COLUMBIA UNIVERSITY

Modern Regression

STAT5241 Section 2

Statistical Machine Learning

Xiaofei Shi

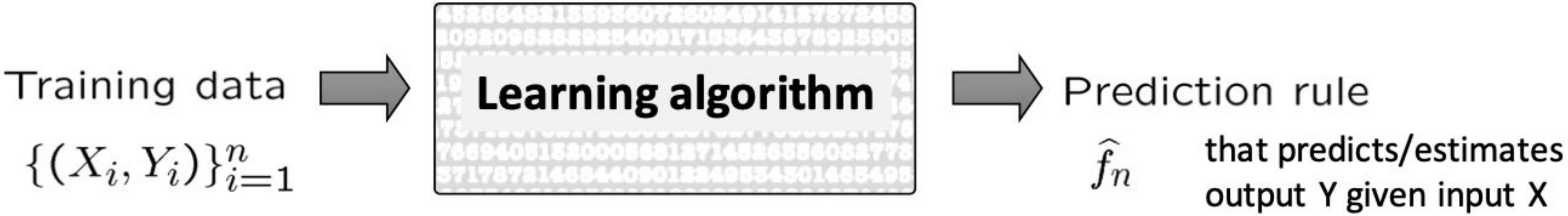
Tasks for supervised learning

Input $\longrightarrow$ Regressor $\longrightarrow$ Predict real number

Input $\longrightarrow$ Classifier $\longrightarrow$ Predict category

Columbia University

Regression:



COLUMBIA UNIVERSITY

Regression:

- In everyday life we need to make decisions by taking into account lots of factors
- The question is what weight we put on each of these factors (how important are they with respect to the others)
- Suppose your task is to help build a evaluation system for food delivery apps

	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

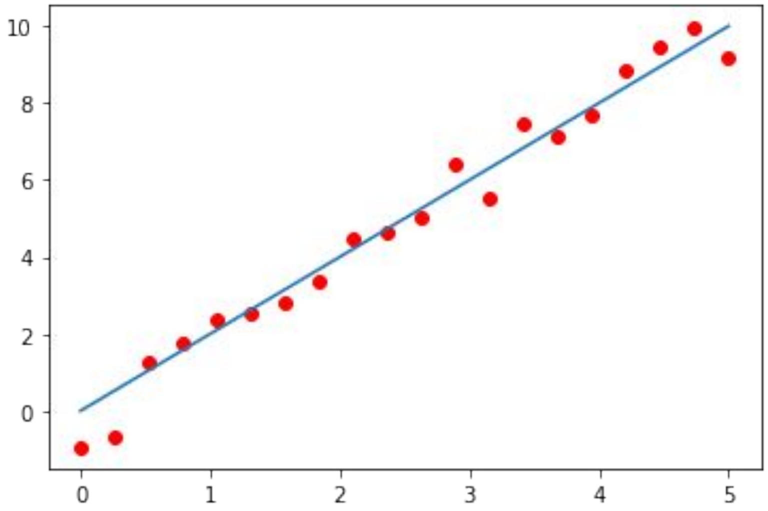
Columbia University

Regression:

- Given an input x, we want to compute an output y
- For example:

  - predict Google's stock price using the current price of Bitcoin

  - predict arrival time using the traffic condition



Linear Regression:

- Given an input x, we want to compute an output y
- In linear regression we assume that y and x are related with the following

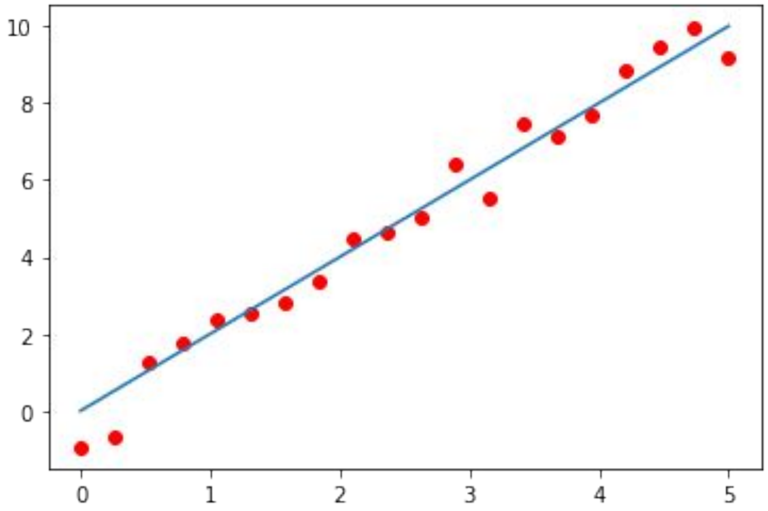
What we are trying to predict

Observed values

$$y = wx+\varepsilon$$

parameter we want to determine

noise term



Columbia University

Road map

- Build your model:

  1) relationship: $y = wx+\varepsilon$

  2) preference: choose w to minimize $\arg\min_w \sum_i (y_i - wx_i)^2$

- Estimate your model parameters:

  1) plugging in observed data to express your preference

  2) get parameters estimation for your model

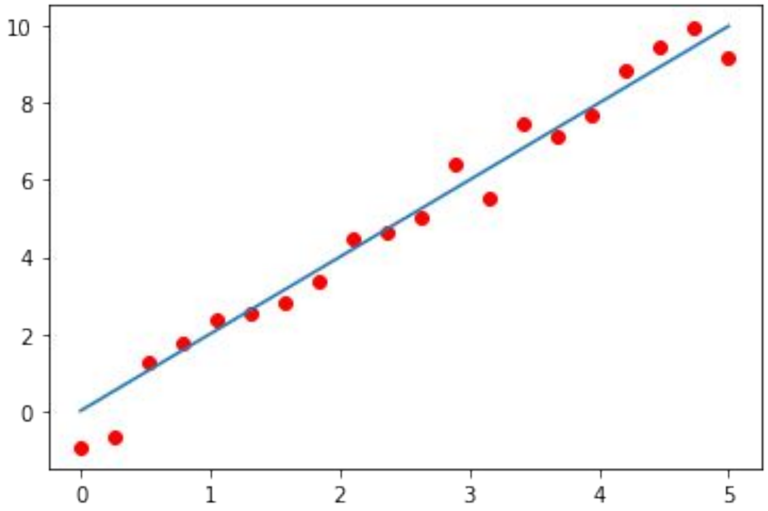
- Understand your model:

Linear Regression:

- Our goal is to estimate w from a training data of <xi,yi> pairs
- One easy way to determine w is to minimize the least squares error:

$$\arg\min_w \sum_i (y_i - wx_i)^2$$

- Why least squares?
  - easy to compute
  - has a nice probabilistic interpretation
- Several other approaches



If the noise is Gaussian with mean 0 then least squares is also the MLE of w

COLUMBIA UNIVERSITY

Solving linear regression using minimization

- Goal function:
- 3-step:
  - take derivative wrt parameter w
  - set it to 0
  - solve optimal w from the equation

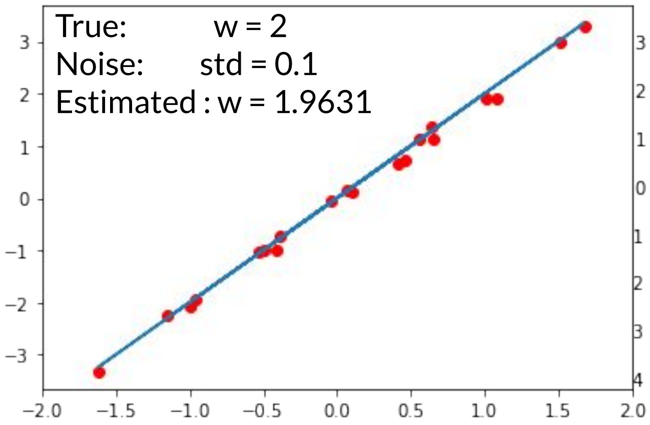
$$\frac{\partial}{\partial w}\sum_i (y_i - wx_i)^2 = 2\sum_i -x_i(y_i - wx_i) \Rightarrow$$

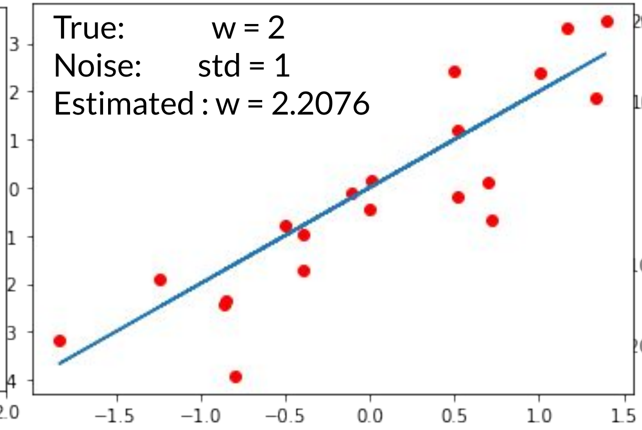
$$2\sum_i x_i(y_i - wx_i) = 0 \Rightarrow$$

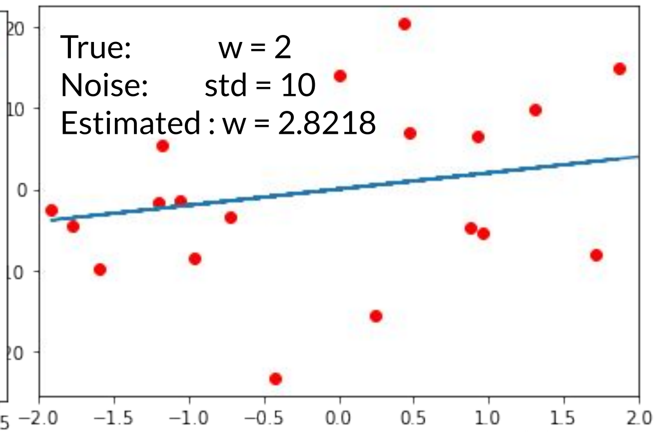
$$\sum_i x_i y_i = \sum_i w x_i^2 \Rightarrow$$

$$w = \frac{\sum_i x_i y_i}{\sum_i x_i^2}$$

Regression Example







COLUMBIA UNIVERSITY

Adding in intercept

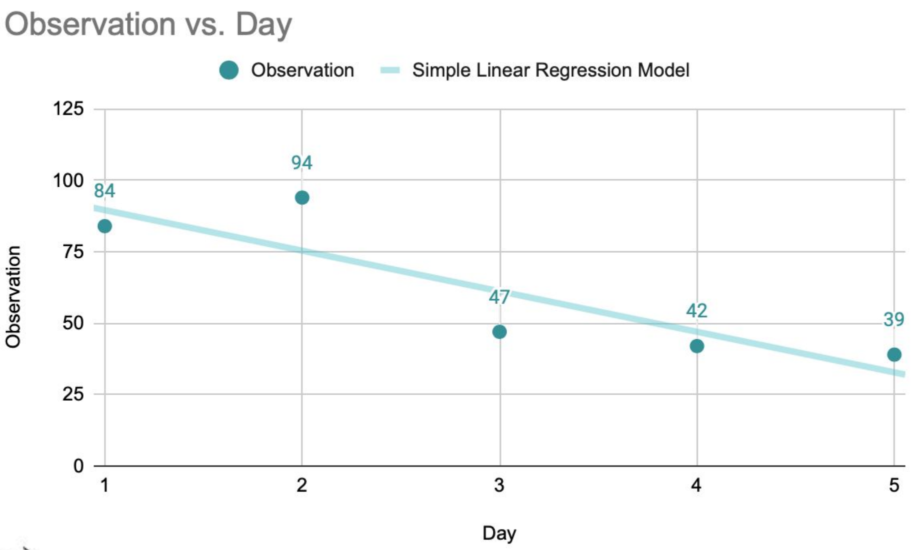
- So far we assume the regression line passes through the origin
- What if the line does not?

$$y = w_0 + w_1 x + \varepsilon$$

Adding in intercept!

- We can determine w = (w0 , w1) explicitly

$$w_0 = \frac{\sum_i y_i - w_1 x_i}{n} \qquad w_1 = \frac{\sum_i x_i (y_i - w_0)}{\sum_i x_i^2}$$



Columbia University

Multivariate Regression:

- What if we want to input more information?
- For example:

  - predict Google's stock price using the current price of Bitcoin might not be enough, we might want to also consider the stock price of Amazon, Apple, and other index

  - predict arrival time using the traffic condition, the weather condition, and the vehicle's condition to make it more accurate
- This becomes a multivariate linear regression problem:

$$y = w_0 + w_1x_1 + \dots + w_kx_k + \varepsilon$$

Google's price Bitcoin S&P 500

Multivariate Regression:

Not all functions can be represented using the input values directly!
How to capture nonlinearity?

- What if we want to input more information?
- For example:

  - predict Google's stock price using the current price of Bitcoin might not be enough, we might want to also consider the stock price of Amazon, Apple, and other index

  - predict arrival time using the traffic condition, the weather condition, and the vehicle's condition to make it more accurate
- This becomes a multivariate linear regression problem:

$$y = w_0 + w_1 x_1 + \dots + w_k x_k + \varepsilon$$

Google's price   Bitcoin   S&P 500

Columbia University

How to capture nonlinearity?

- In some cases we would like to use polynomial or other terms based on the input data

  - Polynomial: $\phi_j(x) = x^j$ for $j=0 \dots n$

  - Gaussian: $\phi_j(x) = \dfrac{(x-\mu_j)}{2\sigma_j^2}$

  - Sigmoid: $\phi_j(x) = \dfrac{1}{1+\exp(-s_j x)}$

- Are these still linear regression problems?

COLUMBIA UNIVERSITY

How to capture nonlinearity?

- In some cases we would like to use polynomial or other terms based on the input data

  - Polynomial: $\phi_j(x) = x^j$ for j=0 … n

  - Gaussian: $\phi_j(x) = \frac{(x-\mu_j)}{2\sigma_j^2}$

  - Sigmoid: $\phi_j(x) = \frac{1}{1+\exp(-s_j x)}$

- Are these still linear regression problems?

As long as the coefficients are linear the equation is still a linear regression problem!

COLUMBIA UNIVERSITY

Nonlinear basis functions

- In some cases we would like to use polynomial or other terms based on the input data

  - Polynomial: $\phi_j(x) = x^j$ for $j=0 \ldots n$

  - Gaussian: $\phi_j(x) = \frac{(x-\mu_j)}{2\sigma_j^2}$

  - Sigmoid: $\phi_j(x) = \frac{1}{1+\exp(-s_j x)}$

Any function of the input values can be used. The solution for the parameters of the regression remains the same.

- Linear regression can be applied in the same way to functions of these values
- As long as these functions can be directly computed from the observed values the parameters are still linear in the data and the problem remains a linear regression problem

Columbia University

Nonlinear basis functions

- We use the new notation for the basis functions, linear regression can be written as

$$y = \sum_{j=0}^{n} w_j \phi_j(x)$$

- Nothing changed! Once again we can use 'least squares' to find the optimal solution to figure out parameter w

General linear regression problem

$$y = \sum_{j=0}^{K} w_j \phi_j(x)$$

Our goal is to minimize the following loss function:

$$J(\mathrm{w}) = \sum_i (y^i - \sum_j w_j \phi_j(x^i))^2$$

w – vector of dimension k+1

$\phi(x^i)$ – vector of dimension k+1

y^i – a scaler

Moving to vector notations we get:

$$J(\mathrm{w}) = \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i))^2$$

We take the derivative w.r.t **w**

$$\frac{\partial}{\partial w} \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i))^2 = 2 \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i)) \phi(x^i)^{\mathrm{T}}$$

Equating to 0 we get $\quad 2 \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i)) \phi(x^i)^{\mathrm{T}} = 0 \Rightarrow$

$$\sum_i y^i \phi(x^i)^{\mathrm{T}} = \mathrm{w}^{\mathrm{T}} \left[\sum_i \phi(x^i) \phi(x^i)^{\mathrm{T}} \right]$$

General linear regression problem

$$J(\mathrm{w}) = \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i))^2$$

We take the derivative w.r.t **w**

$$\frac{\partial}{\partial w} \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i))^2 = 2 \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i)) \phi(x^i)^{\mathrm{T}}$$

Equating to 0 we get

$$2 \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i)) \phi(x^i)^{\mathrm{T}} = 0 \Rightarrow$$

$$\sum_i y^i \phi(x^i)^{\mathrm{T}} = \mathrm{w}^{\mathrm{T}} \left[\sum_i \phi(x^i) \phi(x^i)^{\mathrm{T}} \right]$$

Define:

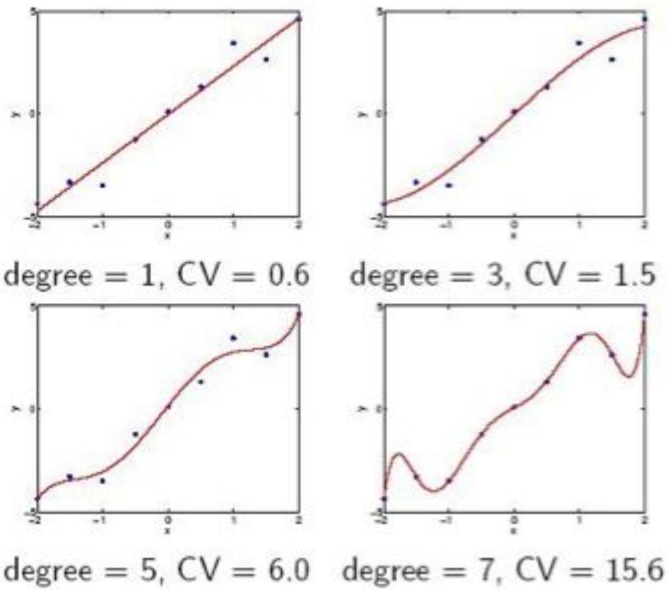
$$\Phi = \begin{pmatrix} \phi_0(x^1) & \phi_1(x^1) & \cdots & \phi_k(x^1) \\ \phi_0(x^2) & \phi_1(x^2) & \cdots & \phi_k(x^2) \\ \vdots & \vdots & \cdots & \vdots \\ \phi_0(x^n) & \phi_1(x^n) & \cdots & \phi_k(x^n) \end{pmatrix}$$

Then deriving w we get:

$$\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$$

COLUMBIA UNIVERSITY

Example: polynomial regression



Thoughts？

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!

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

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

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 → HW1

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

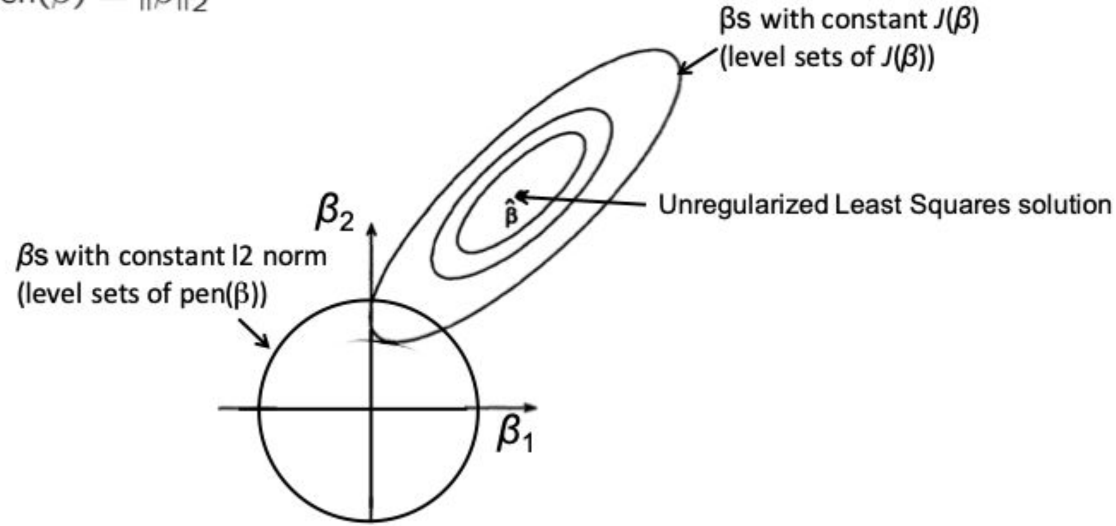
- Don't have to worry about invertibility anymore!

COLUMBIA UNIVERSITY

Regularizer: ridge regression

Ridge Regression:

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



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?

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?

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

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?

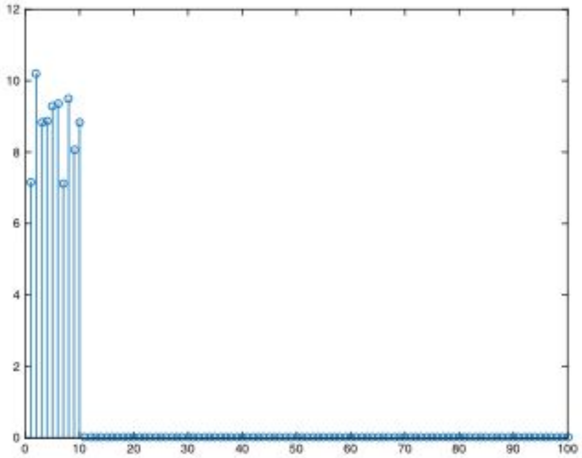
$$J(\beta) = ||y - \Phi(x)\beta||_2^2 + \lambda||\beta||_1$$

No closed form!
Getting easier!

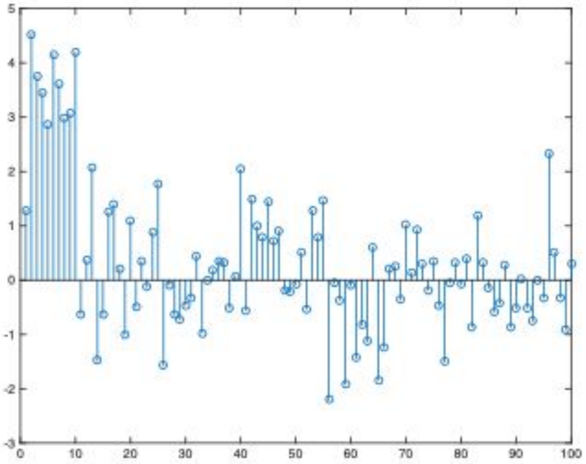
Columbia University

Lasso or Ridge?

Lasso Coefficients



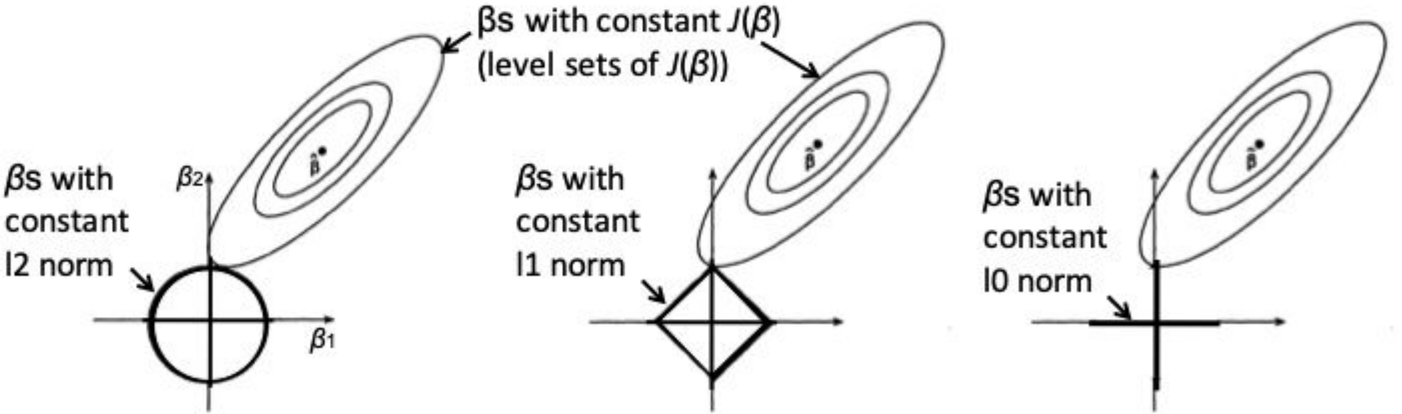
Ridge Coefficients



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

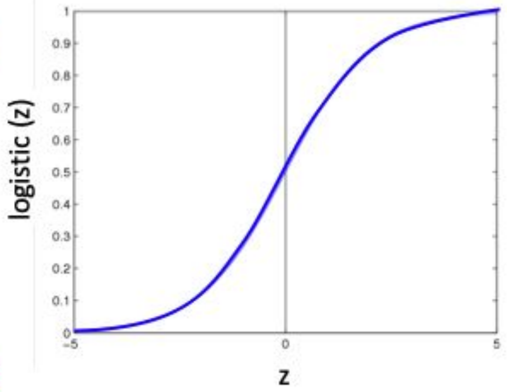
Logistic regression

- 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!

2 categories

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

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

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

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

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

2 categories

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

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

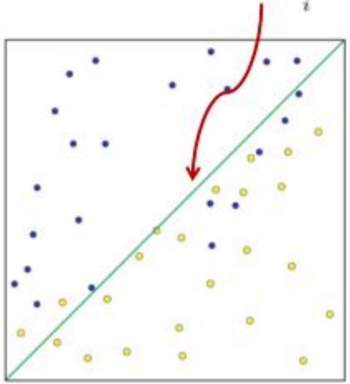
Decision boundary: Note - Labels are 0,1

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

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

(Linear Decision Boundary)

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



COLUMBIA UNIVERSITY

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

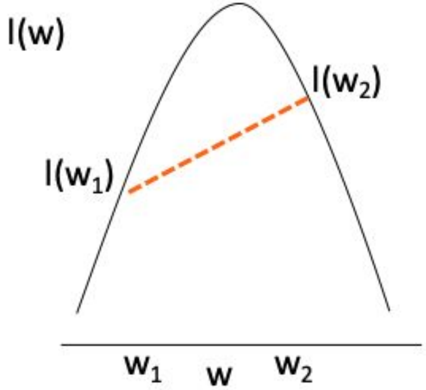
$$P(Y=0|\mathbf{X},\mathbf{w}) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

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

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

- 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

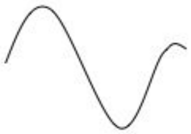


A function l(w) is called **concave** if the line joining two points l(w_1),l(w_2) on the function does not go above the function on the interval [w_1,w_2]

(Strictly) Concave functions have a unique maximum!

Convex

Both Concave & Convex



Neither

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)

Columbia University

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

COLUMBIA UNIVERSITY

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)

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 3, 4
- Kutner, Nachtsheim and Neter: Applied Linear Regression Models.
- Agresti: Foundations of Linear and Generalized Linear Models.
- Ziv Bar-Joseph, Tom Mitchell, Pradeep Ravikumar and Aarti Singh: CMU 10-701

Columbia University