Linear Regression

David M. Blei

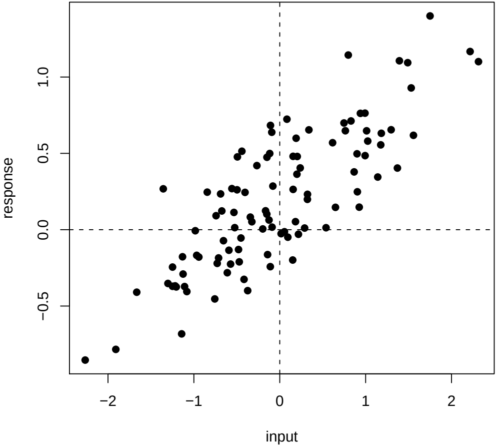
COS424
Princeton University

April 4, 2012

Regression

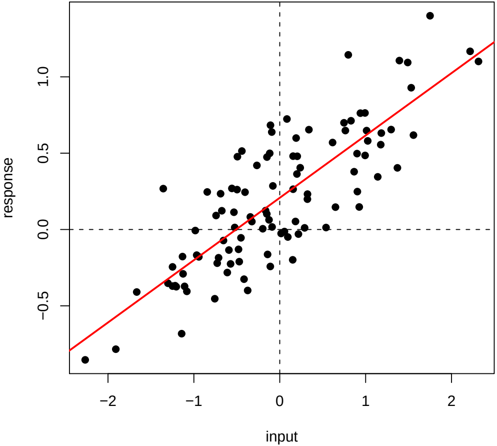
- We have studied classification, the problem of automatically categorizing data into a set of discrete classes.
- E.g., based on its words, is an email spam or ham?
- Regression is the problem of predicting a real-valued variable from data input.

Linear regression



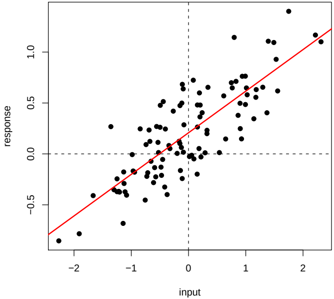
Data are a set of inputs and outputs $\mathscr{D} = \{(x_n, y_n)\}_{n=1}^{N}$

Linear regression



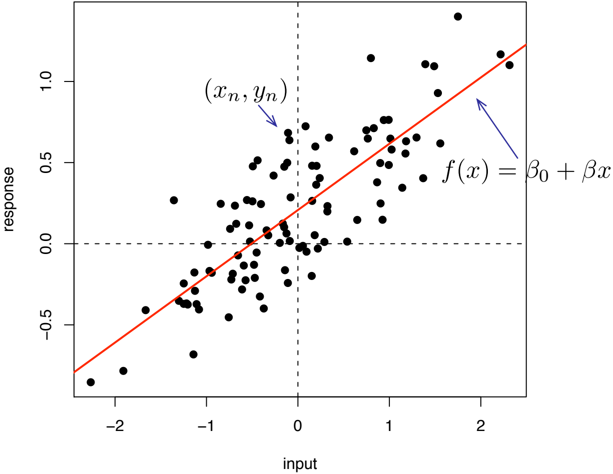
The goal is to predict *y* from *x* using a linear function.

Examples



- Given today’s weather, how much will it rain tomorrow?
- Given today’s market, what will be the price of a stock tomorrow?
- Given her emails, how long will a user stay on a page?
- Others?

Linear regression



Multiple inputs

- Usually, we have a vector of inputs, each representing a different feature of the data that might be predictive of the response.

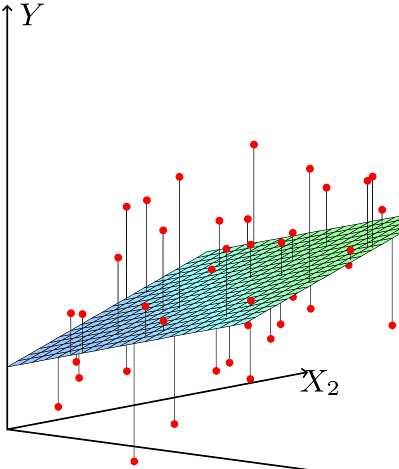
$$x = \langle x_1, x_2, \ldots, x_p \rangle$$

- The response is assumed to be a linear function of the input

$$f(x) = \beta_0 + \sum_{i=1}^{p} x_i \beta_i$$

- Here, $\beta^\top x = 0$ is a hyperplane.

Multiple inputs

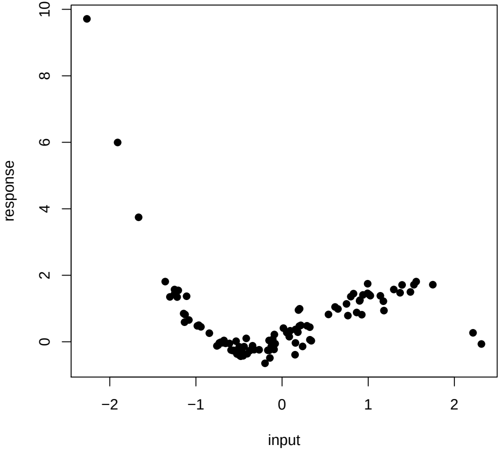


Flexibility of linear regression

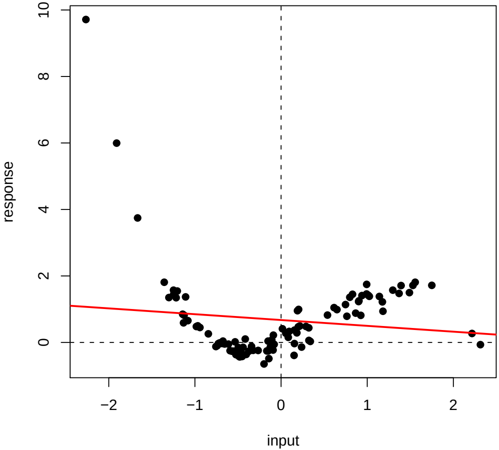
- This set-up is less limiting than you might imagine.
- Inputs can be:
  - Any features of the data
  - Transformations of the original features, e.g., $x_2 = \log x_1$ or $x_2 = \sqrt{x_1}$.
  - A basis expansion, e.g., $x_2 = x_1^2$ and $x_3 = x_1^3$
  - Indicators of qualitative inputs, e.g., category
  - Interactions between inputs, e.g., $x_1 = x_2 x_3$

- Its simplicity and flexibility make linear regression one of the most important and widely used statistical prediction techniques.

Polynomial regression example

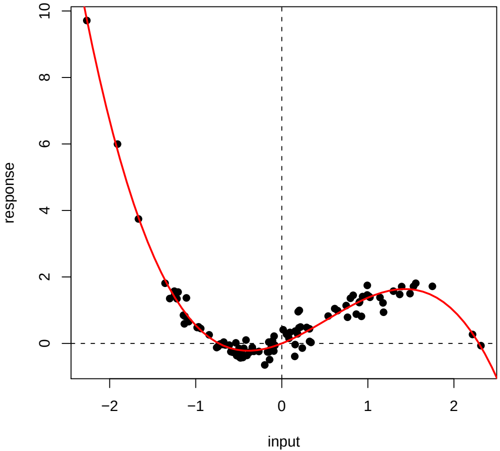


Linear regression



$$f(x) = \beta_0 + \beta x$$

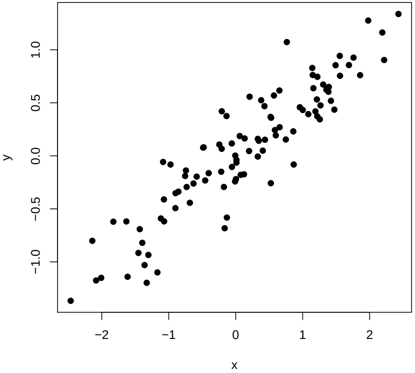
Polynomial regression



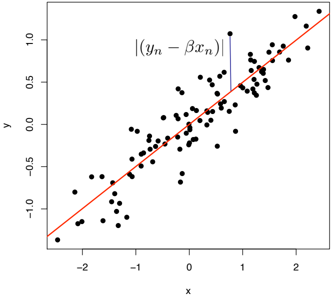
$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

Fitting a regression

- Given data $\mathscr{D} = \{(x_n, y_n)\}_{n=1}^{N}$, find the coefficient β that can predict y_{new} from x_{new}.
- Simplifications:
  - 0-intercept, i.e., $\beta_0 = 0$
  - One input, i.e., $p = 1$
- How should we proceed?



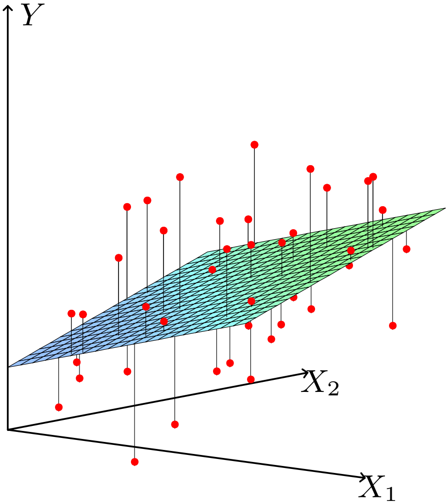
Residual sum of squares



A reasonable approach is to minimize sum of the squared Euclidean distance between each prediction βx_n and the truth y_n

$$\mathrm{RSS}(\beta) = \frac{1}{2}\sum_{n=1}^{N}(y_n - \beta x_n)^2$$

RSS for two inputs



Optimizing β

The objective function is

$$\mathrm{RSS}(\beta) = \frac{1}{2}\sum_{n=1}^{N}(y_n - \beta x_n)^2$$

The derivative is

$$\frac{d}{d\beta}\mathrm{RSS}(\beta) = -\sum_{n=1}^{N}(y_n - \beta x_n)x_n$$

The optimal value is

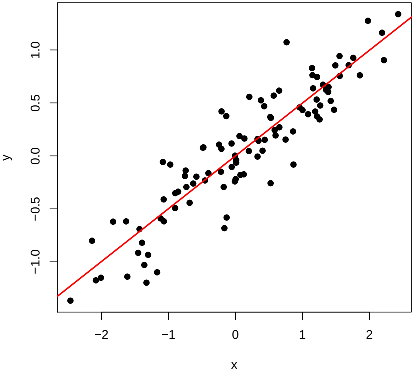
$$\hat{\beta} = \frac{\sum_{n=1}^{N} y_n x_n}{\sum_n x_n^2}$$

The optimal β

- The optimal value is

$$\hat{\beta} = \frac{\sum_{n=1}^{N} y_n x_n}{\sum_n x_n^2}$$

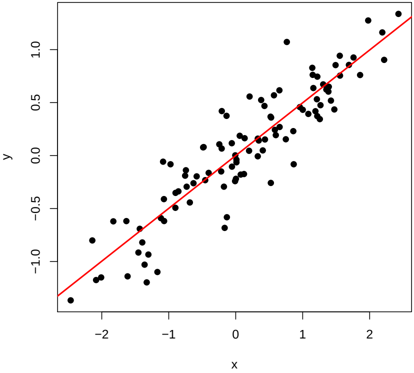
- $+$ values pull the slope up.
- $-$ values pull the slope down



Prediction

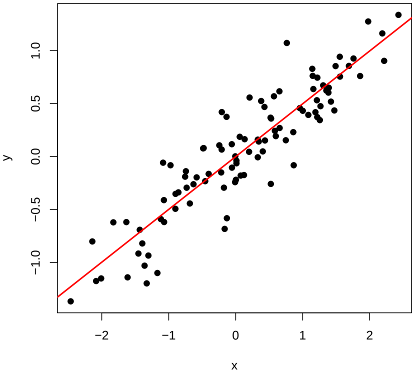
- After finding the optimal β, we would like to predict a new output from a new input.
- We use the point on the line at the input,

$$\hat{y}_{\text{new}} = \hat{\beta} x_{\text{new}}$$



Prediction

- Note the difference between classification and prediction.
- Note that linear regression assumes the input is always observed.



Multiple inputs

In general,

$$y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i$$

To simplify, let β be a $p+1$ vector and set $x_{p+1} = 1$. Now the RSS is

$$\mathrm{RSS}(\beta) = \frac{1}{2} \sum_{n=1}^{N} (y_n - \beta^\top x_n)^2$$

(Note that β_{p+1} is β_0 in the old notation.)

Multiple inputs

The objective is:

$$\mathrm{RSS}(\beta) = \frac{1}{2}\sum_{n=1}^{N}(y_n - \beta^\top x_n)^2$$

The derivative with respect to β_i is:

$$\frac{d}{d\beta_i} = -\sum_{n=1}^{N}(y_n - \beta_i x_{n,i})x_{n,i}$$

As a vector, the gradient is:

$$\nabla_\beta \mathrm{RSS} = -\sum_{n=1}^{N}(y_n - \beta^\top x_n)x_n$$

One option : optimize with some kind of gradient-based algorithm.

The normal equations

The design matrix is an $N \times (p+1)$ matrix:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,p} & 1 \\ x_{2,1} & x_{2,2} & \dots & x_{2,p} & 1 \\ & & \vdots & & \\ x_{N,1} & x_{N,2} & \dots & x_{N,p} & 1 \end{bmatrix}$$

The response vector is an N-vector:

$$y = \langle y_1, y_2, \ldots, y_N \rangle$$

Recall that the parameter vector is a $(p+1)$-vector

$$\beta = \langle \beta_1, \beta_2, \ldots, \beta_{p+1} \rangle$$

The normal equations

With these definitions, the gradient of the RSS is

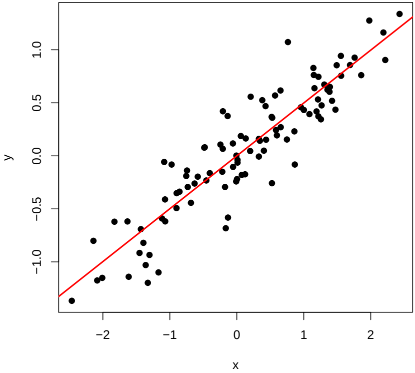
$$\nabla_\beta \mathrm{RSS} = -X^\top (y - X\beta)$$

Setting to the 0-vector and solving for β:

$$\begin{aligned} X^\top y - X^\top X\hat{\beta} &= 0 \\ X^\top X\hat{\beta} &= X^\top y \\ \hat{\beta} &= (X^\top X)^{-1} X^\top y \end{aligned}$$

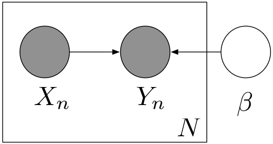
This works as long as $X^\top X$ is invertible, i.e., X is full rank.

Probabilistic interpretation



- Our reasoning so far has not included any probabilities
- It is no surprise that linear regression has a probabilistic interpretation
- What do you think that it is?

Probabilistic interpretation



- Linear regression assumes that the output are drawn from a Normal distribution whose mean is a linear function of the coefficients and the input,

$$Y_n \mid x_n, \beta \sim \mathcal{N}(\beta \cdot x_n, \sigma^2)$$

- This is like putting a Gaussian “bump” around the mean, which is a linear function of the input.
- Note that this is a conditional model. The inputs are not modeled.

Conditional maximum likelihood

We find the parameter vector β that maximizes the conditional likelihood. The conditional log likelihood of data $\mathscr{D} = \{(x_n, y_n)\}_{n=1}^N$ is

$$\begin{aligned}
\mathscr{L}(\beta) &= \log \prod_{n=1}^{N} p(y_n \,|\, x_n, \beta) \\
&= \log \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ \frac{-(y_n - \beta^\top x_n)^2}{2\sigma^2} \right\} \\
&= \sum_{n=1}^{N} -\frac{1}{2} \log 2\pi\sigma^2 - \frac{1}{2}(y_n - \beta^\top x_n)^2/\sigma^2
\end{aligned}$$

Question: What happens when we optimize with respect to β?

Conditional maximum likelihood

Maximizing the conditional log likelihood with respect to β,

$$\mathscr{L}(\beta) = \sum_{n=1}^{N} -\frac{1}{2}\log 2\pi\sigma^2 - \frac{1}{2}(y_n - \beta^\top x_n)^2/\sigma^2$$

is the same as *minimizing* the residual sum of squares

$$\mathrm{RSS}(\beta) = \frac{1}{2}(y_n - \beta^\top x_n)^2$$

The maximum likelihood estimates are identical to the estimates we obtained earlier.

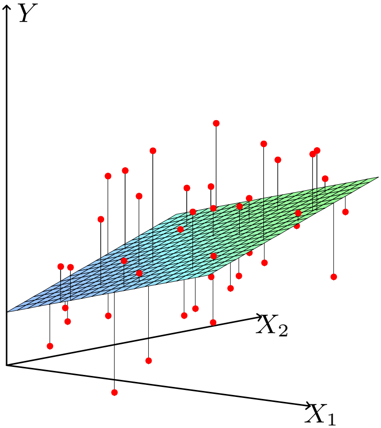
Question: What is the probabilistic interpretation of prediction?

Probabilistic prediction

- In prediction, we estimate the *conditional expectation*:

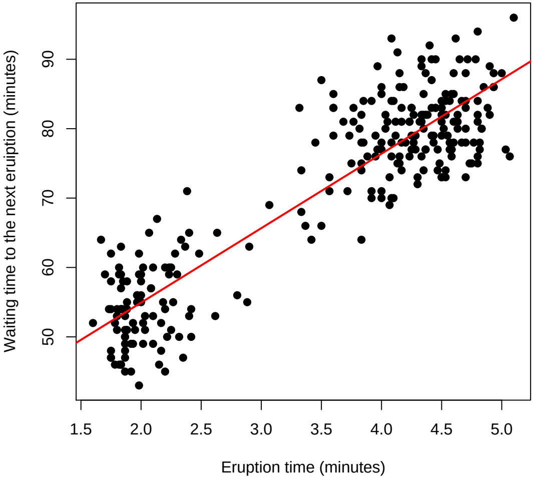
$$\mathrm{E}[y_{\text{new}} \,|\, x_{\text{new}}] = \beta^\top x_{\text{new}}$$

- This is identical to the geometric treatment.
- Note: the variance term σ^2 does not play a role in estimation or prediction.



“Real-world” example

“Real-world” example



Important aside

- A pervasive concept in machine learning and statistics is the bias variance trade-off.
- Consider a random data set that is drawn from a linear regression model,

$$Y_n | x_n, \beta \sim \mathcal{N}(\beta x_n, \sigma^2).$$

- We can contemplate the maximum likelihood estimate $\hat{\beta}$ as a *random variable* whose distribution is governed by the distribution of the data set $\mathscr{D} = \{(x_n, y_n)\}_{n=1}^{N}$.

Bias variance decomposition

Suppose we observe a new data input x, we can consider the mean squared error of our estimate of $\mathrm{E}[y|x] = \hat{\beta}x$.

$$\mathrm{MSE}(\hat{\beta}x) = \mathrm{E}_{\mathscr{D}}[(\hat{\beta}x - \beta x)^2]$$

Note that β is *not* random and $\hat{\beta}$ is random.

$$\begin{aligned} \mathrm{MSE} &= \mathrm{E}[(\hat{\beta}x)^2] - 2\mathrm{E}[\hat{\beta}x]\beta x + (\beta x)^2 \\ &= \mathrm{E}[(\hat{\beta}x)^2] - 2\mathrm{E}[(\hat{\beta}x)](\beta x) + (\beta x)^2 + \mathrm{E}[(\hat{\beta}x)]^2 - \mathrm{E}[(\hat{\beta}x)]^2 \\ &= \left(\mathrm{E}[(\hat{\beta}x)^2 - \mathrm{E}[\hat{\beta}x]^2\right) + \left(\mathrm{E}[\hat{\beta}x] - \beta x\right)^2 \end{aligned}$$

Bias variance decomposition

$$\mathrm{MSE} = \left(\mathrm{E}[(\hat{\beta}x)^2] - \mathrm{E}[\hat{\beta}x]^2\right) + \left(\mathrm{E}[\hat{\beta}x] - \beta x\right)^2$$

- The second term is the squared bias,

  $$\mathrm{bias} = \mathrm{E}[\hat{\beta}x] - \beta x$$

  An estimate for which this term is zero is an unbiased estimate.

- The first term is the variance,

  $$\mathrm{variance} = \mathrm{E}[(\hat{\beta}x)^2] - \mathrm{E}[\hat{\beta}x]^2$$

  This reflects how sensitive the estimate is to the randomness inherent in the data.

Bias variance and prediction error

What about prediction error, which is what we ultimately care about? Suppose we see a new input x. The expected squared prediction error is

$$\mathrm{E}_{\mathscr{D}}[\mathrm{E}_Y[(\hat{\beta}x - Y)^2]]$$

The first expectation is taken for the randomness of $\hat{\beta}$. The second is taken for the randomness of Y given x.

$$\begin{aligned}\mathrm{E}_{\mathscr{D}}[\mathrm{E}_Y[(\hat{\beta}x - Y)^2]] &= \mathrm{Var}(Y) + \mathrm{MSE}(\hat{\beta}x) \\ &= \sigma^2 + \mathrm{Bias}^2(\hat{\beta}x) + \mathrm{Var}(\hat{\beta}x)\end{aligned}$$

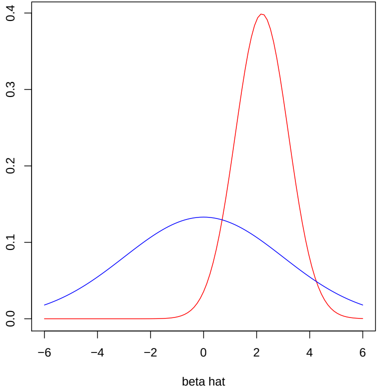
The first term is the inherent uncertainty around the true mean; the second two terms are the bias variance decomposition of the estimator.

Gauss-Markov theorem

$$\mathrm{MSE} = \left(\mathrm{E}[(\hat{\beta}x)^2] - \mathrm{E}[\hat{\beta}x]^2\right) + \left(\mathrm{E}[\hat{\beta}x] - \beta x\right)^2$$

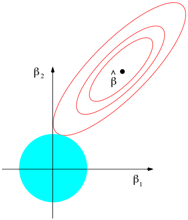
The *Gauss-Markov* theorem states that the MLE/least squares estimate of β is the unbiased estimate with smallest variance.

Bias variance trade-off



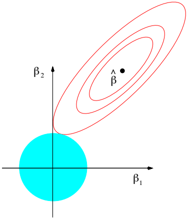
- Classical statistics focuses on unbiased estimates.
- Modern statistics has explored the *trade-off*.
- We might sacrifice a little bias for a larger reduction in variance.

Regularization



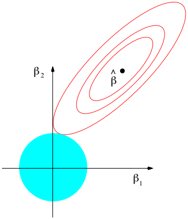
- In regression, we can make this trade-off with regularization, which means placing constraints on the coefficients β.
- Intuitively, this reduces the variance because it limits the space that the parameter vector β can live in.
- If the true MLE of β lives outside that space, then the resulting estimate *must* be biased because of the Gauss-Markov theorem.

Regularization



- Regularization encourages smaller and simpler models.
- Intuitively, simpler models are more robust to overfitting, generalizing pooly because of a close match to the training data.
- Simpler models can also be more interpretable, which is another goal of regression.

Ridge regression

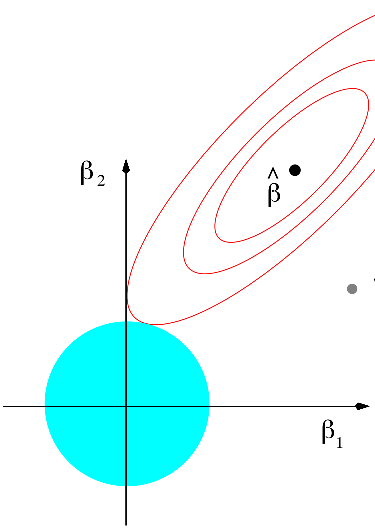


- In ridge regression, we optimize the RSS subject to a constraint on the sum of squares of the coefficients,

$$\begin{array}{ll} \text{minimize} & \sum_{n=1}^{N} \frac{1}{2}(y_n - \beta x_n)^2 \\ \text{subject to} & \sum_{i=1}^{p} \beta_i^2 \leq s \end{array}$$

- This constrains the coefficients to live within a sphere of radius s.

Ridge regression



- What happens as *s* increases?

Ridge regression

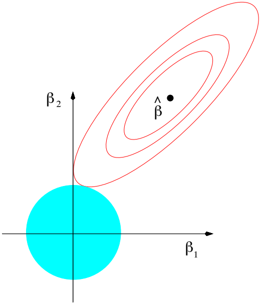
- The ridge regression estimate can also be expressed as

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \sum_{n=1}^{N} \frac{1}{2}(y_n - \beta x_n)^2 + \lambda \sum_{i=1}^{p} \beta_i^2$$

- This problem is convex.
- If the covariates are uncorrelated, it has an analytic solution. (You'll see this on your homework.)

Ridge regression

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \sum_{n=1}^{N} \frac{1}{2}(y_n - \beta x_n)^2 + \lambda \sum_{i=1}^{p} \beta_i^2$$

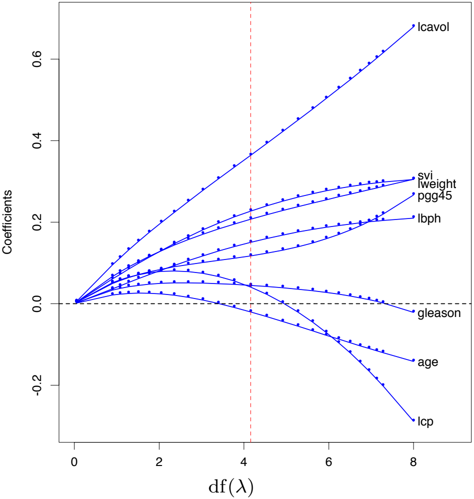


- There is a 1-1 mapping between s and λ.
- λ is the complexity parameter
- It determines the radius of the sphere
- Trades off an increase in bias for a decrease in variance

Prostate cancer data

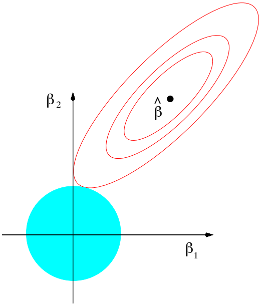
- Study from Stamey et al. (1989)
- Examined the correlation between the level of prostate-specific antigen and a number of clinical measures in mean about to receive a procedure
- Variables are
  - log cancer volume
  - log prostate weight
  - age
  - log of the amount of benign prostatic hyperplasia
  - seminal besicle invasion
  - log of capsular penetration
  - Gleason score
  - percent of Gleason scores 4 or 5

Coefficients as a function of λ



How can we choose λ?

Choosing λ



- The choice of complexity parameter greatly affects our estimate
- What would happen if we used training error as the criterion?
- In practice, λ is chosen by cross validation.
- This is an attempt to minimize *test error*.

Cross-validation to choose the complexity parameter

- Divide the data into 10 folds
- Decide on candidate values of λ (e.g., a grid between 0 and 1)
- For each fold and value of λ,
    - Estimate $\hat{\beta}^{\text{ridge}}$ on the out-of-fold samples.
    - For each within-fold sample x_n, compute its squared error

$$\epsilon_n = (\hat{y}_n - y_n)^2$$

- The score for that value of λ is

$$\text{MSE}(\lambda) = \frac{1}{N}\sum_{n=1}^{N} \epsilon_n$$

- Choose the value of λ that minimizes this score.

Cross-validation to choose the complexity parameter

- The score for that value of λ is

$$\mathrm{MSE}(\lambda) = \frac{1}{N}\sum_{n=1}^{N} \epsilon_n$$

- Choose the value of λ that minimizes this value.
- Notice that each ϵ_n was computed from a model that did not include the nth data point in its fit.
- Thus, $MSE(\lambda)$ is an estimate of test error.
- Dave, draw a picture on the board.

Aside: Bayesian statistics

- In Bayesian statistics, we treat the *parameter* as a *random variable*.
- In the model, it is endowed with a prior distribution.
- Rather than estimate the parameter, we perform posterior inference.
- In general,

$$\begin{aligned} \theta &\sim G_0(\alpha) \\ y_n &\sim F(\theta) \end{aligned}$$

and posterior inference is concerned with

$$p(\theta \,|\, y_1, \ldots, y_N, \alpha)$$

- The parameter to the prior α is called a hyperparameter.

Aside: Bayesian statistics

There are two usual ways of using the posterior to obtain an estimate

- Maximum a posteriori estimates

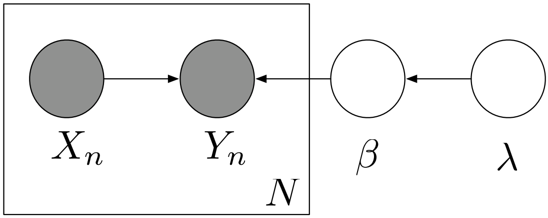
$$\theta^{\mathrm{MAP}} = \arg\max_{\theta} p(\theta \,|\, y_1, \ldots, y_N, \alpha)$$

- Posterior mean estimate

$$\theta^{\mathrm{mean}} = \mathrm{E}[\theta \,|\, y_1, \ldots, y_N, \alpha]$$

- *Why are these different from the MLE?*

Ridge regression



Ridge regression corresponds to MAP estimation in the following model:

$$\begin{aligned} \beta_i &\sim \mathcal{N}(0, 1/\lambda) \\ Y_n | x_n, \beta &\sim \mathcal{N}(\beta^\top x_n, \sigma^2) \end{aligned}$$

Bayesian interpretation of ridge regression

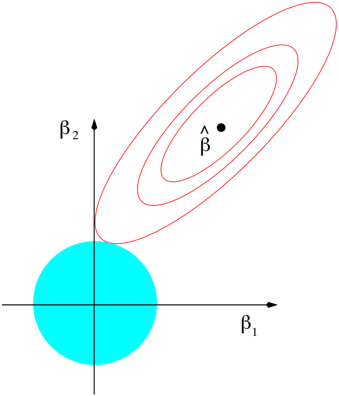
Note that

$$p(\beta_i \,|\, \lambda) = \frac{1}{\sqrt{2\pi(1/\lambda)}} \exp\{\lambda \beta_i^2\}$$

Let's compute the MAP estimate of β:

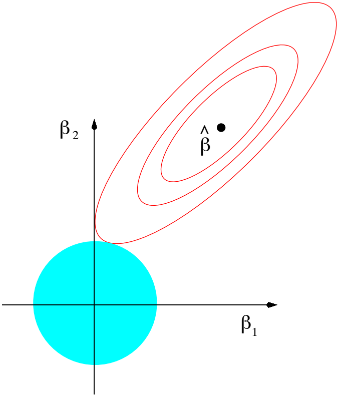
$$\begin{aligned}
\max_{\beta} p(\beta \,|\, y_{1:N}, x_{1:N}, \lambda) &= \max_{\beta} \log p(\beta \,|\, y_{1:N}, x_{1:N}, \lambda) \\
&= \max_{\beta} \log p(\beta, y_{1:N} \,|\, x_{1:N}, \lambda) \\
&= \max_{\beta} \log \left(p(y_{1:N} \,|\, x_{1:N}, \beta) \prod_{i=1}^{p} p(\beta_i \,|\, \lambda) \right) \\
&= \max_{\beta} -RSS(\beta; y_{1:N}, x_{1:N}) - \sum_{i=1}^{p} \lambda \beta_i^2
\end{aligned}$$

Bayesian intuitions



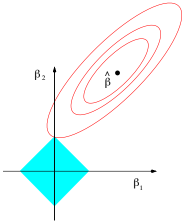
- The hyperparameter controls how far away the estimate will be from the MLE
- A small hyperparameter (large variance) will choose the MLE, i.e., the data totally determine the estimate
- As the hyperparameter gets larger, the estimate moves further from the MLE. The prior ($\mathrm{E}[\beta] = 0$) becomes more influential.
- A theme in Bayesian estimation: Both the data and the prior influence the answer.

Summary of ridge regression



- We constrain β to be in a hypersphere around 0.
- This is equivalent to minimizing the RSS plus a regularization term.
- We no longer find the $\hat{\beta}$ that minimizes the RSS. (Contours illustrate constant RSS.)
- Also called *shrinkage*, because we are reducing the components to be close to 0 and close to each other
- Ridge estimates trade off bias for variance.

The lasso



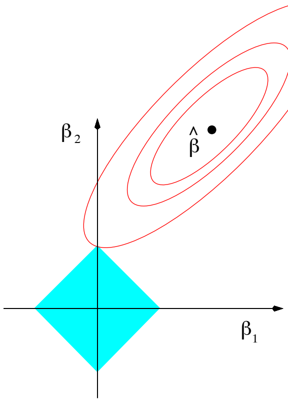
- A related regularization method is called the lasso.
- We optimize the RSS subject to a different constraint.

$$\text{minimize} \quad \sum_{n=1}^{N} \frac{1}{2}(y_n - \beta x_n)^2$$

$$\text{subject to} \quad \sum_{i=1}^{p} |\beta_i| \leq s$$

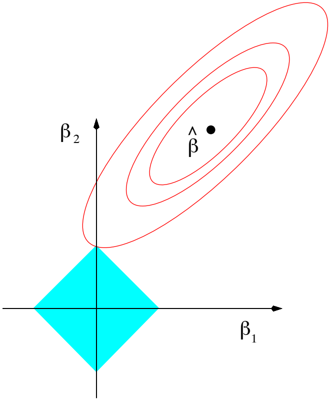
- This small change yields very different estimates.

Lasso



- What happens as s increases?
- Where is the solution going to lie?

Lasso



- It's a fact: unless it chooses $\hat{\beta}$, the lasso will set some of the coefficients to exactly zero.
- This is a form of feature selection, identifying a relevant subset of our inputs to perform prediction.
- Trades off an increase in bias with a decrease in variance
- And, provides interpretable (sparse) models

Lasso

- The lasso is equivalent to

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \sum_{n=1}^{N} \frac{1}{2}(y_n - \beta x_n)^2 + \lambda \sum_{i=1}^{p} |\beta_i|$$

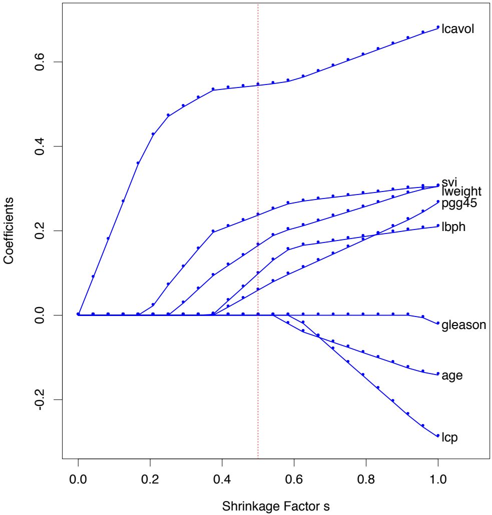
- Again, there is a 1-1 mapping between λ and s
- This objective is still convex!

Why the lasso is exciting

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \sum_{n=1}^{N} \frac{1}{2}(y_n - \beta x_n)^2 + \lambda \sum_{i=1}^{p} |\beta_i|$$

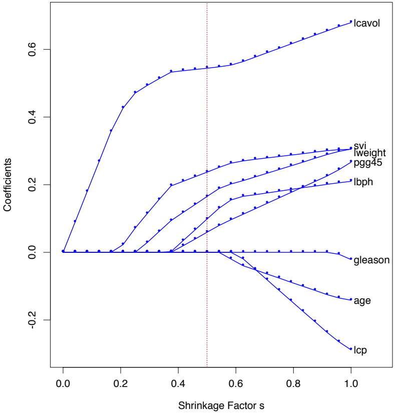
- Prior to the lasso, the only “sparse” method was subset selection, finding the best subset of features with which to model the data
- But, searching over all subsets is very computationally expensive
- The lasso efficiently finds a sparse solution with convex optimization.
- This is akin to a “smooth version” of subset selection.
- Note: the lasso won’t consider all possible subsets.

Optimizing λ



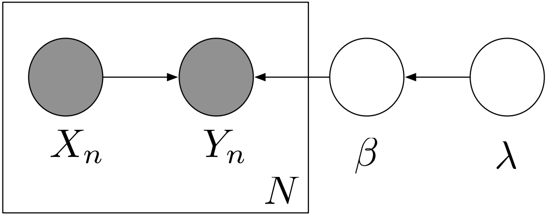
As we increase s (decrease λ), coefficients become non-zero.

Choosing λ with LARS



- Again, we choose the complexity parameter λ with cross-validation.
- The LARS algorithm (Efron et al., 2004) lets us efficiently explore the entire regularization path of λ.

Bayesian interpretation of the lasso



Lasso regression corresponds to MAP estimation in the following model:

$$\begin{aligned} \beta_i &\sim \mathrm{Laplace}(\lambda) \\ Y_n \mid x_n, \beta &\sim \mathcal{N}(\beta^\top x_n, \sigma^2) \end{aligned}$$

Where the coefficients come from a Laplace distribution

$$p(\beta_i \mid \lambda) = \frac{1}{2} \exp\{-\lambda |\beta_i|\}$$

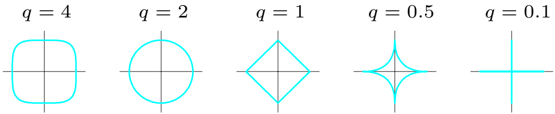
Generalized regularization

- In general, regularization can be seen as minimizing the RSS with a constraint on a q-norm,

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{n=1}^{N} \frac{1}{2}(y_n - \beta x_n)^2 \\ \text{subject to} \quad & ||\beta||_q \leq s \end{aligned}$$

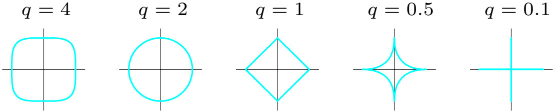
- The methods we discussed so far:
  - $q = 2$: ridge regression
  - $q = 1$: lasso
  - $q = 0$: subset selection

Generalized regularization



- This brings us away from the minimum RSS solution, but might provide better test prediction via the bias/variance trade-off.
- Complex models have less bias; simpler models have less variance. Regularization encourages simpler models.

Generalized regularization



- Each of these methods correspond to a Bayesian solution with a different choice of prior.

$$\hat{\beta}^{\mathrm{ridge}} = \arg\min_{\beta} \sum_{n=1}^{N} \frac{1}{2}(y_n - \beta x_n)^2 + \lambda ||\beta||_q$$

- The complexity parameter λ can be chosen with cross validation.
- Lasso ($q = 1$) is the only norm that provides sparsity and convexity.

Regularization comparison

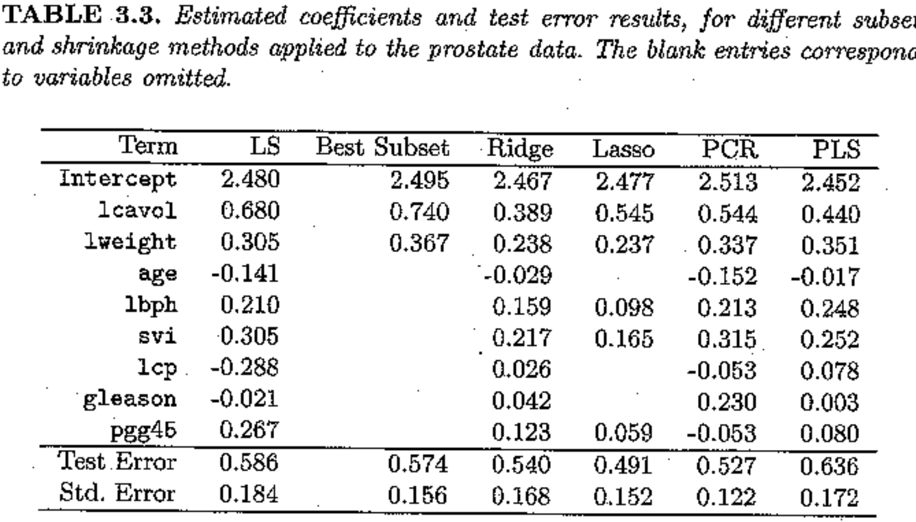


TABLE 3.3. *Estimated coefficients and test error results, for different subse and shrinkage methods applied to the prostate data. The blank entries correspond to variables omitted.*

Term	LS	Best Subset	Ridge	Lasso	PCR	PLS
Intercept	2.480	2.495	2.467	2.477	2.513	2.452
lcavol	0.680	0.740	0.389	0.545	0.544	0.440
lweight	0.305	0.367	0.238	0.237	0.337	0.351
age	-0.141		-0.029		-0.152	-0.017
lbph	0.210		0.159	0.098	0.213	0.248
svi	0.305		0.217	0.165	0.315	0.252
lcp	-0.288		0.026		-0.053	0.078
gleason	-0.021		0.042		0.230	0.003
pgg45	0.267		0.123	0.059	-0.053	0.080
Test Error	0.586	0.574	0.540	0.491	0.527	0.636
Std. Error	0.184	0.156	0.168	0.152	0.122	0.172