ColumbiaX: Machine Learning
Lecture 5

Prof. John Paisley

Department of Electrical Engineering
& Data Science Institute

Columbia University

BAYESIAN LINEAR REGRESSION

Model

Have vector $y \in \mathbb{R}^n$ and covariates matrix $X \in \mathbb{R}^{n \times d}$. The ith row of y and X correspond to the ith observation (y_i, x_i).

In a Bayesian setting, we model this data as:

$$\begin{aligned} \textbf{Likelihood}: &\quad y \sim N(Xw, \sigma^2 I) \\ \textbf{Prior}: &\quad w \sim N(0, \lambda^{-1} I) \end{aligned}$$

The unknown model variable is $w \in \mathbb{R}^d$.

- The "likelihood model" says how well the observed data agrees with w.
- The "model prior" is our prior belief (or constraints) on w.

This is called Bayesian linear regression because we have defined a prior on the unknown parameter and will try to learn its posterior.

Review: Maximum a posteriori inference

MAP solution

MAP inference returns the maximum of the log joint likelihood.

$$\textbf{Joint Likelihood}: \quad p(y, w|X) = p(y|w, X)p(w)$$

Using Bayes rule that this point also maximizes the *posterior* of w.

$$\begin{aligned} w_{\text{MAP}} &= \arg\max_{w} \ \ln p(w|y, X) \\ &= \arg\max_{w} \ \ln p(y|w, X) + \ln p(w) \\ &= \arg\max_{w} \ -\frac{1}{2\sigma^2}(y - Xw)^T(y - Xw) - \frac{\lambda}{2}w^T w + \text{const.} \end{aligned}$$

We saw that this solution for w_{MAP} is the same as for ridge regression:

$$w_{\text{MAP}} = (\lambda\sigma^2 I + X^T X)^{-1} X^T y \quad \Leftrightarrow \quad w_{\text{RR}}$$

Point estimates vs Bayesian inference

Point estimates

w_{MAP} and w_{ML} are referred to as *point estimates* of the model parameters.

They find a specific value (point) of the vector w that maximizes an objective function (MAP or ML).

- **ML**: Only consider data model: $p(y|w,X)$.
- **MAP**: Takes into account model prior: $p(y,w|X) = p(y|w,X)p(w)$.

Bayesian inference

Bayesian inference goes one step further by characterizing uncertainty about the values in w using Bayes rule.

Bayes rule and linear regression

Posterior calculation

Since w is a continuous-valued random variable in $\mathbb{R}^d$, Bayes rule says that the *posterior* distribution of w given y, X is

$$p(w|y,X) = \frac{p(y|w,X)p(w)}{\int_{\mathbb{R}^d} p(y|w,X)p(w)\,dw}$$

That is, we get an updated distribution on w through the transition

$$\text{prior} \quad \rightarrow \quad \text{likelihood} \quad \rightarrow \quad \text{posterior}$$

Quote: "The posterior of __ is proportional to the likelihood times the prior."

Fully Bayesian inference

Bayesian linear regression

In this case, we can update the posterior distribution $p(w|y,X)$ analytically.

We work with the proportionality first:

$$\begin{aligned} p(w|y,X) &\propto p(y|w,X)p(w) \\ &\propto \left[\mathrm{e}^{-\frac{1}{2\sigma^2}(y-Xw)^T(y-Xw)}\right]\left[\mathrm{e}^{-\frac{\lambda}{2}w^Tw}\right] \\ &\propto \mathrm{e}^{-\frac{1}{2}\{w^T(\lambda I+\sigma^{-2}X^TX)w-2\sigma^{-2}w^TX^Ty\}} \end{aligned}$$

The $\propto$ sign lets us multiply and divide this by anything *as long as it doesn't contain w*. We've done this in two lines above.

BAYESIAN INFERENCE FOR LINEAR REGRESSION

We need to normalize:

$$p(w|y,X) \quad \propto \quad \mathrm{e}^{-\frac{1}{2}\{w^T(\lambda I+\sigma^{-2}X^TX)w-2\sigma^{-2}w^TX^Ty\}}$$

There are two key terms in the exponent:

$$\underbrace{w^T(\lambda I+\sigma^{-2}X^TX)w}_{\text{quadratic in } w}-\underbrace{2w^TX^Ty/\sigma^2}_{\text{linear in } w}$$

We can conclude that $p(w|y,X)$ is Gaussian. Why?

1. We can multiply and divide by anything not involving w.
2. A Gaussian has $(w-\mu)^T\Sigma^{-1}(w-\mu)$ in the exponent.
3. We can "complete the square" by adding terms not involving w.

BAYESIAN INFERENCE FOR LINEAR REGRESSION

Compare: In other words, a Gaussian looks like:

$$p(w|\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \mathrm{e}^{-\frac{1}{2}(w^T\Sigma^{-1}w - 2w^T\Sigma^{-1}\mu + \mu^T\Sigma^{-1}\mu)}$$

and we've shown for some setting of Z that

$$p(w|y,X) \;=\; \frac{1}{Z}\mathrm{e}^{-\frac{1}{2}(w^T(\lambda I + \sigma^{-2}X^TX)w - 2w^TX^Ty/\sigma^2)}$$

Conclude: What happens if in the above Gaussian we define:

$$\Sigma^{-1} = (\lambda I + \sigma^{-2}X^TX), \qquad \Sigma^{-1}\mu = X^Ty/\sigma^2\,?$$

Using these specific values of μ and Σ we only need to set

$$Z = (2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}\mathrm{e}^{\frac{1}{2}\mu^T\Sigma^{-1}\mu}$$

BAYESIAN INFERENCE FOR LINEAR REGRESSION

The posterior distribution

Therefore, the posterior distribution of w is:

$$p(w|y,X) \quad = \quad N(w|\mu,\Sigma),$$

$$\begin{aligned} \Sigma &= (\lambda I + \sigma^{-2} X^T X)^{-1}, \\ \mu &= (\lambda \sigma^2 I + X^T X)^{-1} X^T y \quad \Leftarrow \quad w_{\text{MAP}} \end{aligned}$$

Things to notice:

- $\mu = w_{\text{MAP}}$ after a redefinition of the regularization parameter λ.
- Σ captures uncertainty about w as $\text{Var}[w_{\text{LS}}]$ and $\text{Var}[w_{\text{RR}}]$ did before.
- However, now we have a full probability distribution on w.

USES OF THE POSTERIOR DISTRIBUTION

Understanding w

We saw how we could calculate the variance of w_{LS} and w_{RR}. Now we have an entire distribution. Some questions we can ask are:

Q: Is $w_i > 0$ or $w_i < 0$? Can we confidently say $w_i \neq 0$?

A: Use the *marginal posterior distribution*: $w_i \sim N(\mu_i, \Sigma_{ii})$.

Q: How do w_i and w_j relate?

A: Use their joint marginal posterior distribution:

$$\left[\begin{array}{c} w_i \\ w_j \end{array}\right] \sim N\left(\left[\begin{array}{c} \mu_i \\ \mu_j \end{array}\right], \left[\begin{array}{cc} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{array}\right]\right)$$

Predicting new data

The posterior $p(w|y, X)$ is perhaps most useful for predicting new data.

PREDICTING NEW DATA

PREDICTING NEW DATA

Recall: For a new pair (x_0, y_0) with x_0 measured and y_0 unknown, we can predict y_0 using x_0 and the LS or RR (i.e., ML or MAP) outputs:

$$y_0 \approx x_0^T w_{\text{LS}} \quad \text{or} \quad y_0 \approx x_0^T w_{\text{RR}}$$

With Bayes rule, we can make a *probabilistic* statement about y_0:

$$\begin{aligned} p(y_0|x_0, y, X) &= \int_{\mathbb{R}^d} p(y_0, w|x_0, y, X)\, dw \\ &= \int_{\mathbb{R}^d} p(y_0|w, x_0, y, X)\, p(w|x_0, y, X)\, dw \end{aligned}$$

Notice that *conditional independence* lets us write

$$p(y_0|w, x_0, y, X) = \underbrace{p(y_0|w, x_0)}_{\textit{likelihood}} \quad \text{and} \quad p(w|x_0, y, X) = \underbrace{p(w|y, X)}_{\textit{posterior}}$$

PREDICTING NEW DATA

Predictive distribution (intuition)

This is called the *predictive distribution*:

$$p(y_0|x_0, y, X) = \int_{\mathbb{R}^d} \underbrace{p(y_0|x_0, w)}_{\textit{likelihood}} \underbrace{p(w|y, X)}_{\textit{posterior}} \, dw$$

Intuitively, we evaluate the likelihood of a new y_0 for a particular w and observed x_0, and weight it by our current belief about w given data (y, X).

We then sum (integrate) over all possible values of w.

PREDICTING NEW DATA

We know from the model and Bayes rule that

$$\begin{aligned} \text{Model:} \quad p(y_0|x_0, w) &= N(y_0|x_0^T w, \sigma^2), \\ \text{Bayes rule:} \quad p(w|y, X) &= N(w|\mu, \Sigma). \end{aligned}$$

With μ and Σ calculated on a previous slide.

The predictive distribution can be calculated exactly with these distributions. Again we get a Gaussian distribution:

$$\begin{aligned} p(y_0|x_0, y, X) &= N(y_0|\mu_0, \sigma_0^2), \\ \mu_0 &= x_0^T \mu, \\ \sigma_0^2 &= \sigma^2 + x_0^T \Sigma x_0. \end{aligned}$$

Notice that the expected value is the MAP prediction since $\mu = x_0^T w_{\text{MAP}}$, but we now quantify our confidence in this prediction with the variance σ_0^2.

ACTIVE LEARNING

PRIOR → POSTERIOR → PRIOR

Bayesian learning is naturally thought of as a sequential process. That is, the posterior after seeing some data becomes the prior for the next data.

Let y and X be "old data" and y_0 and x_0 be some "new data". By Bayes rule

$$p(w|y_0, x_0, y, X) \propto p(y_0|w, x_0)p(w|y, X).$$

The posterior after (y, X) has become the prior for (y_0, x_0).

Simple modifications can be made sequentially:

$$\begin{aligned} p(w|y_0, x_0, y, X) &= N(w|\mu, \Sigma), \\ \Sigma &= (\lambda I + \sigma^{-2}(x_0 x_0^T + \textstyle\sum_{i=1}^n x_i x_i^T))^{-1}, \\ \mu &= (\lambda\sigma^2 I + (x_0 x_0^T + \textstyle\sum_{i=1}^n x_i x_i^T)^{-1}(x_0 y_0 + \textstyle\sum_{i=1}^n x_i y_i). \end{aligned}$$

INTELLIGENT LEARNING

Of course, we could also have written

$$p(w|y_0, x_0, y, X) \propto p(y_0, y|w, X, x_0)p(w)$$

but often we want to use the sequential aspect of inference to help us learn.

Learning w and making predictions for new y_0 is a two-step procedure:

- Form the predictive distribution $p(y_0|x_0, y, X)$.
- Update the posterior distribution $p(w|y, X, y_0, x_0)$.

Question: Can we learn $p(w|y, X)$ intelligently?

That is, if we're in the situation where we can pick which y_i to measure with the knowledge of $\mathcal{D} = \{x_1, \dots, x_n\}$, can we come up with a good strategy?

ACTIVE LEARNING

An "active learning" strategy

Imagine we already have a measured dataset (y, X) and posterior $p(w|y, X)$. We can construct the predictive distribution for every remaining $x_0 \in \mathcal{D}$.

$$\begin{aligned} p(y_0|x_0, y, X) &= N(y_0|\mu_0, \sigma_0^2), \\ \mu_0 &= x_0^T \mu, \\ \sigma_0^2 &= \sigma^2 + x_0^T \Sigma x_0. \end{aligned}$$

For each x_0, σ_0^2 tells how confident we are. This suggests the following:

1. Form predictive distribution $p(y_0|x_0, y, X)$ for all unmeasured $x_0 \in \mathcal{D}$
2. Pick the x_0 for which σ_0^2 is largest and measure y_0
3. Update the posterior $p(w|y, X)$ where $y \leftarrow (y, y_0)$ and $X \leftarrow (X, x_0)$
4. Return to $\#1$ using the updated posterior

Entropy (i.e., uncertainty) minimization

When devising a procedure such as this one, it's useful to know what *objective function* is being optimized in the process.

We introduce the concept of the *entropy* of a distribution. Let $p(z)$ be a continuous distribution, then its (differential) entropy is:

$$\mathcal{H}(p) = -\int p(z) \ln p(z) dz.$$

This is a measure of the spread of the distribution. Larger values correspond to a more "uncertain" distribution (more variance).

The entropy of a multivariate Gaussian is

$$\mathcal{H}(N(w|\mu, \Sigma)) = \frac{d}{2} \ln \Big(2\pi \mathrm{e} |\Sigma| \Big).$$

ACTIVE LEARNING

The entropy of a Gaussian changes with its covariance matrix. With sequential Bayesian learning, the covariance transitions from

$$\begin{aligned} \text{Prior}: &\quad (\lambda I + \sigma^{-2} X^T X)^{-1} && \equiv \; \Sigma \\ &\qquad\qquad \Downarrow \\ \text{Posterior}: &\quad (\lambda I + \sigma^{-2}(x_0 x_0^T + X^T X))^{-1} && \equiv \; (\Sigma^{-1} + \sigma^{-2} x_0 x_0^T)^{-1} \end{aligned}$$

Using a rank-one update property of the determinant, the entropy of the prior $\mathcal{H}_{\text{prior}}$ is related to the entropy of the posterior $\mathcal{H}_{\text{post}}$ as follows:

$$\mathcal{H}_{\text{post}} = \mathcal{H}_{\text{prior}} - \frac{d}{2} \ln(1 + \sigma^{-2} x_0^T \Sigma x_0)$$

Therefore, the x_0 that minimizes $\mathcal{H}_{\text{post}}$ also maximizes $\sigma^2 + x_0^T \Sigma x_0$. We are minimizing $\mathcal{H}$ myopically, so this is called a “greedy algorithm”.

Model selection

SELECTING λ

We've discussed λ as a "nuisance" parameter that can impact performance.

Bayes rule gives a principled way to do this via *evidence maximization*:

$$p(w|y,X,\lambda) \;=\; \underbrace{p(y|w,X)}_{\textit{likelihood}}\underbrace{p(w|\lambda)}_{\textit{prior}} \;/\; \underbrace{p(y|X,\lambda)}_{\textit{evidence}}.$$

The "evidence" gives the likelihood of the data with *w* integrated out. It's a measure of how good our model and parameter assumptions are.

SELECTING λ

If we want to set λ, we can also do it by maximizing the evidence.

$$\hat{\lambda} = \arg\max_{\lambda} \; \ln p(y|X, \lambda).$$

We can show that the distribution of y is $p(y|X, \lambda) = N(y|0, \sigma^2 I + \lambda^{-1} X^T X)$. This requires an algorithm to maximize over λ.

We notice that this looks exactly like maximum likelihood, and it is:

Type-I ML: Maximize the likelihood over the "main parameter" (w).

Type-II ML: Integrate out "main parameter" (w) and maximize over the "hyperparameter" (λ). Also called *empirical Bayes*.

The difference is only in their perspective.

This approach requires that we can solve this integral, but often we can't for more complex models. Cross-validation is the method that always works.

ColumbiaX: Machine Learning
Lecture 6

Prof. John Paisley

Department of Electrical Engineering
& Data Science Institute

Columbia University

UNDERDETERMINED LINEAR EQUATIONS

We now consider the regression problem $y = Xw$ where $X \in \mathbb{R}^{n \times d}$ is “fat” (i.e., $d \gg n$). This is called an “underdetermined” problem.

- w now has an infinite number of solutions satisfying $y = Xw$.

$$\begin{bmatrix} \\ y \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \qquad\qquad X \qquad\qquad \\ \\ \end{bmatrix} \begin{bmatrix} \\ \\ w \\ \\ \\ \\ \end{bmatrix}$$

These sorts of high-dimensional problems often come up:

- In gene analysis there are 1000’s of genes but only 100’s of subjects.
- Images can have millions of pixels.
- Even polynomial regression can quickly lead to this scenario.

Minimum ℓ_2 regression

ONE SOLUTION (LEAST NORM)

One possible solution to the underdetermined problem is

$$w_{\text{ln}} = X^T(XX^T)^{-1}y \quad \Rightarrow \quad Xw_{\text{ln}} = XX^T(XX^T)^{-1}y = y.$$

We can construct another solution by adding to w_{ln} a vector $\delta \in \mathbb{R}^d$ that is in the *null space* $\mathcal{N}$ of X:

$$\delta \in \mathcal{N}(X) \quad \Rightarrow \quad X\delta = 0 \ \text{ and } \ \delta \neq 0$$

and so $X(w_{\text{ln}} + \delta) = Xw_{\text{ln}} + X\delta = y + 0$.

In fact, there are an infinite number of possible δ, because $d > n$.

We can show that w_{ln} is the solution with smallest ℓ_2 norm. We will use the proof of this fact as an excuse to introduce two general concepts.

TOOLS: ANALYSIS

We can use *analysis* to prove that w_{ln} satisfies the optimization problem

$$w_{\text{ln}} = \arg\min_{w} \|w\|^2 \quad \text{subject to} \quad Xw = y.$$

(Think of mathematical analysis as the use of inequalities to prove things.)

Proof: Let w be another solution to $Xw = y$, and so $X(w - w_{\text{ln}}) = 0$. Also,

$$\begin{aligned}(w - w_{\text{ln}})^T w_{\text{ln}} &= (w - w_{\text{ln}})^T X^T (XX^T)^{-1} y \\ &= \underbrace{(X(w - w_{\text{ln}}))}_{=0}{}^T (XX^T)^{-1} y = 0\end{aligned}$$

As a result, $w - w_{\text{ln}}$ is *orthogonal* to w_{ln}. It follows that

$$\|w\|^2 = \|w - w_{\text{ln}} + w_{\text{ln}}\|^2 = \|w - w_{\text{ln}}\|^2 + \|w_{\text{ln}}\|^2 + 2\underbrace{(w - w_{\text{ln}})^T w_{\text{ln}}}_{=0} > \|w_{\text{ln}}\|^2$$

TOOLS: LAGRANGE MULTIPLIERS

Instead of starting from the solution, start from the problem,

$$w_{\text{ln}} = \arg\min_{w}\ w^T w \quad \text{subject to} \quad Xw = y.$$

- Introduce Lagrange multipliers: $\mathcal{L}(w, \eta) = w^T w + \eta^T (Xw - y)$.
- Minimize $\mathcal{L}$ over w maximize over η. If $Xw \neq y$, we can get $\mathcal{L} = +\infty$.
- The optimal conditions are

$$\nabla_w \mathcal{L} = 2w + X^T \eta = 0, \qquad \nabla_\eta \mathcal{L} = Xw - y = 0.$$

We have everything necessary to find the solution:

1. From first condition: $w = -X^T \eta / 2$
2. Plug into second condition: $\eta = -2(XX^T)^{-1} y$
3. Plug this back into #1: $w_{\text{ln}} = X^T (XX^T)^{-1} y$

Sparse ℓ_1 regression

LS and RR in high dimensions

Usually not suited for high-dimensional data

- Modern problems: Many dimensions/features/predictors
- Only a few of these may be important or relevant for predicting y
- Therefore, we need some form of "feature selection"

- Least squares and ridge regression:
  - Treat all dimensions equally without favoring subsets of dimensions
  - The relevant dimensions are averaged with irrelevant ones
  - Problems: Poor generalization to new data, interpretability of results

Regression with Penalties

Penalty terms

Recall: General ridge regression is of the form

$$\mathcal{L} = \sum_{i=1}^{n} (y_i - f(x_i; w))^2 + \lambda \|w\|^2$$

We've referred to the term $\|w\|^2$ as a *penalty term* and used $f(x_i; w) = x_i^T w$.

Penalized fitting

The general structure of the optimization problem is

$$\text{total cost} \;=\; \text{goodness-of-fit term} \;+\; \text{penalty term}$$

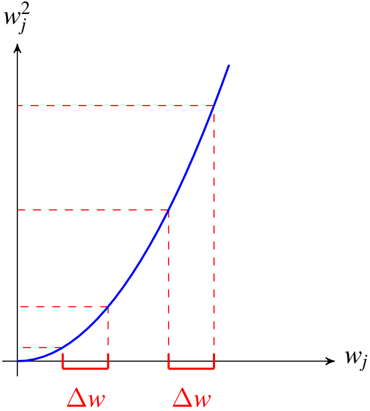
- Goodness-of-fit measures how well our model f approximates the data.
- Penalty term makes the solutions we don't want more "expensive".

What kind of solutions does the choice $\|w\|^2$ favor or discourage?

Quadratic Penalties

Intuitions

- Quadratic penalty: Reduction in cost depends on $|w_j|$.
- Suppose we reduce w_j by Δw. The effect on $\mathcal{L}$ depends on the starting point of w_j.
- Consequence: We should favor vectors w whose entries are of similar size, preferably small.



SPARSITY

Setting

- Regression problem with n data points $x \in \mathbb{R}^d$, $d \gg n$.
- Goal: Select a small subset of the d dimensions and switch off the rest.
- This is sometimes referred to as “feature selection”.

What does it mean to “switch off” a dimension?

- Each entry of w corresponds to a dimension of the data x.
- If $w_k = 0$, the prediction is

  $$f(x, w) = x^T w = w_1 x_1 + \cdots + 0 \cdot x_k + \cdots + w_d x_d ,$$

  so the prediction does not depend on the kth dimension.
- Feature selection: Find a w that (1) predicts well, and (2) has only a small number of non-zero entries.
- A w for which most dimensions $= 0$ is called a *sparse* solution.

Sparsity and Penalties

Penalty goal

Find a penalty term which encourages sparse solutions.

Quadratic penalty vs sparsity

- Suppose w_k is large, all other w_j are very small but non-zero
- Sparsity: Penalty should keep w_k, and push other w_j to zero
- Quadratic penalty: Will favor entries w_j which all have similar size, and so it will push w_k towards small value.

Overall, a quadratic penalty favors many small, but non-zero values.

Solution

Sparsity can be achieved using *linear* penalty terms.

LASSO

Sparse regression

LASSO: Least Absolute Shrinkage and Selection Operator

With the LASSO, we replace the ℓ_2 penalty with an ℓ_1 penalty:

$$w_{\text{lasso}} = \arg\min_{w} \|y - Xw\|_2^2 + \lambda \|w\|_1$$

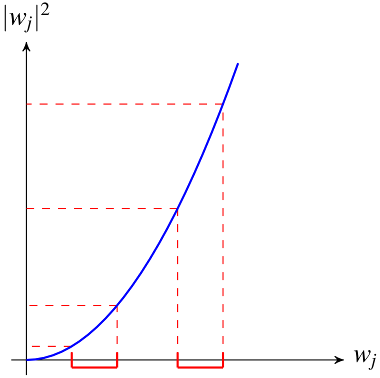
where

$$\|w\|_1 = \sum_{j=1}^{d} |w_j|.$$

This is also called ℓ_1-regularized regression.

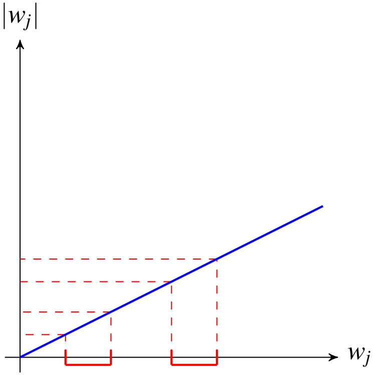
QUADRATIC PENALTIES

Quadratic penalty



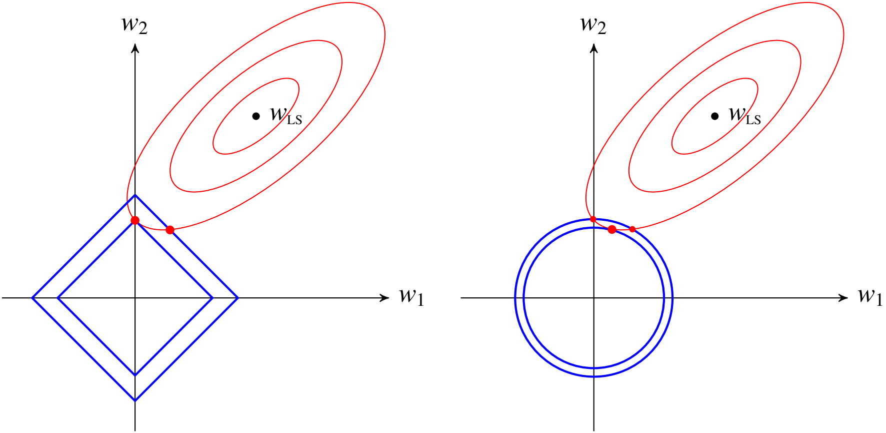
Reducing a large value w_j achieves a larger cost reduction.

Linear penalty



Cost reduction does not depend on the magnitude of w_j.

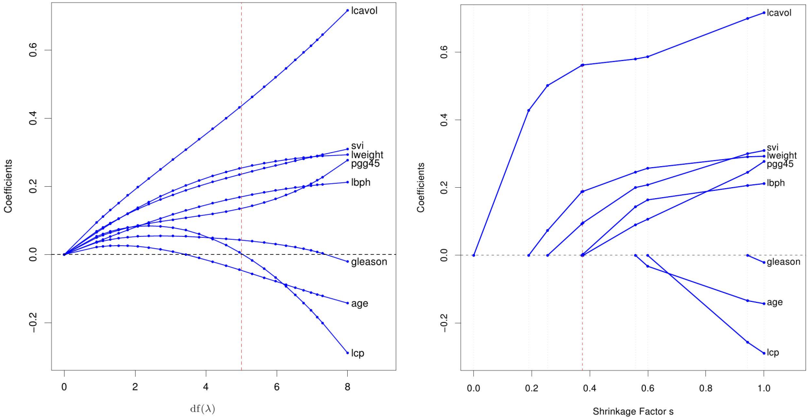
Ridge Regression vs LASSO



This figure applies to $d < n$, but gives intuition for $d \gg n$.

- Red: Contours of $(w - w_{\text{LS}})^T (X^T X)(w - w_{\text{LS}})$ (see Lecture 3)
- Blue: (left) Contours of $\|w\|_1$, and (right) contours of $\|w\|_2^2$

Coefficient profiles: RR vs LASSO



(a) $||w||_2$ penalty

(b) $||w||_1$ penalty

ℓ_p Regression

ℓ_p-norms

These norm-penalties can be extended to all norms:

$$\|w\|_p = \Bigl(\sum_{j=1}^{d} |w_j|^p\Bigr)^{\frac{1}{p}} \qquad \text{for } 0 < p \leq \infty$$

ℓ_p-regression

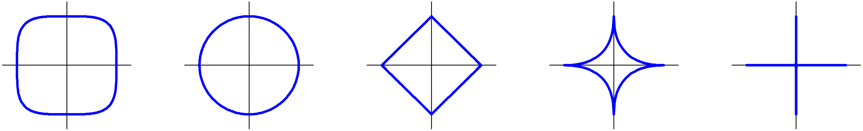
The ℓ_p-regularized linear regression problem is

$$w_{\ell_p} := \arg\min_{w} \; \|y - Xw\|_2^2 + \lambda\|w\|_p^p$$

We have seen:

- ℓ_1-regression = LASSO
- ℓ_2-regression = ridge regression

ℓ_p Penalization Terms



$p = 4$ $p = 2$ $p = 1$ $p = 0.5$ $p = 0.1$

p	Behavior of $\|\| \, . \, \|\|_p$
$p = \infty$	Norm measures largest absolute entry, $\|w\|_\infty = \max_j \|w_j\|$
$p > 2$	Norm focuses on large entries
$p = 2$	Large entries are expensive; encourages similar-size entries
$p = 1$	Encourages sparsity
$p < 1$	Encourages sparsity as for $p = 1$, but contour set is not convex (i.e., no “line of sight” between every two points inside the shape)
$p \to 0$	Simply records whether an entry is non-zero, i.e. $\|w\|_0 = \sum_j \mathbb{I}\{w_j \neq 0\}$

COMPUTING THE SOLUTION FOR ℓ_p

Solution of ℓ_p problem

- $\boldsymbol{\ell_2}$ aka ridge regression. Has a closed form solution
- $\boldsymbol{\ell_p}$ $(\boldsymbol{p \geq 1, p \neq 2})$ — By "convex optimization". We won't discuss convex analysis in detail in this class, but two facts are important
  - There are no "local optimal solutions" (i.e., local minimum of $\mathcal{L}$)
  - The true solution can be found *exactly* using iterative algorithms

  $(\boldsymbol{p < 1})$ — We can only find an approximate solution (i.e., the best in its "neighborhood") using iterative algorithms.

Three techniques formulated as optimization problems

Method	Good-o-fit	penalty	Solution method
Least squares	$\|\|y - Xw\|\|_2^2$	none	Analytic solution exists if $X^T X$ invertible
Ridge regression	$\|\|y - Xw\|\|_2^2$	$\|\|w\|\|_2^2$	Analytic solution exists always
LASSO	$\|\|y - Xw\|\|_2^2$	$\|\|w\|\|_1$	Numerical optimization to find solution