Regression: Modern regression

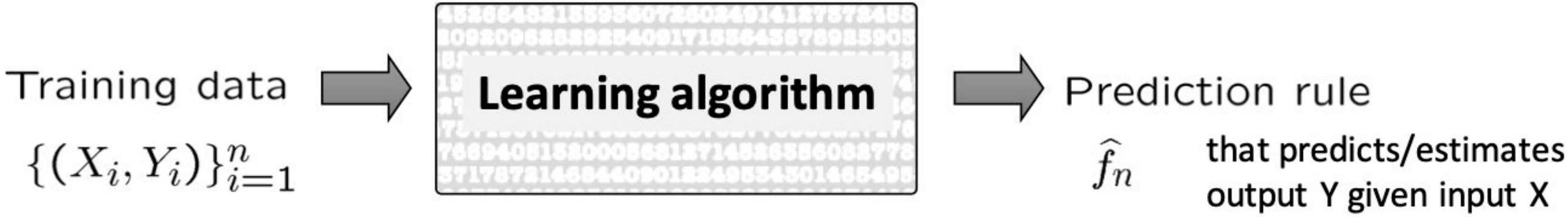
GU 4241

Statistical Machine Learning

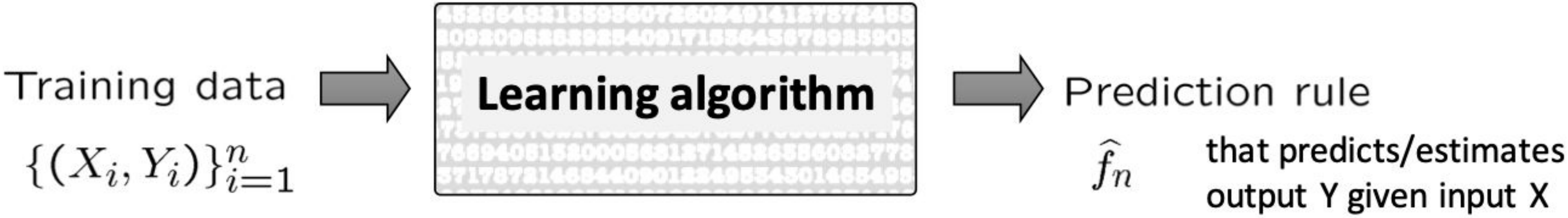
Xiaofei Shi

Columbia University

Regression:


Training data
$\{(X_i, Y_i)\}_{i=1}^{n}$
Learning algorithm
Prediction rule
$\widehat{f}_n$
that predicts/estimates output Y given input X

Regression:



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

Recap of linear regression:

- Build your model:

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

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

- 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

Recap of 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

Columbia University

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 → HW1

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

COLUMBIA UNIVERSITY

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:

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

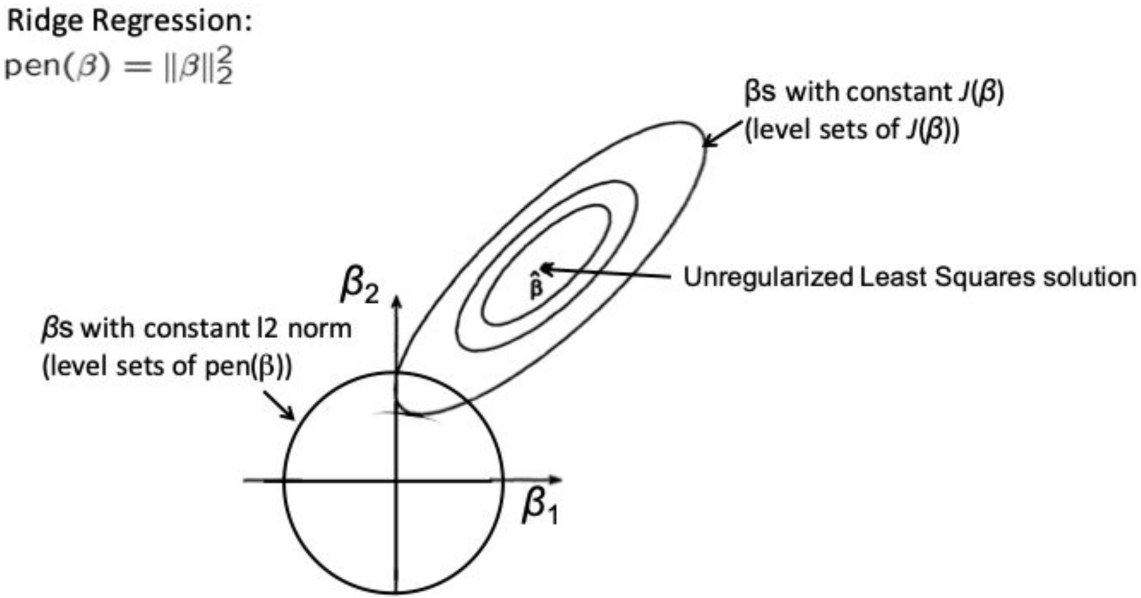
different norms of matrix and vectors

- Equivalent to a MAP optimization problem $\longrightarrow$ HW1

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

- Don't have to worry about invertibility anymore!

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!

Columbia University

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?

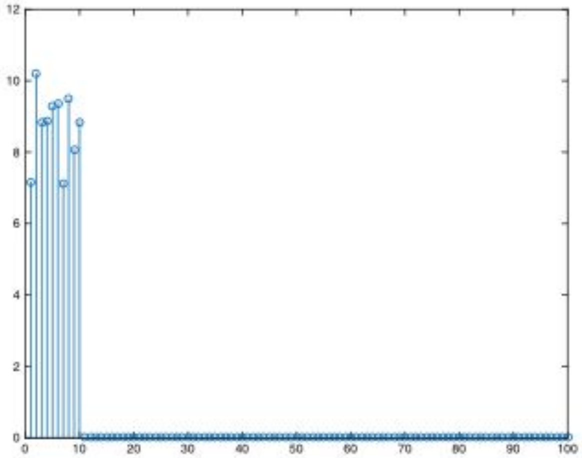
$$J(\beta) = ||y - \Phi(x)\beta||_2^2 + \lambda||\beta||_1$$

No closed form!
Getting easier!

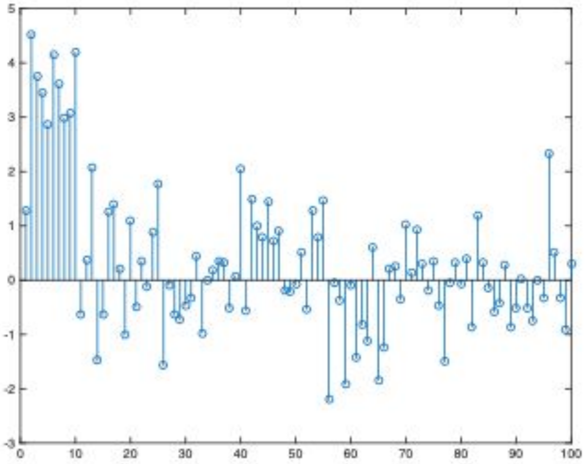
COLUMBIA UNIVERSITY

Lasso or Ridge?

Lasso Coefficients



Ridge Coefficients

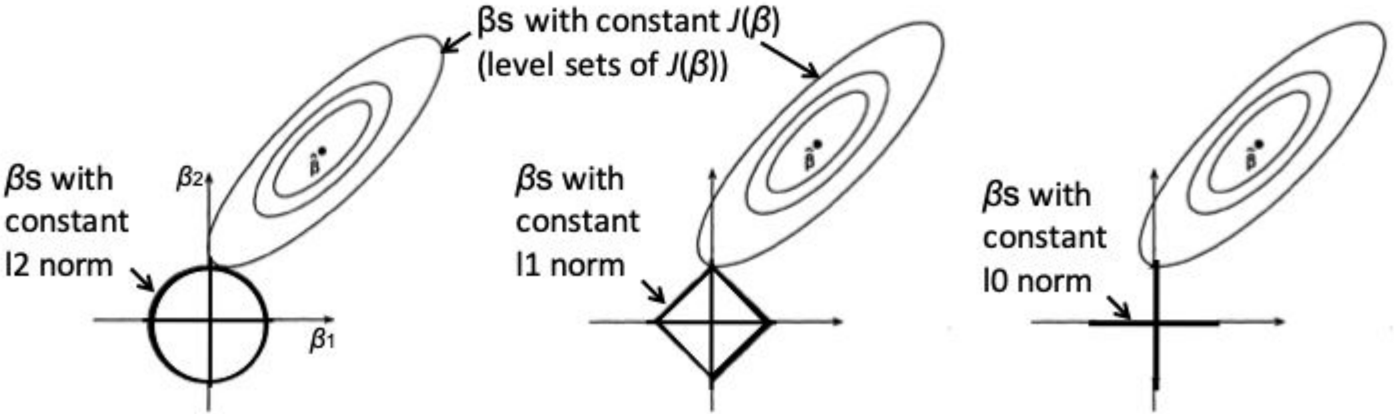


COLUMBIA UNIVERSITY

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!

COLUMBIA UNIVERSITY

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

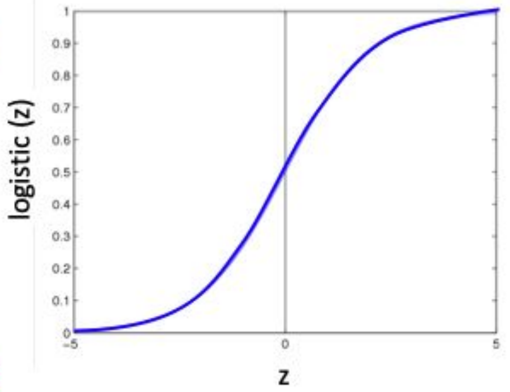
- Instead of modeling Y =0 or 1 directly, we modify the probability of P(Y=0|x) as

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

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!



Columbia University

2 categories

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

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

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

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

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

COLUMBIA UNIVERSITY

2 categories

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

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

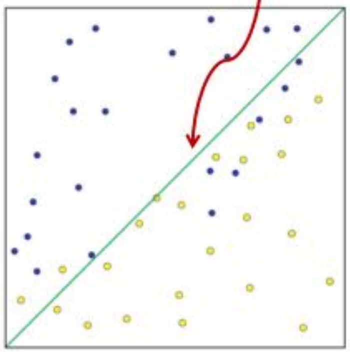
$$w_0 + \sum_i w_i X_i = 0$$

Decision boundary: Note - Labels are 0,1

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

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

(Linear Decision Boundary)



Expressing conditional likelihood

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

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

$$l(\mathbf{w}) = \sum_j \left[y^j (w_0 + \sum_i^d w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^d w_i x_i^j)) \right]$$

Expressing conditional likelihood

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

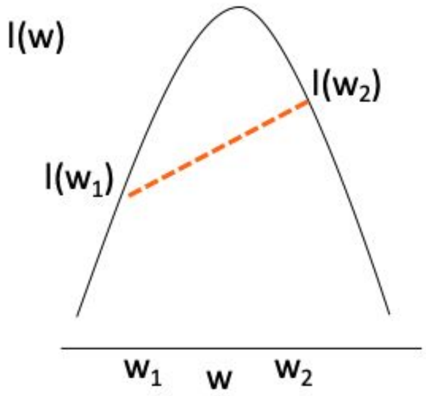
$$P(Y=1|\mathbf{X},\mathbf{w}) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

$$\Longrightarrow$$

$$l(\mathbf{w}) = \sum_j \left[y^j (w_0 + \sum_i^d w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^d w_i x_i^j)) \right]$$

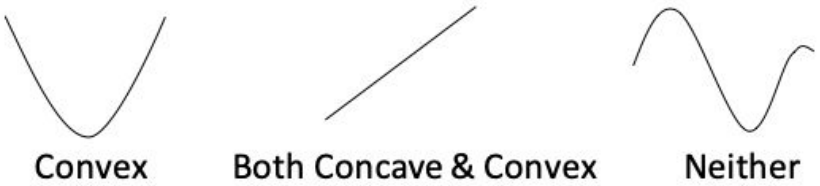
- Bad: we cannot find explicit solution anymore
- Good: it is guaranteed to have a unique solution, and we can still solve this problem numerically

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 optimization for logistic regression

Gradient ascent rule for w_0:

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \left. \frac{\partial l(\mathbf{w})}{\partial w_0} \right|_t$$

$$l(\mathbf{w}) = \sum_j \left[y^j (w_0 + \sum_i^d w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^d w_i x_i^j)) \right]$$

$$\frac{\partial l(\mathbf{w})}{\partial w_0} = \sum_j \left[y^j - \underbrace{\frac{1}{1 + exp(w_0 + \sum_i^d w_i x_i^j)} \cdot exp(w_0 + \sum_i^d w_i x_i^j)}_{} \right]$$

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]$$

Columbia University

Convex optimization for logistic regression

Gradient ascent algorithm: iterate until change < ε

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]$$

For i=1,…,d,

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \underbrace{\hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})}_{\text{Predict what current weight thinks label Y should be}}]$$

repeat

- Gradient ascent is simplest of optimization approaches
  - e.g., Newton method, Conjugate gradient ascent, IRLS (see Bishop 4.3.3)

More than 2 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)

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

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

More than 2 categories

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

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

Columbia University

References

- Christopher Bishop: Pattern Recognition and Machine Learning, Chapter 4
- Kutner, Nachtsheim and Neter: Applied Linear Regression Models.
- Agresti: Foundations of Linear and Generalized Linear Models.
- Ziv Bar-Joseph, Pradeep Ravikumar and Aarti Singh: CMU 10-701