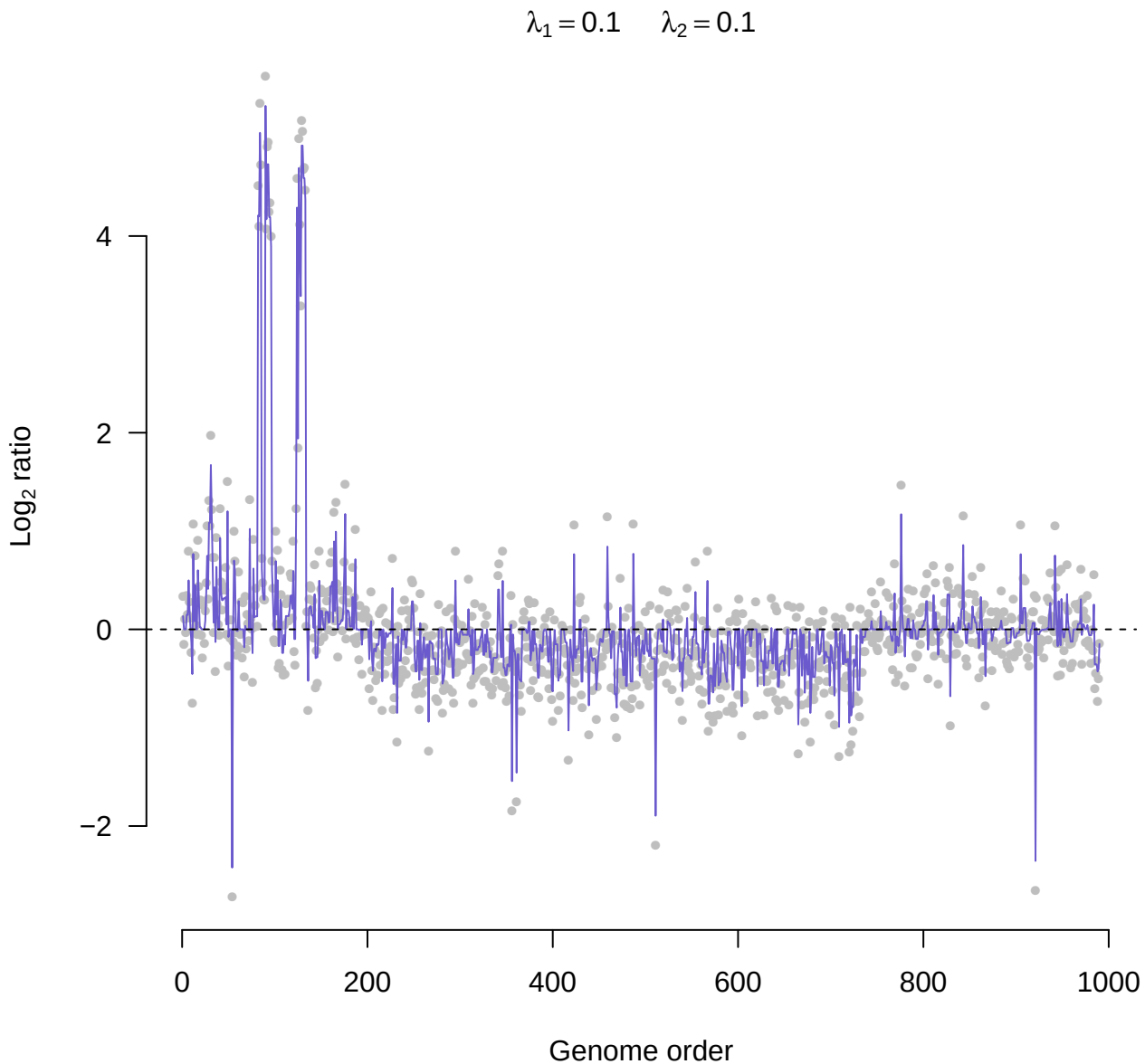
$$\min_{\gamma \in \mathbb{R}^n} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \theta)^2 + \lambda_2 \sum_{i=1}^{n} |\gamma_i| \right\} \qquad \text{where now } \theta = M^{-1}\gamma$$

- The reparametrized problem can be solved using any efficient algorithm for the lasso, including coordinate descent. Now, γ is the new vector of parameters and M^{-1} are new "variables".
- However, M^{-1} is a lower-triangular matrix with all nonzero entries equal to 1, and hence has large correlations among the "variables"
- Neither coordinate-descent nor LARS performs well under these circumstances

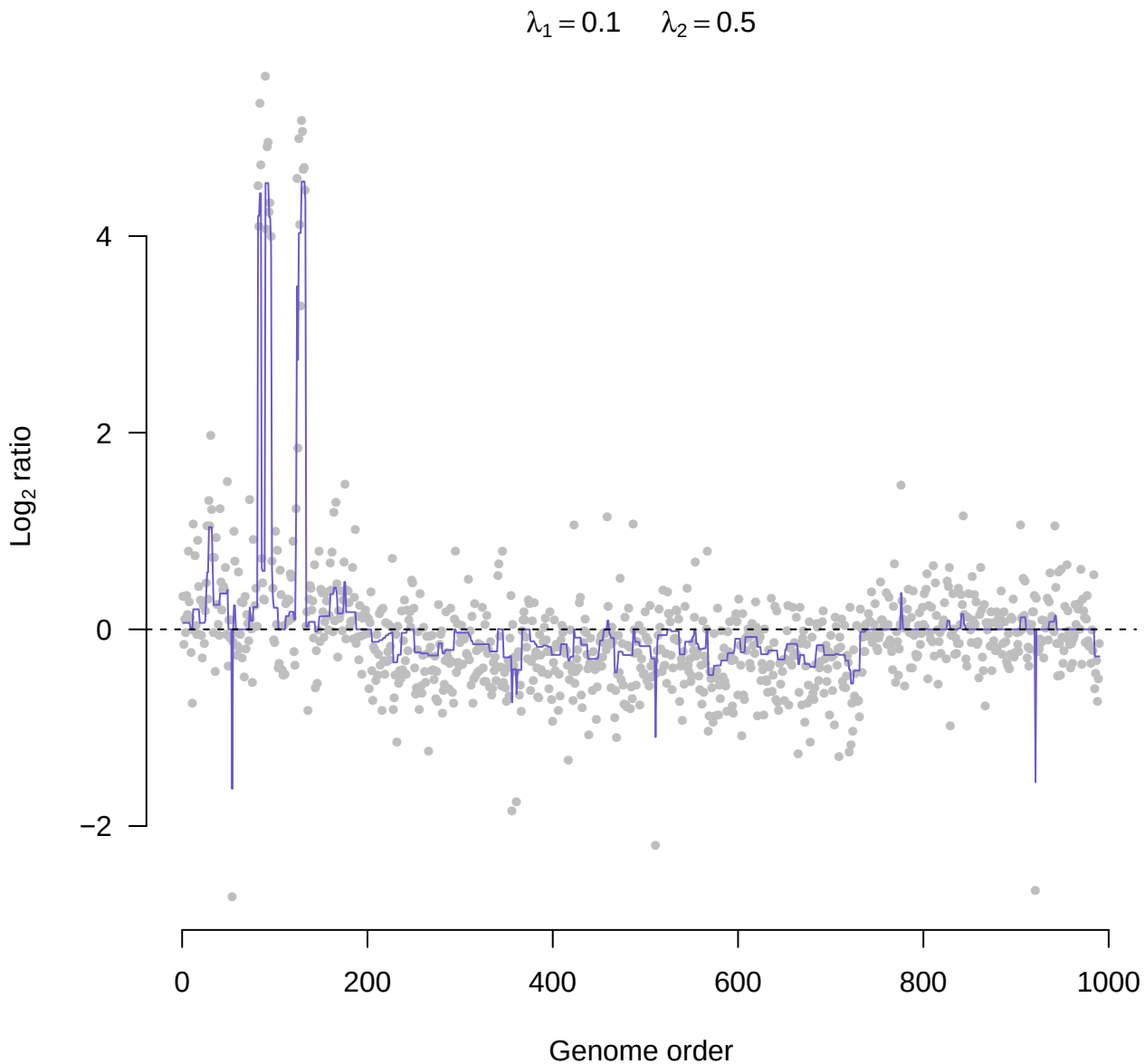
Algorithms

- Other algorithms can be used such as *alternating direction method of multipliers* (ADDM) or *path algorithms*
- The essence of the ADMM algorithm is that it alternates between updating θ, updating γ, and reconciling their differences
- ADMM algorithms converge for a wider range of problems than coordinate descent; in addition, they lend themselves to parallelization in a way that coordinate descent algorithms do not
- In the specific context of the *fused Lasso signal approximator*, there are also a variety of exact solutions (sometimes called *path algorithms*) that can be calculated using an algorithm somewhat analogous to the LARS algorithm for the regular lasso
- These exact algorithms tend to be quite a bit faster for small problems; for larger problems ADMM is often better

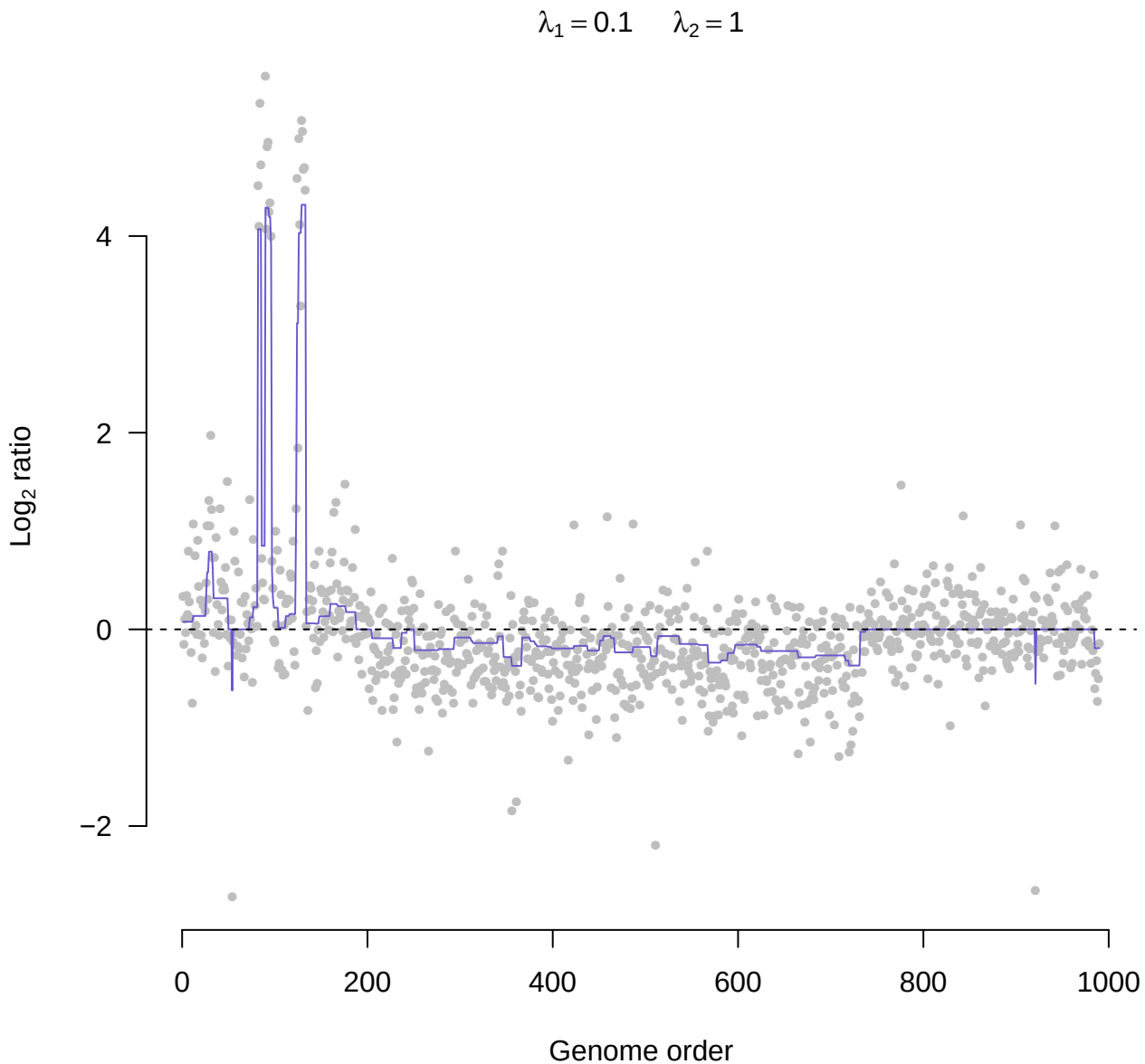
glioma data


$\lambda_1 = 0.1$ $\lambda_2 = 0.1$
Log_2 ratio
4
2
0
−2
0
200
400
600
800
1000
Genome order

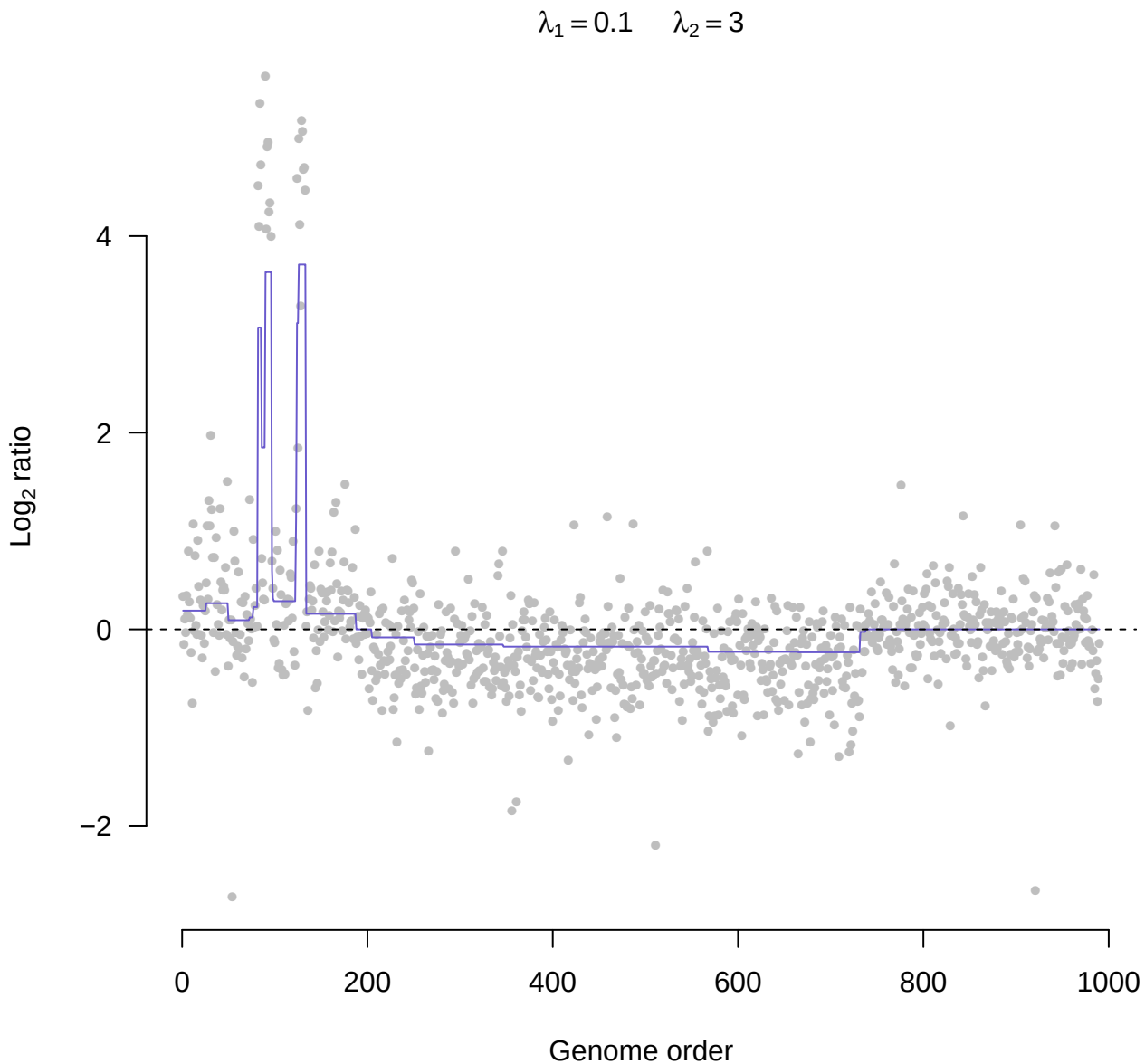
glioma data


$\lambda_1 = 0.1$ $\lambda_2 = 0.5$
Log_2 ratio
4
2
0
−2
0
200
400
600
800
1000
Genome order

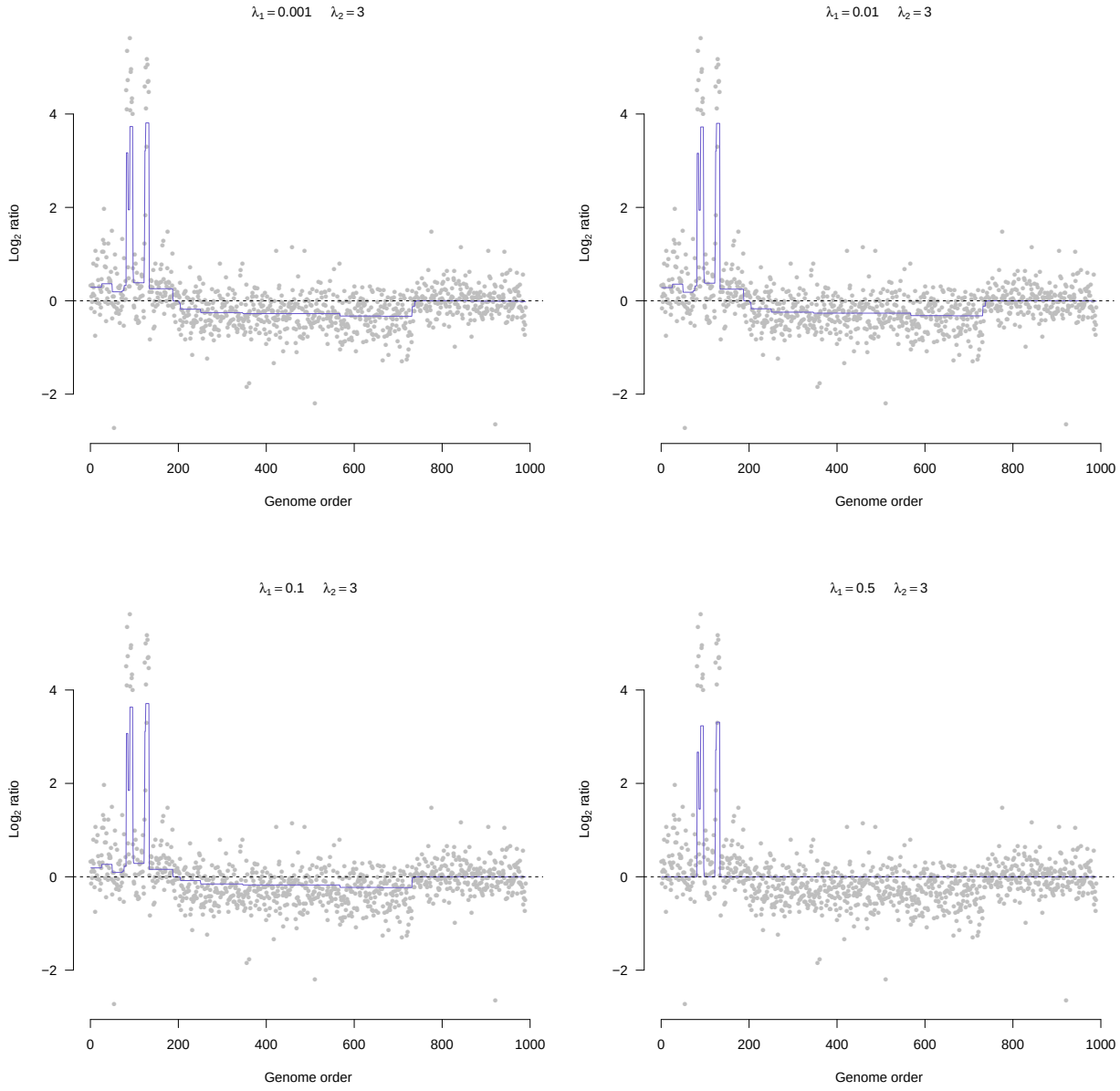
glioma data


$\lambda_1 = 0.1$ $\lambda_2 = 1$
Log_2 ratio
4
2
0
−2
0
200
400
600
800
1000
Genome order

glioma data


$\lambda_1 = 0.1$ $\lambda_2 = 3$
Log_2 ratio
4
2
0
−2
0
200
400
600
800
1000
Genome order

glioma data


λ1 = 0.001 λ2 = 3
λ1 = 0.01 λ2 = 3
λ1 = 0.1 λ2 = 3
λ1 = 0.5 λ2 = 3
Log2 ratio
Genome order
4
2
0
−2
0
200
400
600
800
1000

Further applications of penalization and sparsity

Basis functions

- Suppose for the moment that we have just a single feature x and we are interested in estimating $\mathbb{E}\{y|x\} = f(x)$
- A common approach for extending the linear model $f(x) = x\beta$ is to augment x with additional, known functions of x:

$$f(x) = \sum_{m=1}^{M} \beta_m h_m(x),$$

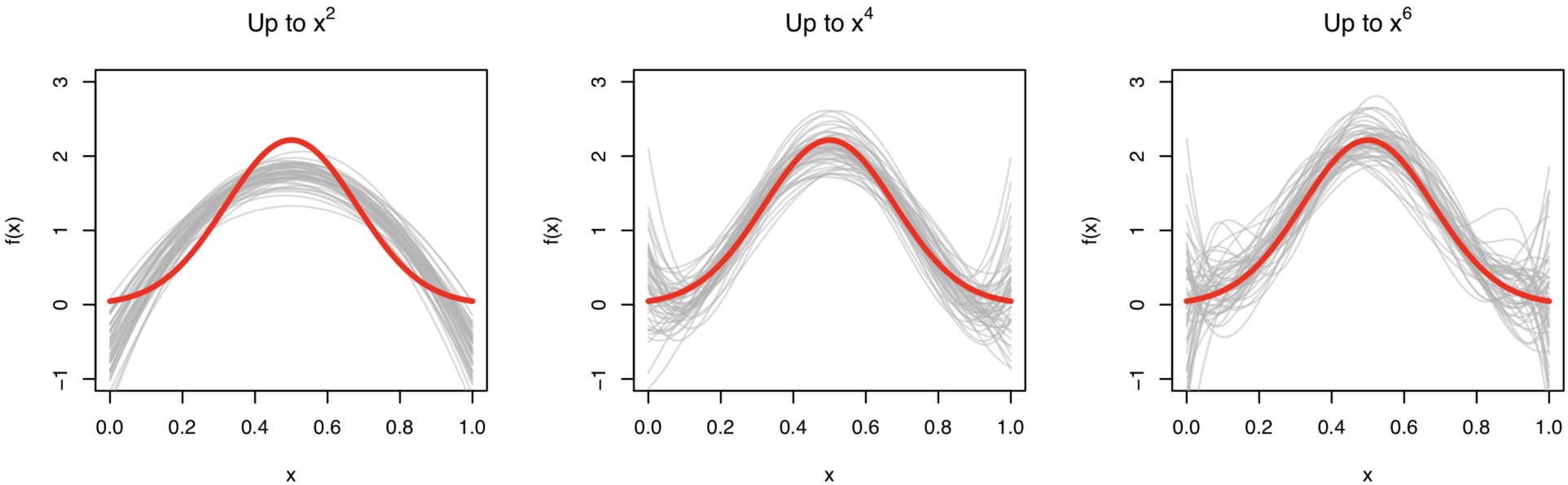
  where the $\{h_m\}$ are called basis functions
- Because the basis functions $\{h_m(\cdot)\}$ are prespecified and the model is linear in the new variables, ordinary least squares approaches can be used (at least in low-dimensional settings)

Problems with polynomial regression

- We may think to polynomial terms as $h_m(x)$
- However, polynomial terms introduce undesirable side effects: each observation affects the entire curve, even for x values far from the observation
- Not only does this introduce bias, but it also results in extremely high variance near the edges of the range of x

Problems with polynomial regression

Consider the following simulated example (50 samples; gray lines are models fit to 100 observations arising from the true f, colored red):



Regression splines

- For this reason, local basis functions, which ensure that a given observation affects only the nearby fit, not the fit of the entire line, are often preferred
- We will focus on a specific type of local bases the splines, which are just piecewise polynomials joined together to make a single smooth curve

Regression splines

- To understand splines, we will gradually build up a piecewise model, starting at the simplest one: the piecewise constant model
- First, we partition the range of x into $K+1$ intervals by choosing K points $\{\xi_k\}_{k=1}^{K}$ called knots

Basis functions for piecewise continuous models

- These constraints can be incorporated directly into the basis functions:

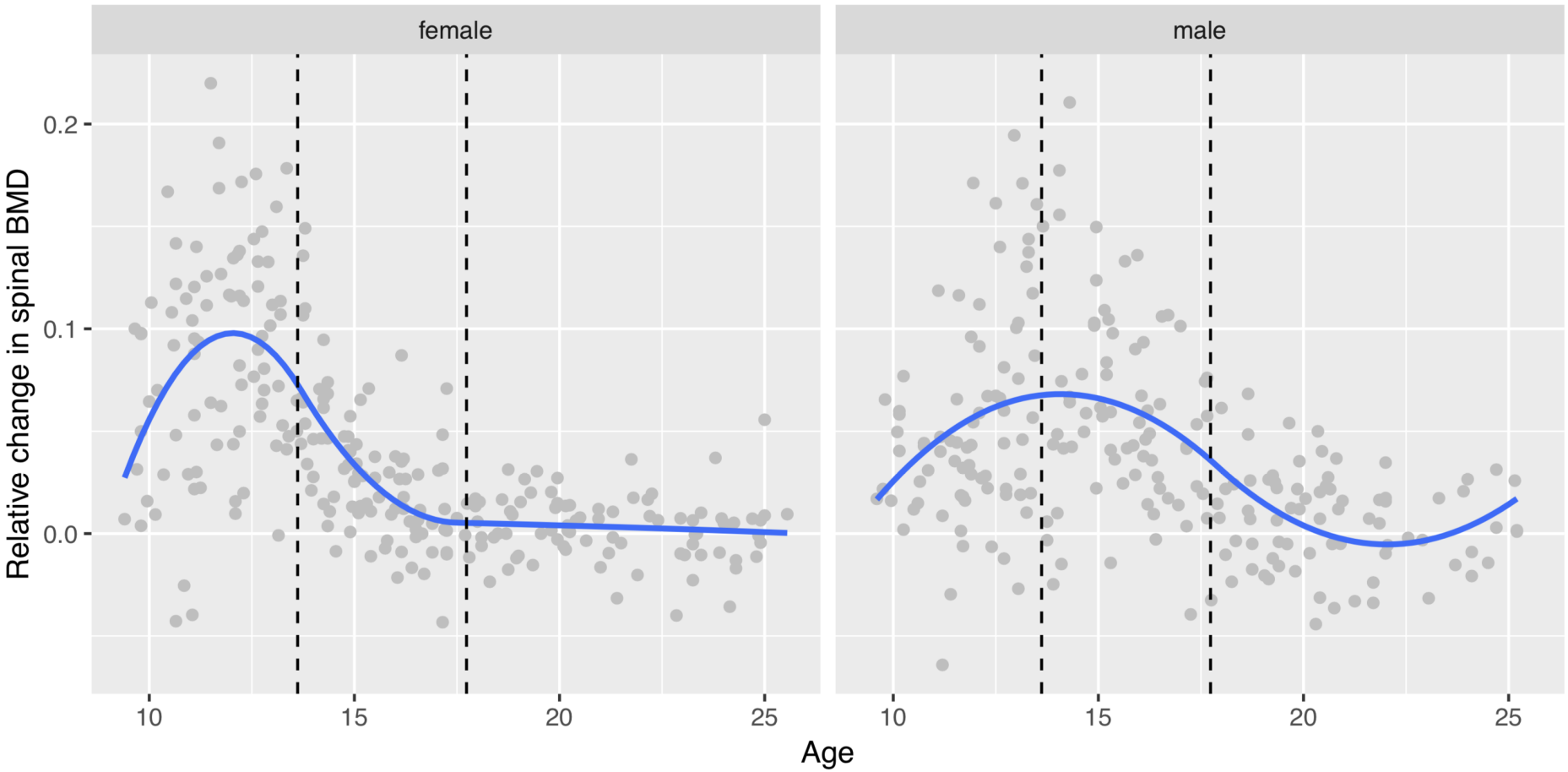
$$h_1(x) = 1, \quad h_2(x) = x, \quad h_3(x) = (x - \xi_1)_+, \quad h_4(x) = (x - \xi_2)_+,$$

  where $(\cdot)_+$ denotes the positive portion of its argument:

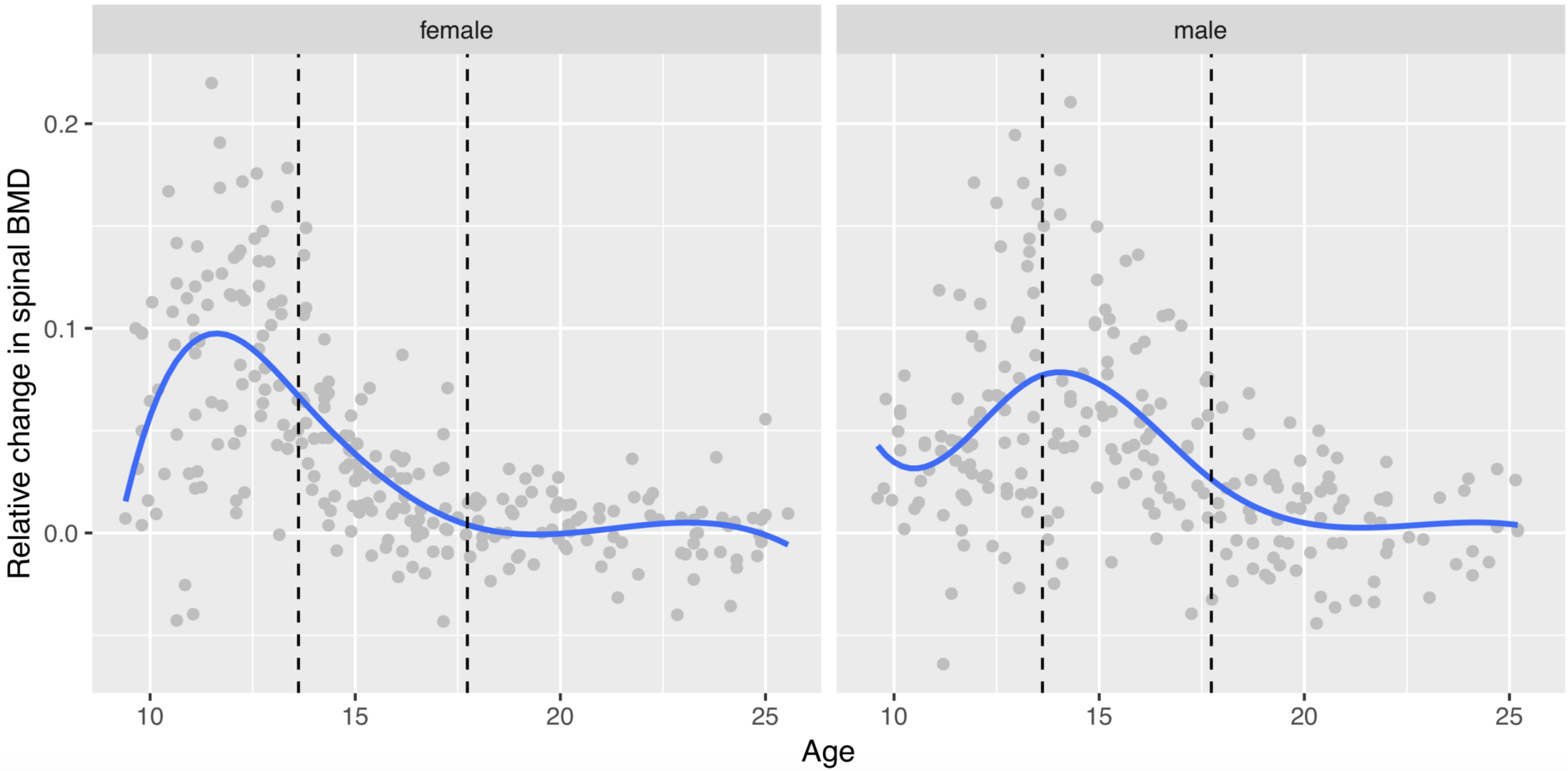
$$r_+ = \begin{cases} r & \text{if } r \geq 0 \\ 0 & \text{if } r < 0 \end{cases}$$

- Note that the degrees of freedom add up: 3 regions $\times$ 2 df/region $-$ 2 constraints $=$ 4 basis functions
- This set of basis functions is an example of what is called the *truncated power basis*; it can be extended to any order of polynomials

Quadratic splines



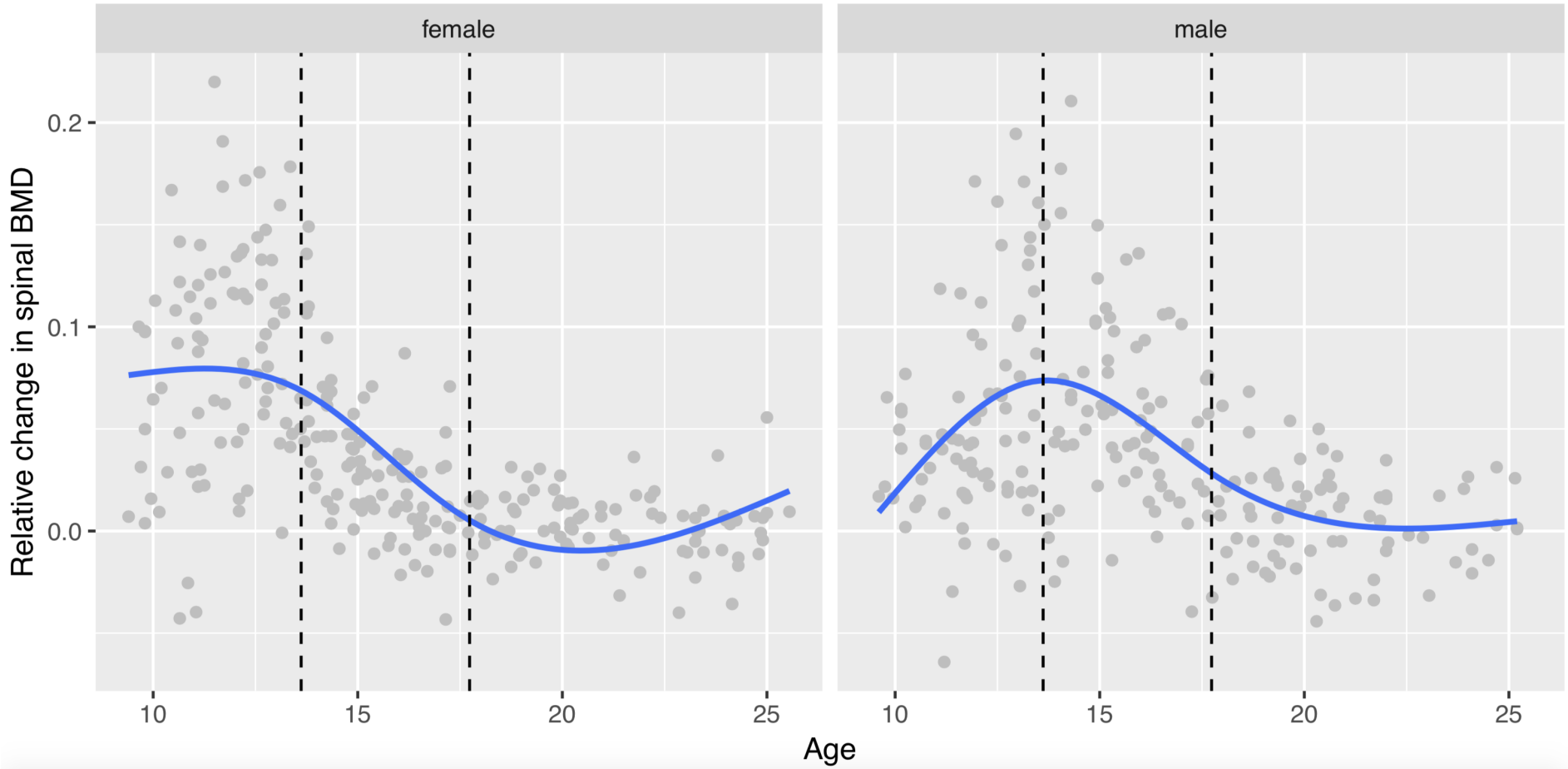
Cubic splines



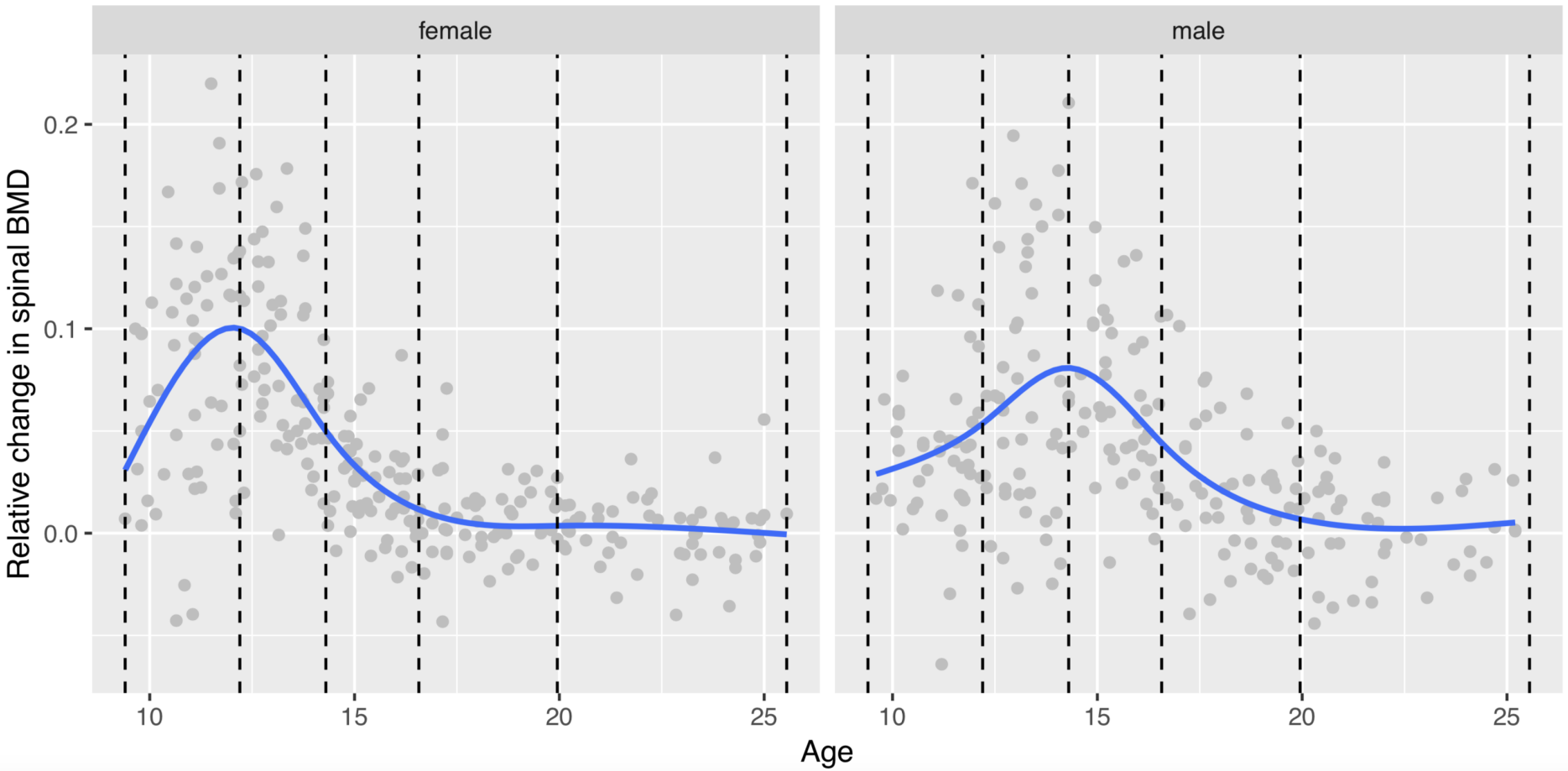
Natural cubic splines

- Polynomial fits tend to be erratic at the boundaries of the data; naturally, cubic splines share the same flaw
- Natural cubic splines ameliorate this problem by adding the additional (4) constraints that the function is linear beyond the boundaries of the data

Natural cubic splines



Natural cubic splines, 6df



Additive models

- When we have multiple features, a natural extension of basis functions is to assume an additive relationship:

$$f(x) = \sum_{j=1}^{p} f_j(x_j)$$

  such models are called *additive models* or *generalized additive models* (GAMs)
- Here $f(x)$ may be estimated via some scatterplot smoother $\mathcal{S}$, such as regression splines $\hat{f}_j(x_j) = \beta_{mj} h_{mj}(x_j)$
- We are considering

$$y = f(x_1, \ldots, x_p) + \varepsilon = \alpha + \sum_{j=1}^{p} f_j(x_j) + \varepsilon$$

  where the p functions $f_j(x_j)$ are estimated by the *backfitting* algorithm (which is, in fact, a *coordinate descent* algorithm) based of a method for smoothing $\mathcal{S}$.

Additive models: estimation algorithm

- In order to avoid an overparametrisation of the intercept, each f_j needs to be of 'zero mean'.
- We may write

$$f_j(x_j) = y - \left(\alpha + \sum_{k \neq j} f_k(x_k) + \varepsilon \right)$$

- *Backfitting* algorithm
  - Initialisation: $\hat{\alpha} = \sum_i y_i / n$, $\hat{f}_j = 0, \forall j$
  - Cycle: for $j = 1, 2, \ldots, p, 1, 2, \ldots, p, 1, 2, \ldots,$

$$\begin{aligned} \hat{f}_j &\leftarrow \mathcal{S}\left[\left\{ y_i - \hat{\alpha} - \textstyle\sum_{k \neq j} \hat{f}_k(x_{ik}) \right\}_1^n \right], \\ \hat{f}_j &\leftarrow \hat{f}_j - n^{-1} \textstyle\sum_{i=1}^n \hat{f}_j(x_{ij}) \end{aligned}$$

  until the functions $\hat{f}_j$ are stable.

Sparse Additive Models

- Representing the problem as a group lasso model, we have

$$\min_{f_j \in \mathcal{F}_j, j=1,\ldots,p} \left[\mathbb{E}\left\{ \left(Y - \sum_{j=1}^{p} f_j(x_j) \right)^2 \right\} + \lambda \sum_{j=1}^{p} \|f_j\|_2 \right]$$

  where the $\mathcal{F}_j$ are a fixed set of univariate function classes and $\|f_j\|_2 = \sqrt{\mathbb{E}\{f_j^2(x_j)\}}$ is the $L_2(P_j)$ norm applied to component j.

- This idea was originally proposed by Ravikumar et al. (2009), who named it sparse additive models (SPAM)

Sparse Additive Models (SPAM)

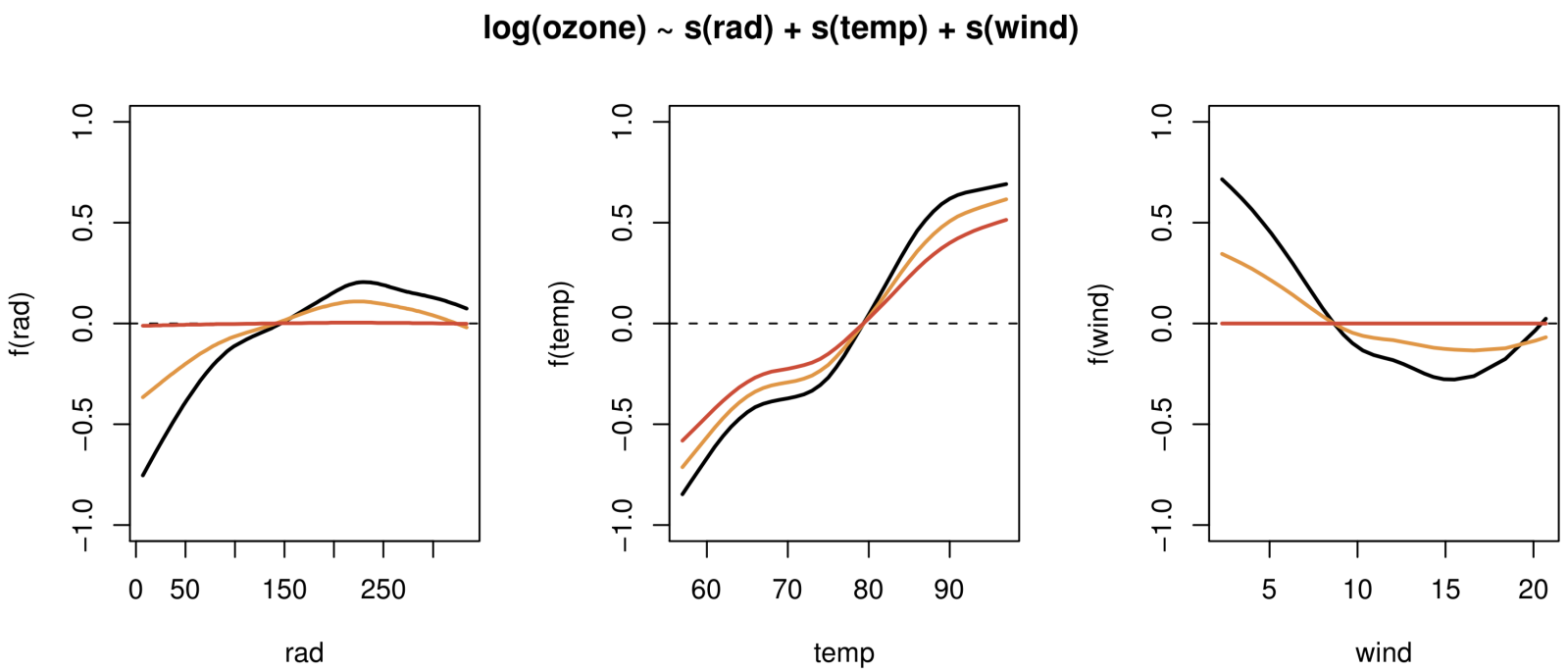
- Data driven sparse backfitting equations are

$$\begin{aligned} \tilde{f}_j &= \mathcal{S}_j\left(y - \sum_{k\neq j} \hat{f}_k(x_k)\right), \\ \hat{f}_j &= \left(1 - \frac{\lambda}{\|\tilde{f}\|_2}\right)_+ \tilde{f}_j \end{aligned}$$

for $j = 1, \ldots, p, 1, \ldots, p, \ldots$, iterating until convergence

SPAM

- Air pollution on 111 days from May to September 1973 in New York
- response: `log(ozone concentration)`,
- predictors: `radiation`, `temperature`, and `wind speed`.
- Smoothing splines were used in the additive model fits, each with fixed equivalent degree of freedom $df = 5$
- black: $\lambda = 0$; orange: $\lambda = 2$; red: $\lambda = 4$
- We see that while the shrinkage leaves the functions of `temp` relatively untouched, it has a more dramatic effect on `rad` and `wind`.
- Note that when $\lambda = 4$ the entire curves $f(\texttt{rad})$ and $f(\texttt{wind})$ are 0.



Connection with group lasso

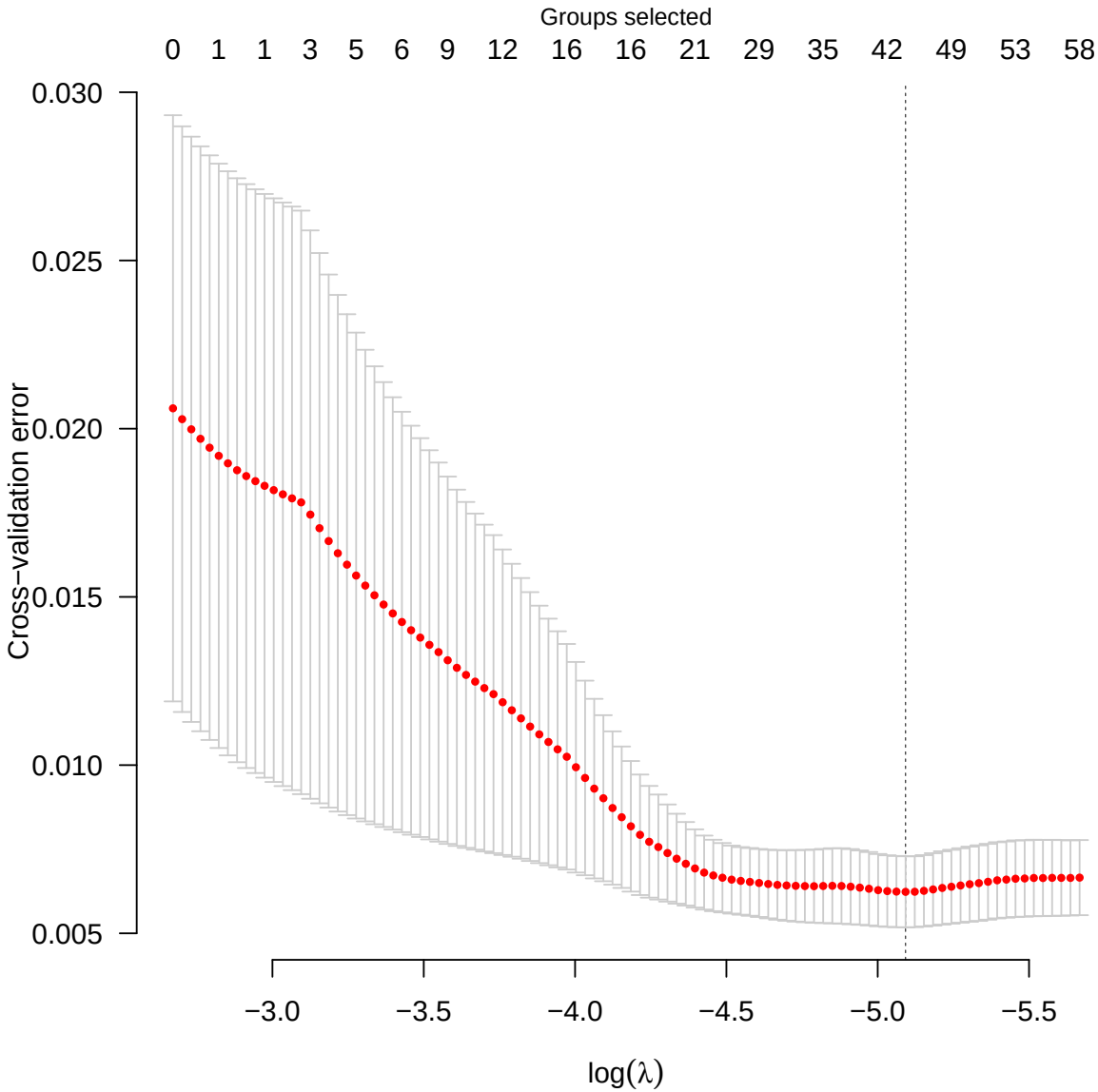
- If the smoothing method for variable x_j is a projection on to a set of basis functions $f_j(\cdot) = \sum_{m=1}^{p_j} h_{mj}(\cdot)\beta_{mj}$, such as cubic splines.
- Here p_j is the number of basis functions for variable j
- In this case $\|f_j\|_2 = \sum_{m=1}^{p_j} \beta_{mj}^2$
- Can be shown that the SPAM optimization is equivalent of a group lasso with predictors $h_{mj}(\cdot)$ and a corresponding block vector of coefficients β_{mj}
- Representing the problem as a group lasso model, we have

$$\min_{\beta_j \in \mathbb{R}^{p_j}, j=1,\ldots,p} \left\{ \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \sum_{m=1}^{p_j} h_{mj}(x_{ij})\beta_{mj} \right)^2 + \lambda \sum_{j=1}^{p} \sum_{m=1}^{p_j} \beta_{mj}^2 \right\}$$

Rat eye data

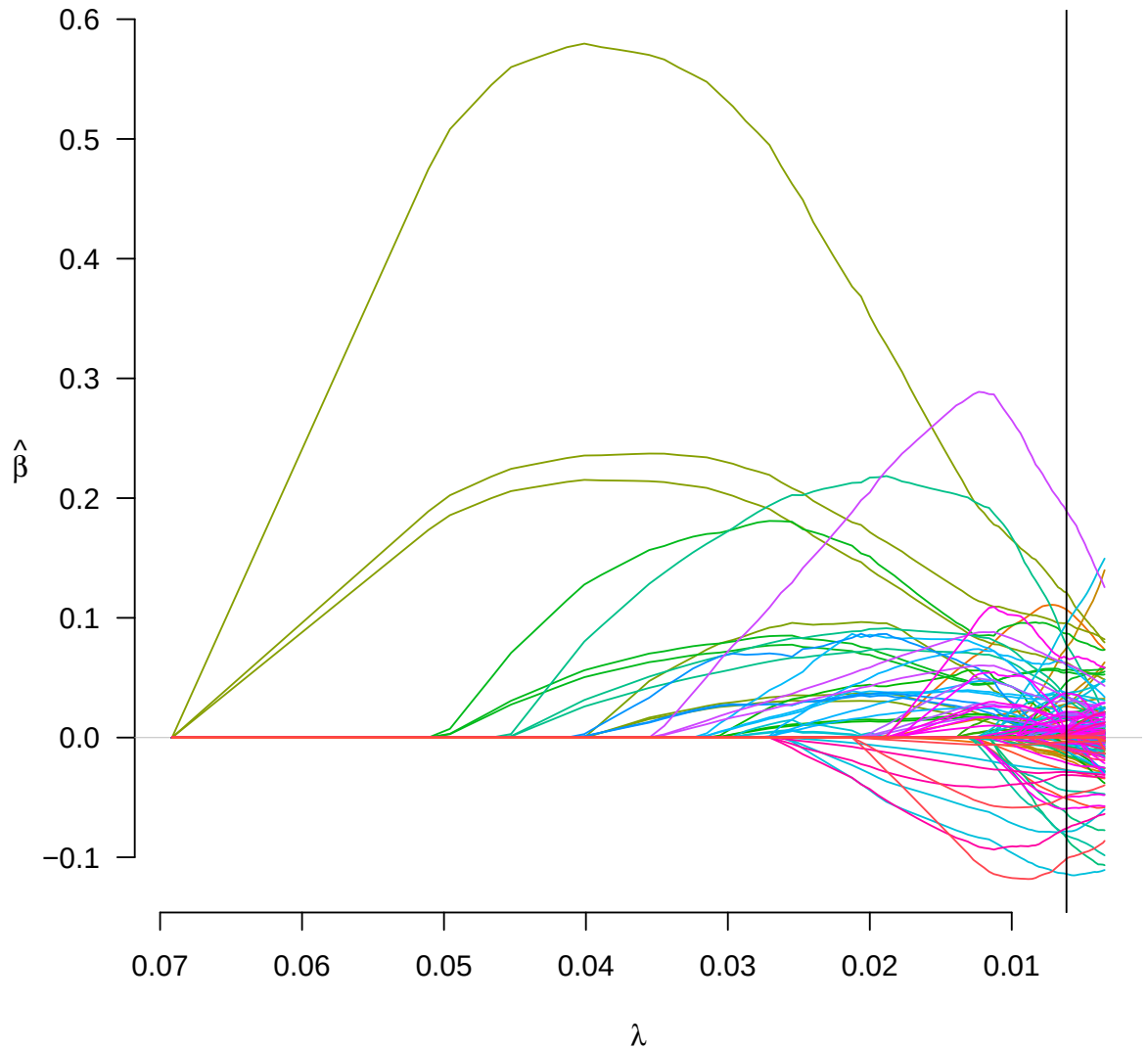
- Study of gene expression data, this time gathered from the eye tissue of 120 twelve-week-old male rats
- The goal of the study was to detect genes whose expression patterns are related to that of the gene TRIM32, a gene known to be linked to a genetic disorder called Bardet-Biedl Syndrome (which, among other symptoms, leads to a number of problems with vision and proper formation of the retina)
- The analysis was restricted to the 857 genes on chromosome 5
- First task is setting up the splines, each with 3 degrees of freedom
- Setting up the splines produces, 3×857 coefficients for each rat ($n = 120$)
- *Groups* are the three coefficients for each gene

Rat eye data

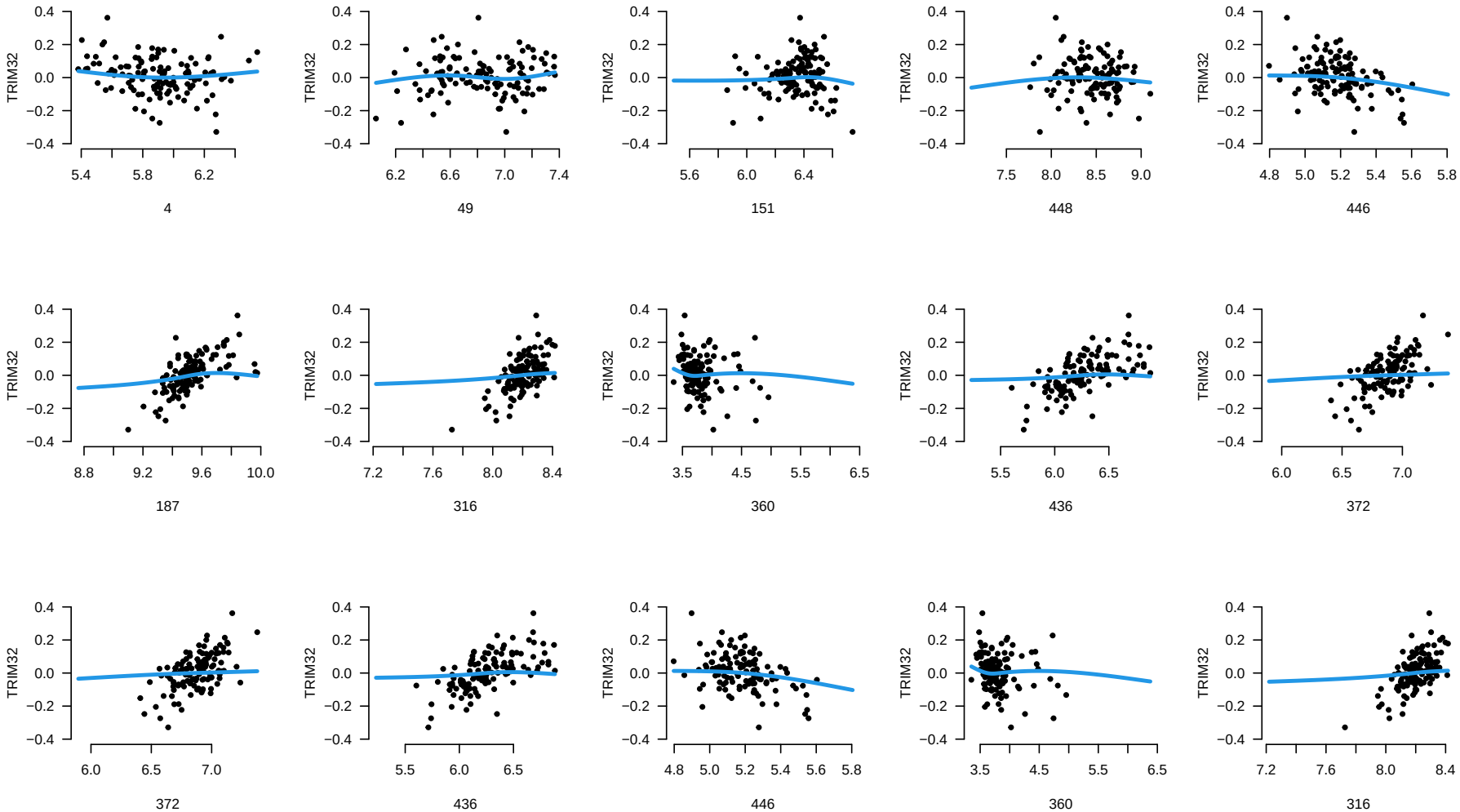

Groups selected
0 1 1 3 5 6 9 12 16 16 21 29 35 42 49 53 58
0.030
0.025
0.020
0.015
0.010
0.005
Cross−validation error
−3.0
−3.5
−4.0
−4.5
−5.0
−5.5
log(λ)

Rat eye data

The group lasso model selects 49 genes, achieving an R^2 of 0.72



Effects


TRIM32
0.4
0.2
0.0
−0.2
−0.4
5.4 5.8 6.2
4
6.2 6.6 7.0 7.4
49
5.6 6.0 6.4
151
7.5 8.0 8.5 9.0
448
4.8 5.0 5.2 5.4 5.6 5.8
446
8.8 9.2 9.6 10.0
187
7.2 7.6 8.0 8.4
316
3.5 4.5 5.5 6.5
360
5.5 6.0 6.5
436
6.0 6.5 7.0
372
6.0 6.5 7.0
372
5.5 6.0 6.5
436
4.8 5.0 5.2 5.4 5.6 5.8
446
3.5 4.5 5.5 6.5
360
7.2 7.6 8.0 8.4
316

Spike and slab prior – Horseshoe prior

Variable selection

- We focus on selecting sets of variables at a time using a model-based approach.
- We consider each variable in the presence of the rest.
- If the model is a generalized linear model, of the form

$$p(y|x) = p(y|x = f(\beta^\top x))$$

for some link function f, then we can perform feature selection by encouraging the weight vector to be sparse (have lots of zeros).

Linear model

- Consider the linear model

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

or

$$y_i \sim \mathcal{N}\left(\beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j, \sigma^2\right)$$

- For simplicity, let us, now, call $\beta = [\beta_0, \beta_1, \ldots, \beta_p]^\top$ the vector with all the $p+1$ parameters
- Let call $p(y|x, \beta, \sigma^2) = L(\beta, \sigma^2; y, x)$ the likelihood function for the parameters, where we stress the probability interpretation of the likelihood

$$p(y|x, \beta, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2\right\}$$

Prior information

- Sometimes we have some prior information about the parameters
- Clearly if those hints are 'deterministic' (such as: β_7 must be positive, or β_{23} can not be included in the set $\{3, 7\}$) we can easily include constraints in the model, and modify the likelihood
- However, we may have some 'probabilistic' hint about some parameter, reflecting our uncertainty about our prior information. For example we may know, a priori, that β_3 is most likely around 10 and that its standard deviation can be around 2 (which may be 'translated', for example, in saying that it follows a Gaussian distribution with mean 10 and standard deviation 2). Or we can have some ideas about quantiles of the distribution, etc.
- These beliefs can derive from past experience or from our (or the *client*'s) subjective opinion (or both).
- And what we are interested in our analysis is to modify our opinion on the basis of the observed new data
- Now we are considering the parameters $\beta_j, j = 0, \ldots, p$ as random variables, and not as fixed unknown values.

Prior distribution

- Assume, for now, that variance σ^2 is known.
- Suppose we are able to elicit out prior information about all the parameters through a joint probability distribution $p(\beta_0, \beta; \theta)$ depending on a vector of parameters θ usually called hyperparameters.
- The distribution of parameters before observing data is called *prior distribution* or *a priori*.
- Typically we also suppose that priors for each parameter are independent each other, i.e.
  $p(\beta_0, \beta; \theta) = p(\beta_0; \theta) \cdot p(\beta_1; \theta) \cdot p(\beta_2; \theta) \cdot \dots \cdot p(\beta_p; \theta)$
- For example we may think that

$$\beta_j \sim \mathcal{N}(\mu_j, \tau_j^2) \qquad \text{for } j = 0, \dots, p$$

i.e.,

$$p(\beta_j; \mu_j, \tau_j) = \frac{1}{\sqrt{2\pi\tau^2}} \exp\left\{-\frac{1}{2\tau}(\beta_j - \mu_j)^2\right\}$$

where θ is the set of all μ_j and τ_j

Posterior distribution

- Now β_j, $j = 0, 1, \ldots, p$ describe our opinion, our 'belief', about the parameters.
- There is not anymore a true and unknown value of the parameters that we are looking for.
  We are not interested in the truth (there exists a 'truth'?), but only on our opinion, supported and modified by the data, about it. And our opinion is tipically uncertain.
- How can we update our opinion? By using the Bayes theorem

$$p(\beta|y, x, \theta) \propto p(y|x, \beta, \sigma^2) \cdot p(\beta|\theta)$$

- The distribution of the parameters given data is called *posterior distribution* or *a posteriori*
- This approach to inference (parameters are random variables/opinions, subjective interpretation of probability, etc.) and some consequences of it (rejecting the principle of repeated sampling, accepting the 'strong principle of likelihood', etc.) is called Bayesian approach to inference

Mode of the posterior distribution

- Now $\beta_j, j = 0, \ldots, p$ are random variables and 'estimates' are entire distributions, not single values.
- If we are interested in one single quantity, we may use posterior mean or median
- However, the posterior mode (often called MAP estimate), is the most popular choice because it reduces to an optimization problem, for which efficient algorithms often exist.

Mode of the posterior distribution

- Computation of the posterior distribution may be (and often is) tricky, and if we have a high number of parameters no explicit solution is available and computational methods, mainly based on simulations need to be implemented (Markov Chain Monte Carlo, MCMC: Gibbs sampling algorithm, Metropolis-Hastings algorithm, Hamiltonian Monte Carlo; there are also some attempt to approximate distribution in a more efficient way, such as Variational Bayes - fashion now, but not so successful -, etc.).
- This is the main bottleneck in fitting Bayesian models in big data setting.
- For simple cases, we have explicit solutions, so called conjugate distributions (which are the basis of Gibbs sampling algorithm).

Gaussian likelihood - σ^2 known

- Consider σ^2 is known, likelihood Gaussian and priors for β_j's all Gaussian:

$$\beta \sim \mathcal{N}_{p+1}(\mu_0, V_0)$$

  here μ_0 is the vector of prior means and V_0 the prior covariance matrix.

- The posterior is also Gaussian

$$\beta | y, x \sim \mathcal{N}_{p+1}(\mu_n, V_n)$$

  with

$$\begin{aligned} \mu_n &= \left(V_0^{-1} + \frac{1}{\sigma^2} X^\top X \right)^{-1} \left(V_0^{-1} \mu_0 + \frac{1}{\sigma^2} X^\top y \right) \\ V_n &= \left(V_0^{-1} + \frac{1}{\sigma^2} X^\top X \right)^{-1} \end{aligned}$$

- Posterior updates the prior with the data based estimate

σ^2 known - back to penalization

- If $\mu = 0$ and $V_0 = \tau^2 I$ then the posterior mean reduces to the ridge estimate, with $\lambda = \frac{\sigma^2}{\tau^2}$
- In fact, the corresponding MAP estimation problem is equivalent to minimize

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\|\beta\|_2^2$$

(where here β only include variables coefficients)

- If prior instead of Gaussian is a Laplace distribution, i.e.,

$$p(\beta|\lambda) \propto \prod_{j=1}^{p} \exp\{-\lambda|\beta_j|\}$$

the negative log posterior density for $\beta|y, \lambda, \sigma^2$ is given by the lasso

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\|\beta\|_1$$

Gaussian likelihood - σ^2 unknown

- In the general case when σ^2 is unknown, the results are similar although more complex
- Likelihood is still Gaussian.
- Joint prior for all β_j's and σ^2 is 'Normal-InverseGamma':

$$\beta, \sigma^2 \sim \mathcal{N}_{p+1}(\mu_0, \sigma^2 V_0)\mathcal{IG}(a, b)$$

where $\mathcal{IG}(a, b)$ indicate the 'Inverse gamma' distribution which density is

$$p(x) = \frac{b^a}{\Gamma(a)} x^{-(a+1)} \exp\left\{-\frac{b}{x}\right\} \qquad \text{for } x > 0, a > 0 \text{ and } b > 0.$$

- If a random variable is distributed as a *gamma* its inverse is *inverse gamma*
- Therefore the joint prior is

$$p(\beta, \sigma^2; \mu_0, V_0, a, b) = \frac{b^a}{(2\pi\tau^2)^{p/2}|V_0|^{\frac{1}{2}}\Gamma(a)} (\sigma^2)^{-\left(a+\frac{p}{2}+1\right)} \exp\left\{-\frac{1}{2\sigma^2}(\beta-\mu_0)^\top V_0^{-1}(\beta-\mu_0) + 2b\right\}$$

Posterior distribution - σ^2 unknown

- With this prior and likelihood, the posterior has the form

$$\beta, \sigma^2 | y, x \sim \mathcal{N}_{p+1}(\mu_n, \sigma^2 V_n) \mathcal{IG}(a_n, b_n)$$

with

$$\begin{aligned}
\mu_n &= \left(V_0^{-1} + X^\top X\right)^{-1} \left(V_0^{-1}\mu_0 + X^\top y\right) \\
a_n &= a + \frac{n}{2} \\
b_n &= b + \frac{1}{2}\left(\mu_0^\top V_0^{-1}\mu_0 + y^\top y - \mu_n^\top V_n^{-1}\mu_n\right) \\
V_n &= \left(V_0^{-1} + X^\top X\right)^{-1}
\end{aligned}$$

- a_n updates the counts, b_n is the prior sum of squares, b, plus the empirical sum of squares, $y^\top y$, plus a term due to the error in the prior on β.

Bayesian variable selection

- A natural way to pose the variable selection problem is by introducing a new set of variables γ_j, $j = 1, \ldots, p$

$$\gamma_j = \begin{cases} 1, & \text{if feature } j \text{ is relevant} \\ 0, & \text{otherwise} \end{cases}$$

- Our goal is to compute the posterior over models

$$p(\gamma|y) \propto p(y|\gamma) \cdot p(\gamma) = \exp\{-f(\gamma)\}$$

where $f(\gamma)$ is a cost function

$$f(\gamma) = -\{\log p(y|\gamma) + \log p(\gamma)\}$$

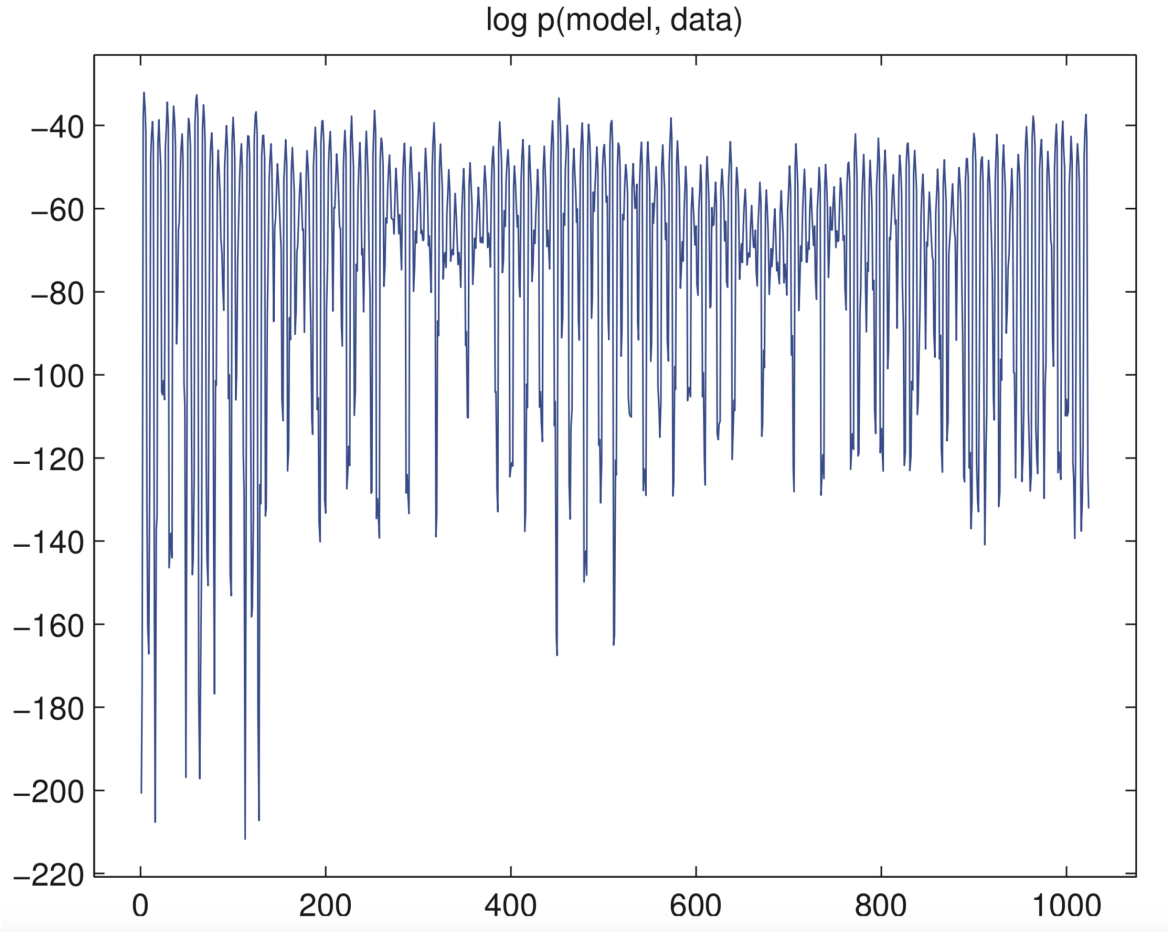
- Posterior is, therefore,

$$p(\gamma|y) = \frac{\exp\{-f(\gamma)\}}{\sum_{\gamma'} \exp\{-f(\gamma')\}}$$

Bayesian variable selection

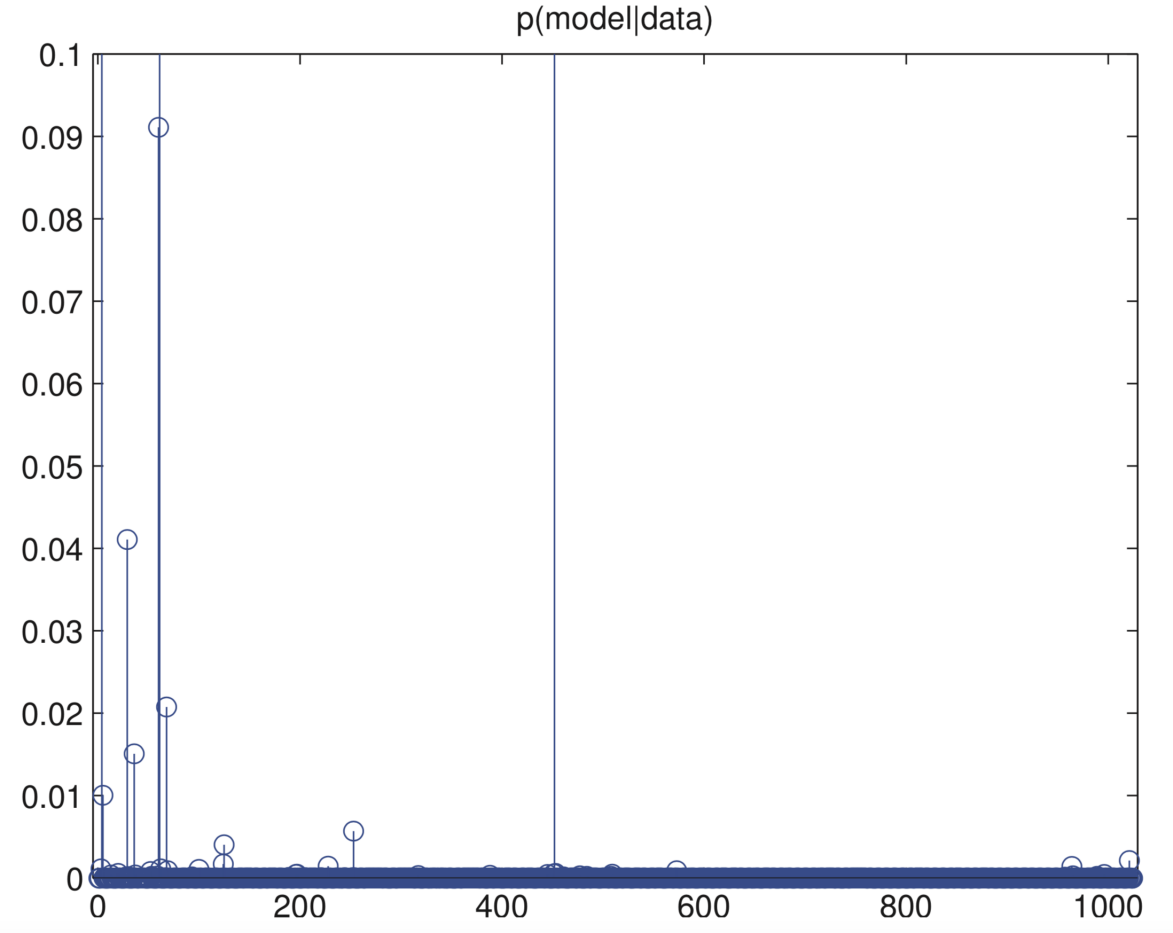
- For example, suppose we generate $n = 20$ samples from $p = 10$ dimensional linear regression model, $y_i \sim \mathcal{N}(\beta^\top x_i, \sigma^2)$, in which $K = 5$ elements are non-zero.
- In particular, we use $\beta = [0.00, -1.67, 0.13, 0.00, 0.00, 1.19, 0.00, -0.04, 0.33, 0.00]^\top$ and $\sigma^2 = 1$
- We enumerate all $2^p = 1024$ models and compute $p(\gamma|y)$ for each one.

The cost of each model $f(\gamma) = \log p(model, data)$ is extremely bumpy.



$p(\gamma|y)$

The results are easier to interpret if we compute the posterior distribution over models, $p(\gamma|y)$.



Posterior over all 1024 models. Vertical scale has been truncated at 0.1 for clarity.

Bayesian variable selection

The top 8 models

model	prob	members
4	0.447	2
61	0.241	2, 6
452	0.103	2, 6, 9
60	0.091	2, 3, 6
29	0.041	2, 5
68	0.021	2, 6, 7
36	0.015	2, 5, 6
5	0.010	2, 3

The "true" model is $\{2, 3, 6, 8, 9\}$.

Bayesian variable selection

- True coefficients associated with features 3 and 8 (0.13 and -0.04 are extremely small compared to $\sigma^2 = 1$, so these variables are harder to detect
- Given enough data, the method will converge to the true model (assuming the data is generated from a linear model), but for finite data sets, there will usually be considerable posterior uncertainty.
- Interpreting the posterior over a large number of models is quite difficult, so we will seek various summary statistics. A natural one is the posterior mode, or MAP estimate

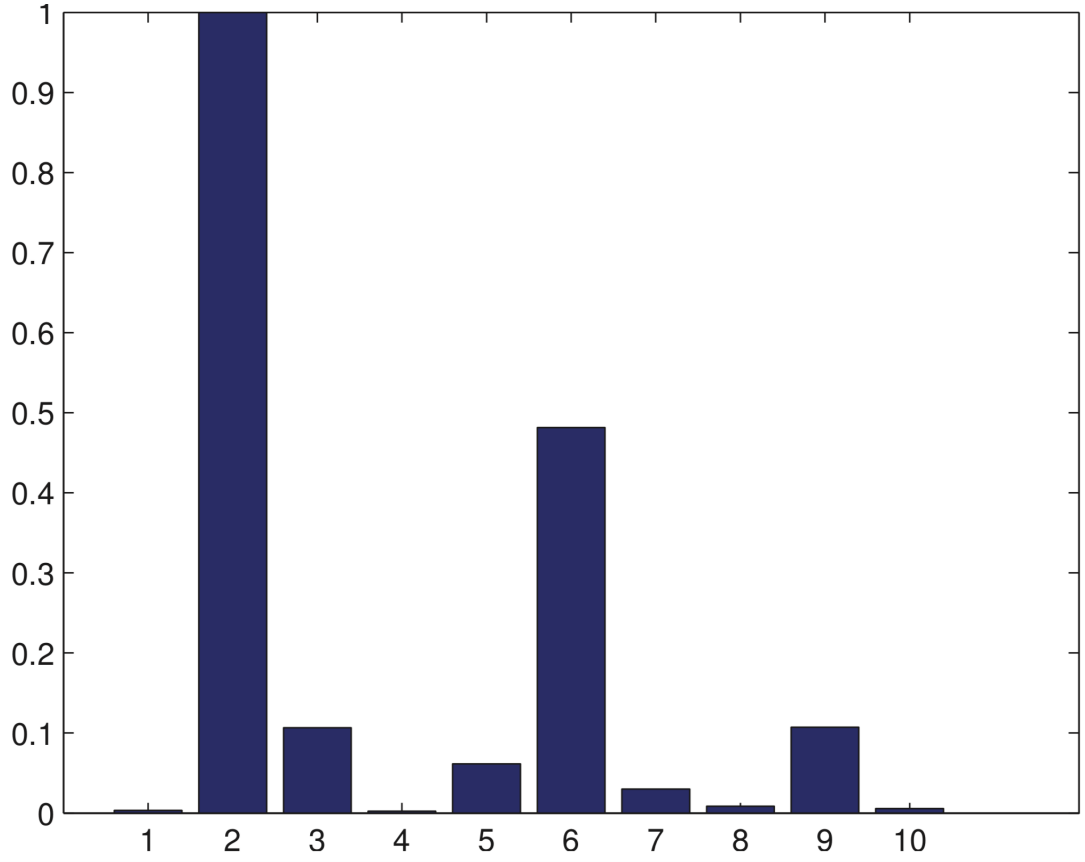
$$\hat{\gamma} = \operatorname{argmax} p(\gamma|y) = \operatorname{argmin} f(\gamma)$$

- A better summary is the median model

$$\hat{\gamma} = \{j : p(\gamma_j = 1|y) > 0.5\}$$

- This requires computing the posterior marginal inclusion probabilities $p(\gamma_j = 1|y)$

Marginal inclusion probabilities



The model is confident that variables 2 and 6 are included
If we lower the decision threshold to 0.1, we would add 3 and 9 as well.
However, if we wanted to 'capture' variable 8, we would incur two false positives 5 and 7.

Comments of the example

- The problem was sufficiently small (only 10 variables) that we were able to compute the full posterior exactly.
- Of course, variable selection is most useful in the cases where the number of dimensions is large.
- Since there are 2^p possible models (bit vectors), it will be impossible to compute the full posterior in general, and even finding summaries, such as the MAP estimate or marginal inclusion probabilities will be intractable.

Spike and slab prior

- The posterior is given by

$$p(\gamma|y) \propto p(\gamma) \cdot p(y|\gamma)$$

- It is common to use the following prior on the bit vector

$$p(\gamma) = \prod_{j=1}^{p} Ber(\gamma_j|\pi_0) = \pi_0^{\|\gamma\|_0}(1-\pi_0)^{p-\|\gamma\|_0}$$

where π_0 is the probability a feature is relevant, and $\|\gamma\|_0 = \sum_{j=1}^{p} \gamma_j$ is the L_0 pseudo-norm that is, the number of non-zero elements of the vector.

- It is useful to write the log prior as

$$\begin{aligned} \log p(\gamma|\pi_0) &= \|\gamma\|_0 \log \pi_0 + (p - \|\gamma\|_0) \log(1-\pi_0) \\ &= \|\gamma\|_0 (\log \pi_0 - \log(\pi_0)) + \text{const} \\ &= -\lambda \|\gamma\|_0 + \text{const} \end{aligned}$$

where $\lambda = \log \frac{1-\pi_0}{\pi_0}$ controls the sparsity of the model.

Spike and slab prior

- We can write the likelihood as follows

$$p(y|\gamma) = \int \int p(y|x, \beta, \gamma) p(\beta|\gamma, \sigma^2) p(\sigma^2) d\beta d\sigma^2$$

- For notational simplicity, we assume the response is centered ($\bar{y} = 0$), so we can ignore the intercept β_0.
- Regarding the prior $p(\beta|\gamma, \sigma^2)$
  - If $\gamma = 0$, feature j is irrelevant, so we expect $\beta_j = 0$.
  - If $\gamma = 1$, we expect β_j to be nonzero.
    If we standardize the inputs, a reasonable prior for $\beta|\gamma, \sigma^2$ is $\mathcal{N}(0, \sigma^2 \Sigma_\gamma)$, where Σ_γ controls how big we expect the coefficients associated with the relevant variables to be (which is scaled by overall noise level σ^2).
  - It is common to use a g-prior of the form $\Sigma_\gamma = g \cdot (X_\gamma^\top X_\gamma)^{-1}$, where X_γ is the matrix of variables for which $\gamma = 1$.
    Various approaches have been proposed for setting g, including cross validation, empirical Bayes, hierarchical Bayes, etc.

Spike and slab prior

- We can summarize this prior

$$p(\beta_j|\sigma^2,\gamma_j) = \begin{cases} \delta_0(\beta_j) & \text{if } \gamma_j = 0 \\ (2\pi)^{-\frac{k}{2}} \frac{1}{\sigma\sqrt{\det(\Sigma_\gamma)}} \exp\left\{-\frac{1}{2}\sigma^{-2}\beta_\gamma^\top \Sigma_\gamma^{-1}\beta_\gamma\right\} & \text{if } \gamma_j = 1 \end{cases}$$

- The first term is a 'spike' at the origin
- As each elements of the diagonal of Σ_γ goes to infinity, $\text{diag}(\Sigma_\gamma)_j \to \infty$, the distribution $p(\beta_j|\gamma_j = 1)$ approaches a Uniform ('slab')

Spike and slab prior

- We can obtain the marginal likelihood, i.e., integrating out β_j.
- If σ^2 is known, we obtain

$$y|\gamma, \sigma^2 \sim \mathcal{N}(0, C_\gamma)$$
$$C_\gamma = \sigma^2 X_\gamma \Sigma_\gamma X_\gamma^\top + \sigma^2 I_n$$

- If σ^2 is unknown, we put a prior, typically $\sigma^2 \sim \mathcal{IG}(a, b)$ on it and integrate it out. We obtain

$$\begin{aligned} p(y|\gamma) &= \int\int p(y|\gamma, \beta, \sigma^2) p(\beta|\gamma, \sigma^2) p(\sigma^2) d\beta d\sigma^2 \\ &\propto |X_\gamma^\top X_\gamma + \Sigma_\gamma^{-1}|^{-\frac{1}{2}} |\Sigma_\gamma|^{-\frac{1}{2}} (2b + S(\gamma))^{-\frac{1}{2}(2a+n-1)} \end{aligned}$$

  here $S(\gamma)$ is the residual sum of squares

$$S(\gamma) = y^\top y - y^\top X_\gamma \left(X_\gamma^\top X_\gamma + \Sigma_\gamma^{-1} \right)^{-1} X_\gamma^\top y$$

- Marginal posterior does not depend on β anymore, meaning that in doing inference on y we are interested in the values of β but only if they are in the model or not (only on γ)

Bernoulli-Gaussian Model

- Another model used is

$$\begin{aligned} y_i|x_i, \beta, \gamma, \sigma^2 &\sim \mathcal{N}\left(\sum_{j=1}^{p} \gamma_j \beta_j x_{ij}, \sigma^2\right) \\ \gamma_j &\sim Ber(\pi_0) \\ \beta_j &\sim \mathcal{N}(0, \tau^2) \end{aligned}$$

- This is called the Bernoulli-Gaussian model. We could also call it the binary mask model, since the γ variables are "masking out" β.
- Unlike the spike and slab model, we do not integrate out the "irrelevant" coefficients.
- In addition, the binary mask model has the form $\gamma_j \to y \leftarrow \beta_j$ whereas the spike and slab model has the form $\gamma_j \to \beta_j \to y$
- In binary mask model, only $\gamma_j \beta_j$ can be identified from the likelihood.

Bernoulli-Gaussian Model

- This model can be used to derive an objective function that is widely used in the (non-Bayesian) subset selection literature.
- The joint prior has the form

$$\gamma, \beta \sim \mathcal{N}(0, \tau^2 I) \pi_0^{\|\gamma\|_0} (1 - \pi_0)^{p - \|\gamma\|_0}$$

- Hence the scaled unnormalized negative log posterior has the form

$$\begin{aligned} f(\gamma, \beta) &= -2\sigma^2 \log p(\gamma, \beta, y | X) \\ &= \|y - X(\gamma \circ \beta)\|^2 + \frac{\sigma^2}{\tau^2} \|\beta\|^2 + \lambda \|\gamma\|_0 + \text{const} \end{aligned}$$

where $A \circ B$ denote the Hadamard product, or element-wise product, and

$$\lambda = 2\sigma^2 \log \frac{\pi_0}{\pi_0}$$

Bernoulli Gaussian Model

- Split β into $\beta{-}\gamma$ and β_γ indexed by the zero & non-zero entries of γ.
- Now consider $\sigma^2 \to \infty$, so we do not regularize the non-zero weights (no complexity coming from the marginal likelihood).
- In this case, $f(\gamma, \beta) = \|y - X(\gamma \circ \beta)\|^2 + \frac{\sigma^2}{\tau^2}\|\beta\|^2 + \lambda\|\gamma\|_0 + \text{const}$ becomes

$$f(\gamma, \beta) = \|y - X_\gamma \beta_\gamma\|^2 + \lambda\|\gamma\|_0 + \text{const}$$

- Instead of keeping track of the bit vector γ, we can define the set of relevant variables to be the support, or set of non-zero entries, of β.
- Then we can rewrite the above equation as

$$f(\beta) = \|y - X\beta\|_2^2 + \lambda\|\beta\|_0$$

- This is called L_0 regularization. The discrete optimization problem over $\gamma \in \{0, 1\}^p$ is now transformed to a continuous one over $\beta \in \mathbb{R}^p$

Horseshoe prior

- To implement Bayesian sparsity we need a prior distribution that allows the data to collapse the entire marginal posterior for each slope towards relevance or irrelevance, but not both.
- The ideal prior distribution for β_j will put a probability mass on zero to reduce variance, and have fat tails to reduce bias.
- Both L_1 and L_2 penalization fail the test, i.e., both doble-exponential and normal distributions have thin tails, and the probability mass they put at 0 is 0.
- Note that in order to put the probability mass > 0 the probability density function must diverge.

Horseshoe prior

- A possible solution, accomplishes this flexibility by setting the scale for each component to the product of a global scale, τ, and a local scale, λ_m, each of which are themselves unknown parameters.

$$
\begin{aligned}
y_i|x_i, \beta, \gamma, \sigma^2 &\sim \mathcal{N}\left(\sum_{j=1}^{p} \beta_j x_{ij}, \sigma^2\right) \\
\beta_j &\sim \mathcal{N}(0, \tau^2 \cdot \lambda_j^2) \\
\tau &\sim \text{half-}\mathcal{C}(0, 1) \\
\lambda_j &\sim \text{half-}\mathcal{C}(0, \tau)
\end{aligned}
$$

  where half-$\mathcal{C}(0, 1)$ is the half-Cauchy distribution with location parameter 0 and scale parameter 1 (which is a Student-t distribution with 1 df).
- In this approach it is proposed to use a half-Cauchy distribution as a prior distribution. It puts non-zero probability mass at 0, and also has a fat tail.

Horseshoe prior

- The heavy-tailed Cauchy prior distribution for the local scales allows the data to push each to large values as needed, which then push the corresponding slopes above the global scale, τ.
- By making the global scale itself a parameter we also allow the data to refine the scale beyond our prior judgement of τ_0.

Horseshoe prior

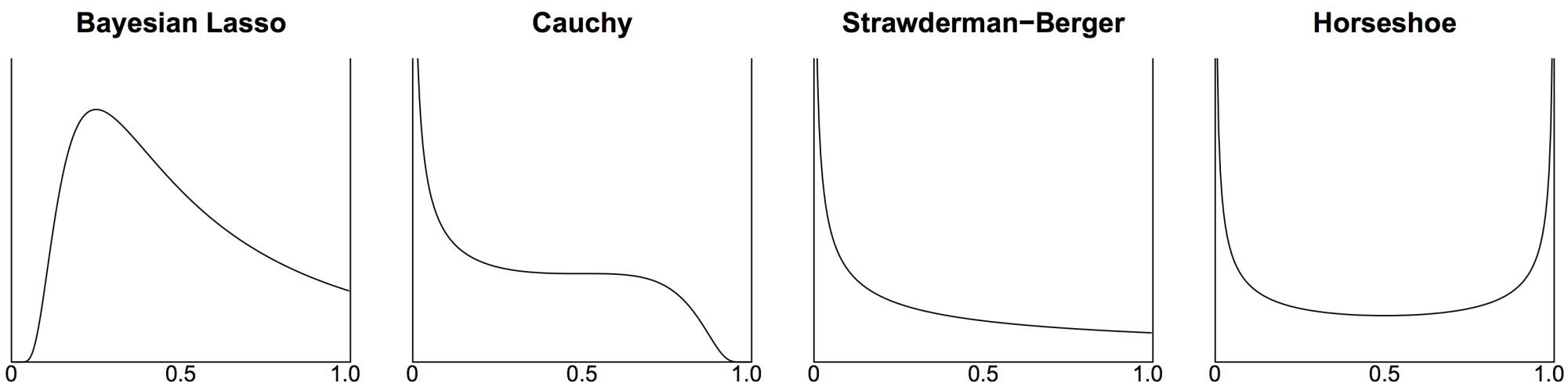
- The name 'horseshoe' came from the shape of the distribution if we reparametrize.
- Consider

$$\mathbb{E}\{\beta_j|y,\lambda_j\} = \left(1 - \frac{1}{1+\lambda_j^2}\right) y_j$$

so that, $\kappa_j = \frac{1}{1+\lambda_j^2}$, which is called the 'shrinkage weight', is a 'random' shrinkage parameter:

  - small a priori values of κ_i yield $\mathbb{E}\{\beta_j|y,\lambda_j\} \approx y_j$ and hence mean high significance/weak shrinkage $\Rightarrow$ this is good for identifying *needles in haystacks*
  - large values allow $\mathbb{E}\{\beta_j|y,\lambda_j\} \approx 0$ and hence mean low significance/strong shrinkage $\Rightarrow$ this is good for identifying straw.
  - To identify both needles and straw, the implied prior for κ_i must allow values both very near 0 and very near 1.

Density of κ_j



The implied priors for κ_i and λ_i associated with some priors for shrinkage and sparsity

Prior for β_i	Prior for λ_i	Prior for κ_i
Lasso/double-exp.	$\lambda_i^2 \sim \text{Ex}(2)$	$\pi_L(\kappa_i) \propto \kappa_i^{-2} e^{-\frac{1}{2\kappa_i}}$
Cauchy	$\lambda_i \sim \mathcal{IG}(1/2, 1/2)$	$\pi_C(\kappa_i) \propto \kappa_i^{-\frac{1}{2}}(1-\kappa_i)^{-\frac{3}{2}} e^{-\frac{\kappa_i}{1(1-\kappa_i)}}$
Strawderman-Berger	$\pi(\lambda_i) \propto \lambda_i(1+\lambda_i^2)^{-\frac{3}{2}}$	$\pi_{SB}(\kappa_i) \propto \kappa_i^{-\frac{1}{2}}$
Horseshoe	$\lambda_i \sim \text{half-}\mathcal{C}(0,1)$	$\pi_H(\kappa_i) \propto \kappa_i^{-\frac{1}{2}}(1-\kappa_i)^{-\frac{1}{2}}$

R software

- lasso per lm: `glmnet` e `lars`
- lasso per glm e multinomiale: `glmnet`
- penalizzazioni concave:
  garrota non negativa `lqa`,
  SCAD e MCP `ncvreg`
- SVM sparso: `sparseSVM` e
- group lasso: `gglasso`, `grpreg`, `glinternet`, `MSGLasso`, `seagull`
- fused lasso: `flsa`,
- sparse additive model: `SAM`, `hgam`, `cosso`, o direttamente via *group lasso*
- MARS: `earth`
- Bayesian variable selection: `BMA`, `BoomSpikeSlab`, `horseshoe`

References

- Hastie T., Tibshirani R., Wainwright M. (2015) Statistical Learning with Sparsity The Lasso and Generalizations, CRC Press Taylor & Francis Group
- Murphy K.P. (2012) Machine Learning: A Probabilistic Perspective, Mit Press, Section 13.2, pag.422
- Bai R., Rockova V., George E.I. (2021) Spike-and-Slab Meets LASSO: A Review of the Spike-and-Slab LASSO, arXiv:2010.06451v4, https://arxiv.org/abs/2010.06451