Modern Regression

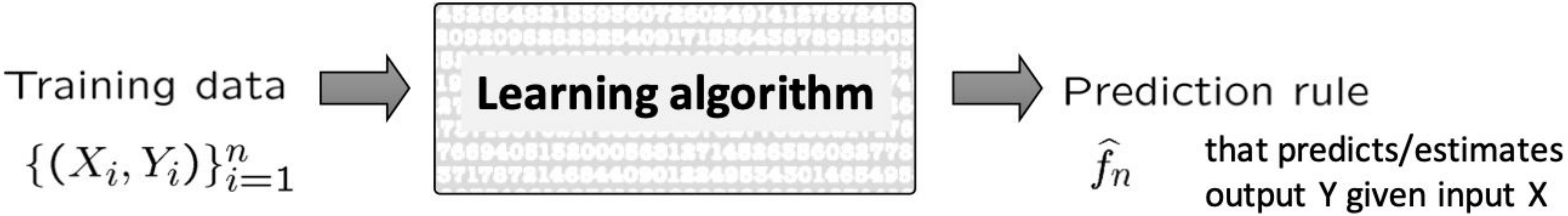
Lecture 7

Xiaofei Shi

Learning objective:

- Modern methods for linear regression and regression with regulation penalty

Regression



- Linear Regression — last time
- Regularized Linear Regression – Ridge regression, Lasso Polynomial Regression — today
- Gaussian Process Regression

Recap on linear regression

- Build your model:

  1) relationship: $y = \sum_{j=0}^{k} w_j \phi_j(x)$

  2) preference: choose w to minimize $J(\mathbf{w}) = \sum_i (y^i - \sum_j w_j \phi_j(x^i))^2$

- Estimate your model parameters:

  1) plugging in observed data to express your preference

  2) get parameters estimation for your model $\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$

- Understand your model

Potential problems:

- collinearity
- too many non-zero but very small coefficients
- too slow

Regularizer: ridge regression!

- If $\Phi^T\Phi$ is not invertible, or its determinant is very small, the optimal w is not going to be stable
- n equations < p unknowns – underdetermined system of linear equations many feasible solutions

Need to impose extra constraints!

Regularizer: ridge regression!

- If $\Phi^T\Phi$ is not invertible, or its determinant is very small, the optimal w is not going to be stable
- n equations < k unknowns – underdetermined system of linear equations many feasible solutions
- Adding in penalty term into loss function:

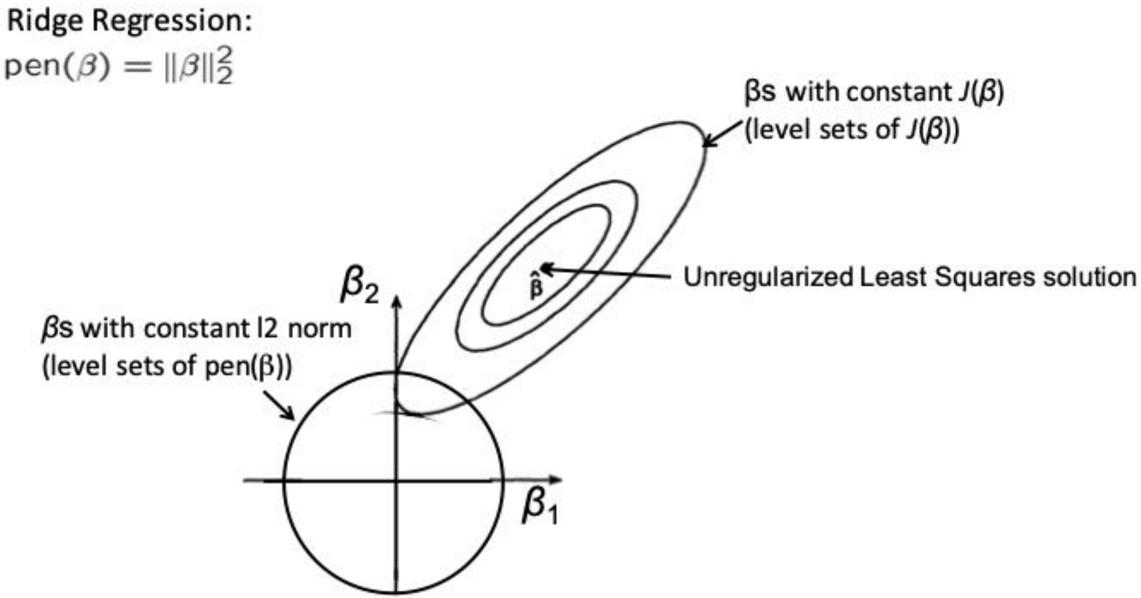
$$J(\beta) = \sum_i \left(y^i - \sum_j \beta_j \phi_j(x^i)\right)^2 + \lambda \sum_j \beta_j^2$$
$$= \|y - \Phi(x)\beta\|_2^2 + \lambda\|\beta\|_2^2$$

different norms of matrix and vectors

- Equivalent to a MAP optimization problem → regression coefficient with Gaussian prior!

$$\widehat{\beta} = (\Phi^\top(x)\Phi(x) + \lambda I)^{-1}\Phi^\top(x)y$$

Regularizer: ridge regression!



Regularizer: Lasso

- n equations < k unknowns – underdetermined system of linear equations many feasible solutions
- Sometimes our goal is to learn a ***sparse*** representation: select the most useful features!
- How to achieve?

Regularizer: Lasso

- n equations < k unknowns – underdetermined system of linear equations many feasible solutions
- Sometimes our goal is to learn a ***sparse*** representation: select the most useful features!
- How to achieve?

$$J(\beta) = \|y - \Phi(x)\beta\|_2^2 + \lambda\|\beta\|_0$$

No closed form!
Hard to solve!

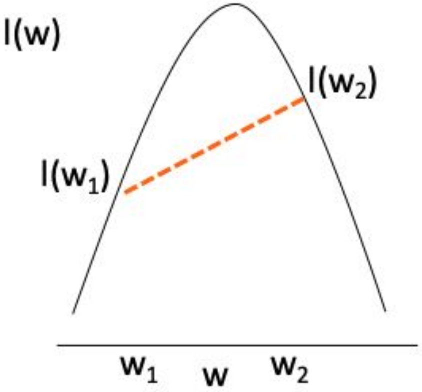
Regularizer: Lasso

- n equations < k unknowns – underdetermined system of linear equations many feasible solutions
- Sometimes our goal is to learn a ***sparse*** representation: select the most useful features!
- How to achieve?

$$J(\beta) = ||y - \Phi(x)\beta||_2^2 + \lambda||\beta||_1$$

No closed form!
Getting easier!

Difference: Convex optimization!



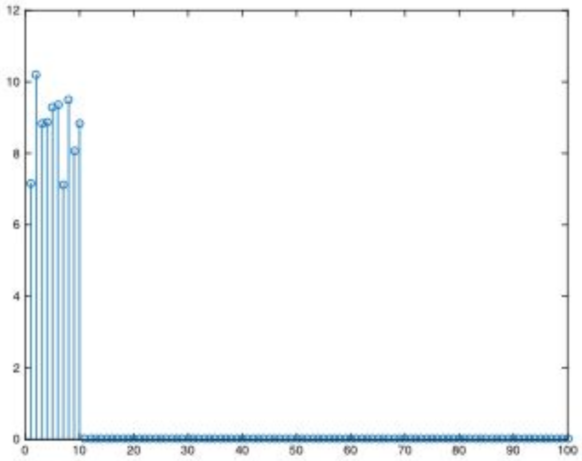
A function $l(w)$ is called **concave** if the line joining two points $l(w_1)$,$l(w_2)$ on the function does not go above the function on the interval $[w_1,w_2]$

(Strictly) Concave functions have a unique maximum!

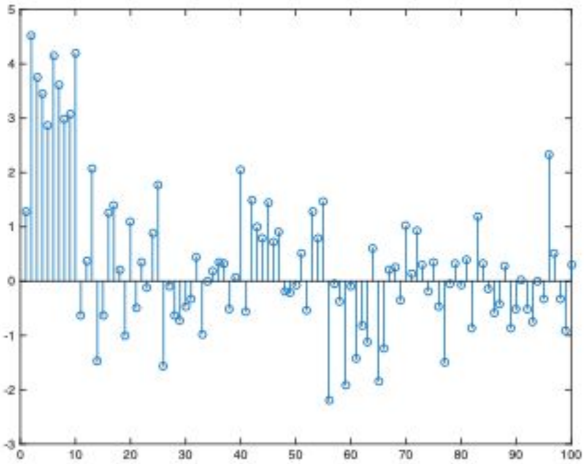
Convex | Both Concave & Convex | Neither

Lasso or ridge? It's a question...

Lasso Coefficients



Ridge Coefficients

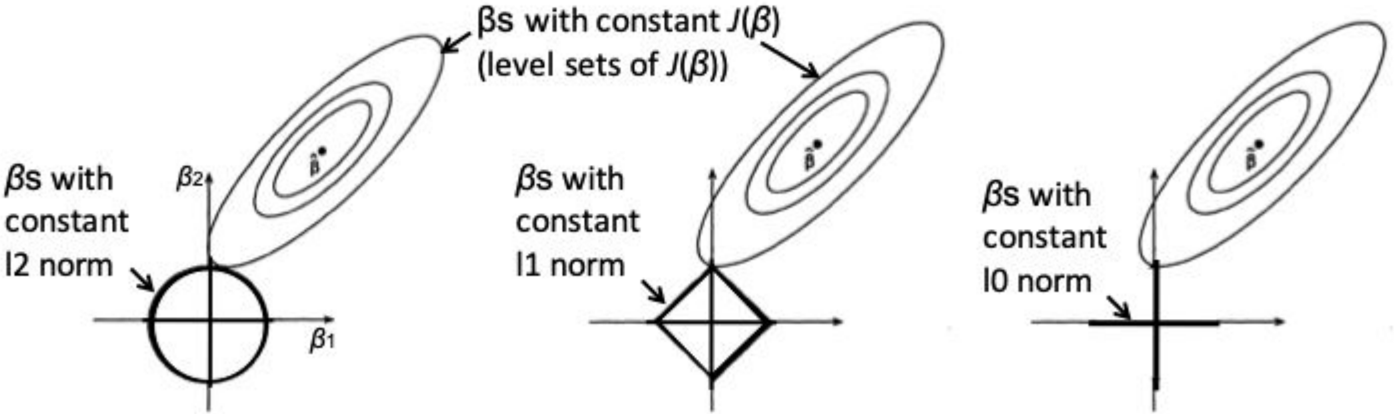


What did they actually do...

Ridge Regression:
$\text{pen}(\beta) = \|\beta\|_2^2$

Lasso:
$\text{pen}(\beta) = \|\beta\|_1$

Ideally l0 penalty,
but optimization
becomes non-convex



Lasso (l1 penalty) results in sparse solutions – vector with more zero coordinates
Good for high-dimensional problems – don't have to store all coordinates,
interpretable solution!

Regression to classification

- Instead of giving scores to these apps, can you tell which app to use?
- Can we predict the “probability” of class label – a real number – using regression methods?
- But output (probability) needs to be in [0,1]

A way to make categorical variables continuous!

	熊猫外卖 HungryPanda	Uber Eats	chowbus
Available restaurants	30	10	20
Average delivery time	Next day	>3hr	1hr
Mandatory service fee	>10%	>20%	>13%
Score	9	7	8

Columbia University

Logistic regression

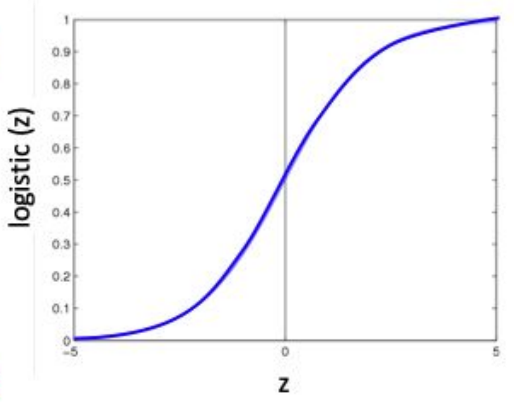
- Instead of modeling Y =0 or 1 directly, we modify the probability of P(Y=0|x) as

$$P(Y=0|X) = \frac{1}{1+\exp(w_0 + \sum_i w_i X_i)}$$

Logistic function applied to a linear function of the data

Logistic function (or Sigmoid): $\frac{1}{1+exp(-z)}$

Features can be discrete or continuous!



Logistic regression for 2 categories

Assumes the following functional form for P(Y|X):

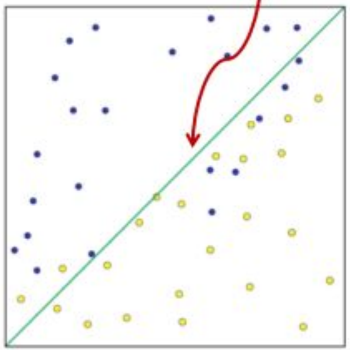
$$P(Y = 0|X) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

$$w_0 + \sum_i w_i X_i = 0$$

Decision boundary: Note - Labels are 0,1

$$P(Y = 0|X) \overset{0}{\underset{1}{\gtrless}} P(Y = 1|X)$$

$$w_0 + \sum_i w_i X_i \overset{1}{\underset{0}{\gtrless}} 0$$



(Linear Decision Boundary)

Logistic regression for K categories

- Logistic regression in more general case, where $Y \in \{y_1,\ldots,y_K\}$

for *k<K*

$$P(Y = y_k|X) = \frac{\exp(w_{k0} + \sum_{i=1}^{d} w_{ki}X_i)}{1 + \sum_{j=1}^{K-1} \exp(w_{j0} + \sum_{i=1}^{d} w_{ji}X_i)}$$

for *k=K* (normalization, so no weights for this class)

Are decision boundaries still linear? Why?

$$P(Y = y_K|X) = \frac{1}{1 + \sum_{j=1}^{K-1} \exp(w_{j0} + \sum_{i=1}^{d} w_{ji}X_i)}$$

Predict $f^*(x) = \arg\max_{Y=y} P(Y = y|X = x)$