Logistic Regression (continued), Linear regression

Sriram Sankararaman

The instructor gratefully acknowledges Fei Sha, Ameet Talwalkar, Eric Eaton, and Jessica Wu whose slides are heavily used, and the many others who made their course material freely available online.

Announcements

- Problem set 2 has been released.
  - Due on Feb 8.
  - Please start early!

Outline

1. Logistic regression
   - Optimization
2. Linear regression

Logistic classification

Setup for binary classification

- Input: $\boldsymbol{x} \in \mathbb{R}^D$
- Output: $y \in \{0, 1\}$
- Training data: $\mathcal{D} = \{(\boldsymbol{x}_n, y_n), n = 1, 2, \ldots, N\}$
- Hypotheses/Model:

$$h_{\boldsymbol{w},b}(x) = p(y = 1|\boldsymbol{x}; b, \boldsymbol{w}) = \sigma(a(\boldsymbol{x}))$$

  where

$$a(\boldsymbol{x}) = b + \sum_d w_d x_d = b + \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}$$

- Given training data N samples/instances: $\mathcal{D}^{\mathrm{TRAIN}} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_{\mathsf{N}}, y_{\mathsf{N}})\}$, train/learn/induce $h_{\boldsymbol{w},b}$. Find values for $(\boldsymbol{w}, b)$.

How to find the optimal parameters for logistic regression?

We will minimize the negative log likelihood

$$J(\boldsymbol{\theta}) = -\sum_n \{y_n \log h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) + (1 - y_n) \log[1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}_n)]\}$$

- $\boldsymbol{\theta} = [\theta_0 \ \theta_1 \ \cdots \ \theta_D]^{\mathrm{T}} = [b \ w_1 \ w_2 \ \cdots \ w_{\mathsf{D}}]^{\mathrm{T}}$
- $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sigma(\theta_0 + \sum_d \theta_d x_d) = \sigma(b + \sum_d w_d x_d)$

Optimization

Given a function $f(x)$, find its minimum (or maximum).

- f is called the objective function.
- Maximizing f is equivalent to minimizing $-f$.

  So we only need to consider minimization problems.

Optimization

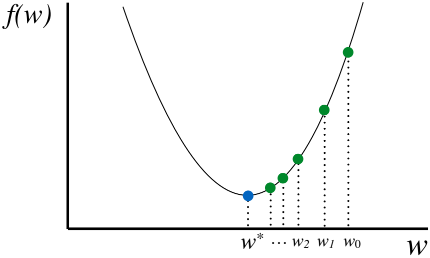
Given a function $f(x)$, find its minimum (or maximum).

- f is called the objective function.
- Maximizing f is equivalent to minimizing $-f$.

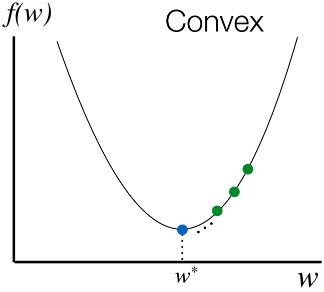
  So we only need to consider minimization problems.
- One way to minimize f is gradient descent.

Gradient Descent

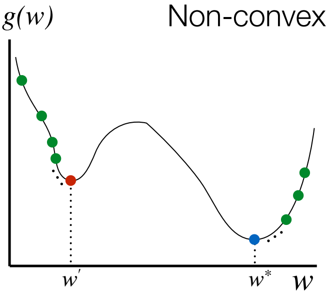
Start at a random point
Repeat
    Determine a descent direction
    Choose a step size
    Update
Until stopping criterion is satisfied



Where Will We Converge?



Any local minimum is a global minimum



Multiple local minima may exist

Least Squares, Ridge Regression and Logistic Regression are all convex!

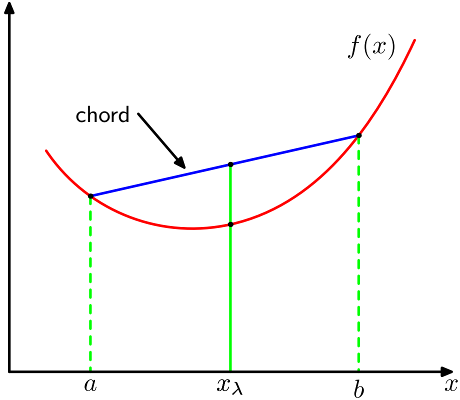
Convex functions

A function $f(x)$ is convex if

$$f(\lambda a + (1-\lambda)b) \leq \lambda f(a) + (1-\lambda) f(b)$$

for

$$0 \leq \lambda \leq 1$$



How to determine convexity?

$f(x)$ is convex if

$$f^{''}(x) \geq 0$$

Examples:

$$f(x) = x^2, f^{''}(x) = 2 > 0$$

Gradient Descent Update for Logistic Regression

Simple fact: derivatives of $\sigma(a)$

$$\begin{aligned}\frac{d\,\sigma(a)}{d\,a} &= \frac{d}{d\,a}\left(1+e^{-a}\right)^{-1} = \frac{-(1+e^{-a})'}{(1+e^{-a})^2}\\ &= \frac{e^{-a}}{(1+e^{-a})^2} = \frac{1}{1+e^{-a}}\frac{e^{-a}}{1+e^{-a}}\\ &= \sigma(a)[1-\sigma(a)]\end{aligned}$$

Gradients of the negative log likelihood

Negative log likelihood

$$J(\boldsymbol{\theta}) = -\sum_n \{y_n \log h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) + (1 - y_n) \log[1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}_n)]\}$$

Gradients

$$\nabla J(\boldsymbol{\theta}) = -\sum_n \left\{y_n[1 - \sigma(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}_n)]\boldsymbol{x}_n - (1 - y_n)\sigma(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}_n)]\boldsymbol{x}_n\right\} \tag{1}$$

$$= \sum_n \left\{\sigma(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}_n) - y_n\right\} \boldsymbol{x}_n \tag{2}$$

$$= \sum_n \{h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) - y_n\} \boldsymbol{x}_n \tag{3}$$

Remark

Gradients of the negative log likelihood

Negative log likelihood

$$J(\boldsymbol{\theta}) = -\sum_n \{y_n \log h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) + (1 - y_n) \log[1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}_n)]\}$$

Gradients

$$\nabla J(\boldsymbol{\theta}) = -\sum_n \left\{y_n[1 - \sigma(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}_n)]\boldsymbol{x}_n - (1 - y_n)\sigma(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}_n)]\boldsymbol{x}_n\right\} \quad (1)$$

$$= \sum_n \left\{\sigma(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}_n) - y_n\right\} \boldsymbol{x}_n \quad (2)$$

$$= \sum_n \left\{h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) - y_n\right\} \boldsymbol{x}_n \quad (3)$$

Remark

- $e_n = \{h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) - y_n\}$ is called *error* for the nth training sample.

Gradient descent to minimize the negative log likelihood

Algorithm 1 Gradient Descent (J)

$t \leftarrow 0$
Initialize $\theta^{(0)}$
repeat
    $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta \nabla J(\boldsymbol{\theta}^{(t)})$
    $t \leftarrow t + 1$
until convergence
Return final value of $\boldsymbol{\theta}$

Need to compute the gradient for the negative log likelihood

Gradient descent to minimize the negative log likelihood

Algorithm 1 Gradient Descent (J)

$t \leftarrow 0$
Initialize $\theta^{(0)}$
repeat
    $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta \sum_n (h_{\boldsymbol{\theta}^{(t)}}(\boldsymbol{x}_n) - y_n)\boldsymbol{x}_n$
    $t \leftarrow t + 1$
until convergence
Return final value of $\boldsymbol{\theta}$

Gradient descent

Remarks

- The step size needs to be chosen carefully to ensure convergence.
- The step size can be adaptive (i.e. varying from iteration to iteration). For example, a technique such as *line search* is often used.

Summary

Setup for binary classification

- Logistic Regression models conditional distribution as: $p(y = 1|\boldsymbol{x}; \boldsymbol{\theta}) = \sigma[a(\boldsymbol{x})]$ where $a(\boldsymbol{x}) = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}$
- Linear decision boundary: $a(\boldsymbol{x}) = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x} = 0$

Minimizing the negative log-likelihood

- $J(\boldsymbol{\theta}) = -\sum_n \{y_n \log \sigma(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}_n) + (1 - y_n) \log[1 - \sigma(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}_n)]\}$
- No closed form solution; must rely on iterative solvers

Numerical optimization

- Gradient descent: simple, scalable to large-scale problems
  - move in direction opposite of gradient!
  - gradient of logistic function takes nice form
- Brief discussion of logistic regression in CIML 6.3

Outline

Regression

Predicting a continuous outcome variable

- Predicting shoe size from height, weight and gender
- Predicting a company's future stock price using its profit and other financial info
- Predicting annual rainfall based on local flaura / fauna
- Predicting song year from audio features

Regression

Predicting a continuous outcome variable

- Predicting shoe size from height, weight and gender
- Predicting a company's future stock price using its profit and other financial info
- Predicting annual rainfall based on local flaura / fauna
- Predicting song year from audio features

Key difference from classification

Regression

Predicting a continuous outcome variable

- Predicting shoe size from height, weight and gender
- Predicting a company's future stock price using its profit and other financial info
- Predicting annual rainfall based on local flaura / fauna
- Predicting song year from audio features

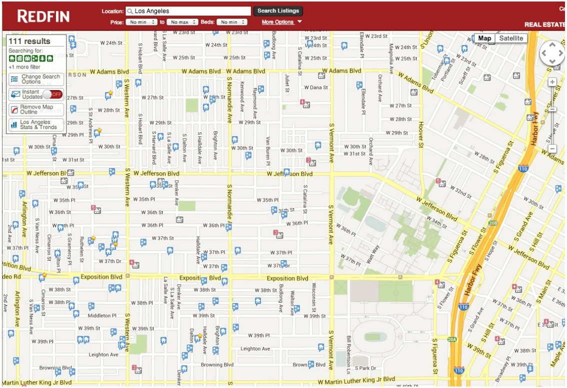
Key difference from classification

- We can measure 'closeness' of prediction and labels, leading to different ways to evaluate prediction errors.
  - Predicting shoe size: better to be off by one size than by 5 sizes
  - Predicting song year: better to be off by one year than by 20 years
- This will lead to different learning models and algorithms

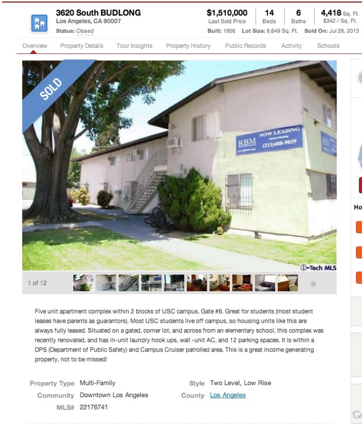
Example: predicting the sale price of a house

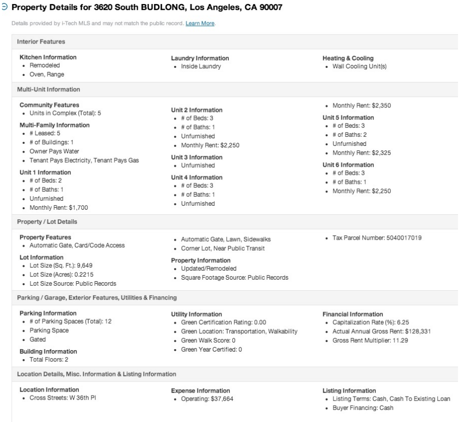
Retrieve historical sales records

(This will be our training data)



Features used to predict





Property Details for 3620 South BUDLONG, Los Angeles, CA 90007

Details provided by i-Tech MLS and may not match the public record. Learn More.

Interior Features

Kitchen Information
- Remodeled
- Oven, Range

Laundry Information
- Inside Laundry

Heating & Cooling
- Wall Cooling Unit(s)

Multi-Unit Information

Community Features
- Units in Complex (Total): 5

Multi-Family Information
- # Leased: 5
- # of Buildings: 1
- Owner Pays Water
- Tenant Pays Electricity, Tenant Pays Gas

Unit 1 Information
- # of Beds: 2
- # of Baths: 1
- Unfurnished
- Monthly Rent: $1,700

Unit 2 Information
- # of Beds: 3
- # of Baths: 1
- Unfurnished
- Monthly Rent: $2,250

Unit 3 Information
- Unfurnished

Unit 4 Information
- # of Beds: 3
- # of Baths: 1
- Unfurnished
- Monthly Rent: $2,350

Unit 5 Information
- # of Beds: 3
- # of Baths: 2
- Unfurnished
- Monthly Rent: $2,325

Unit 6 Information
- # of Beds: 3
- # of Baths: 1
- Monthly Rent: $2,250

Property / Lot Details

Property Features
- Automatic Gate, Card/Code Access
- Automatic Gate, Lawn, Sidewalks
- Corner Lot, Near Public Transit

Lot Information
- Lot Size (Sq. Ft.): 9,649
- Lot Size (Acres): 0.2215
- Lot Size Source: Public Records

Property Information
- Updated/Remodeled
- Square Footage Source: Public Records
- Tax Parcel Number: 5040017019

Parking / Garage, Exterior Features, Utilities & Financing

Parking Information
- # of Parking Spaces (Total): 12
- Parking Space
- Gated

Building Information
- Total Floors: 2

Utility Information
- Green Certification Rating: 0.00
- Green Location: Transportation, Walkability
- Green Walk Score: 0
- Green Year Certified: 0

Financial Information
- Capitalization Rate (%): 6.25
- Actual Annual Gross Rent: $128,331
- Gross Rent Multiplier: 11.29

Location Details, Misc. Information & Listing Information

Location Information
- Cross Streets: W 36th Pl

Expense Information
- Operating: $37,664

Listing Information
- Listing Terms: Cash, Cash To Existing Loan
- Buyer Financing: Cash

How to learn the unknown parameters?

training data (past sales record)

sqft	sale price
2000	800K
2100	907K
1100	312K
5500	2,600K
...	...

Our model

Sale price $=$ price_per_sqft $\times$ square_footage $+$ fixed_expense $+$ unexplainable_stuff

Reduce prediction error

How to measure errors?

- The classification error (*hit* or *miss*) is not appropriate for continuous outcomes.
- How should we evaluate quality of a prediction?

Reduce prediction error

How to measure errors?

- The classification error (*hit* or *miss*) is not appropriate for continuous outcomes.
- How should we evaluate quality of a prediction?
  - *absolute* difference: $|$ prediction - sale price$|$
  - *squared* difference: (prediction - sale price)^2

sqft	sale price	prediction	error	squared error
2000	810K	720K	90K	90^2
2100	907K	800K	107K	107^2
1100	312K	350K	-38K	38^2
5500	2,600K	2,600K	0	0
$\cdots$	$\cdots$			

Minimize squared errors

Our model

Sale price = price_per_sqft × square_footage + fixed_expense + unexplainable_stuff

Training data

sqft	sale price	prediction	error	squared error
2000	810K	720K	90K	90^2
2100	907K	800K	107K	107^2
1100	312K	350K	38K	38^2
5500	2,600K	2,600K	0	0
$\cdots$	$\cdots$			
Total				$90^2 + 107^2 + 38^2 + 0 + \cdots$

Minimize squared errors

Our model

Sale price = price_per_sqft × square_footage + fixed_expense + unexplainable_stuff

Training data

sqft	sale price	prediction	error	squared error
2000	810K	720K	90K	90^2
2100	907K	800K	107K	107^2
1100	312K	350K	38K	38^2
5500	2,600K	2,600K	0	0
...	...			
Total				$90^2 + 107^2 + 38^2 + 0 + \cdots$

Aim

Adjust model such that the sum of the squared error is minimized — i.e., the residual/remaining unexplainable_stuff is minimized.

Linear regression (ordinary least squares)

Setup

- Input: $\boldsymbol{x} \in \mathbb{R}^{\mathsf{D}}$ (covariates, predictors, features, etc)
- Output: $y \in \mathbb{R}$ (responses, targets, outcomes, outputs, etc)

Linear regression (ordinary least squares)

Setup

- Input: $\boldsymbol{x} \in \mathbb{R}^{\mathsf{D}}$ (covariates, predictors, features, etc)
- Output: $y \in \mathbb{R}$ (responses, targets, outcomes, outputs, etc)
- Hypotheses/Model: $h_{\boldsymbol{w},b}$, with $h_{\boldsymbol{w},b}(\boldsymbol{x}) = b + \sum_d w_d x_d = b + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$

  $\boldsymbol{w} = [w_1 \ w_2 \ \cdots \ w_{\mathsf{D}}]^{\mathrm{T}}$: *weights*

  b is called the bias or offset or intercept term.

  $\boldsymbol{\theta} = [b \ w_1 \ w_2 \ \cdots \ w_{\mathsf{D}}]^{\mathrm{T}}$

Linear regression (ordinary least squares)

Setup

- Input: $\boldsymbol{x} \in \mathbb{R}^{\mathsf{D}}$ (covariates, predictors, features, etc)
- Output: $y \in \mathbb{R}$ (responses, targets, outcomes, outputs, etc)
- Hypotheses/Model: $h_{\boldsymbol{w},b}$, with $h_{\boldsymbol{w},b}(\boldsymbol{x}) = b + \sum_d w_d x_d = b + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$

  $\boldsymbol{w} = [w_1\ w_2\ \cdots\ w_{\mathsf{D}}]^{\mathrm{T}}$: *weights*

  b is called the bias or offset or intercept term.

  $\boldsymbol{\theta} = [b\ w_1\ w_2\ \cdots\ w_{\mathsf{D}}]^{\mathrm{T}}$
- Training data: $\mathcal{D} = \{(\boldsymbol{x}_n, y_n), n = 1, 2, \ldots, \mathsf{N}\}$

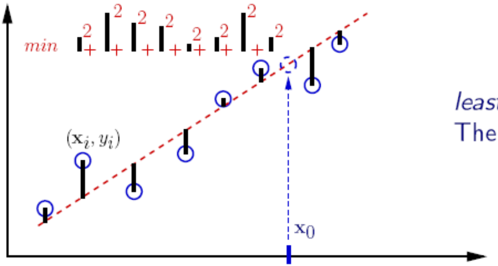
How do we learn parameters?

Minimize prediction error on training data

- Hypothesis:

$$h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x$$

- Minimize the sum of squared errors (also called residual sum of squares RSS): cost function for linear regression.
- Cost function for logistic regression is the negative log likelihood.

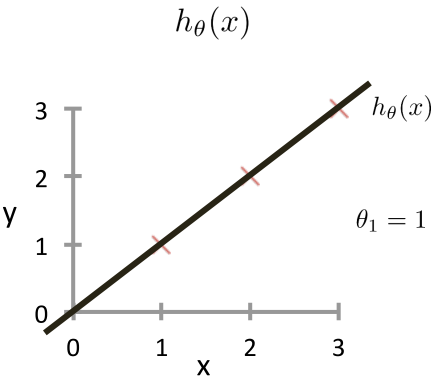


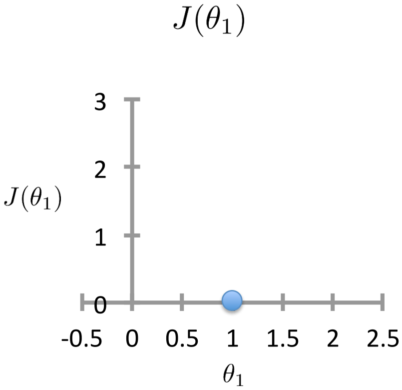
least squares (LSQ)
The fitted line is used as a predictor

Intuiton behind cost function (residual sum of squares RSS)

Assume $x \in \mathbb{R}$, $\theta_0 = 0$.

$h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x = \theta_1 x$

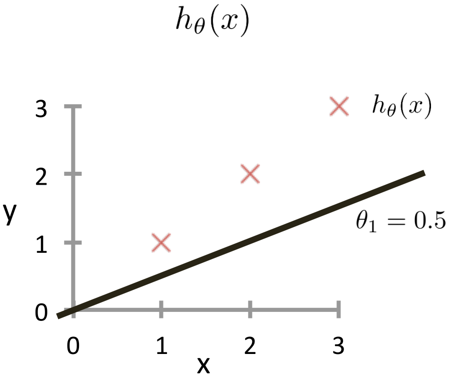


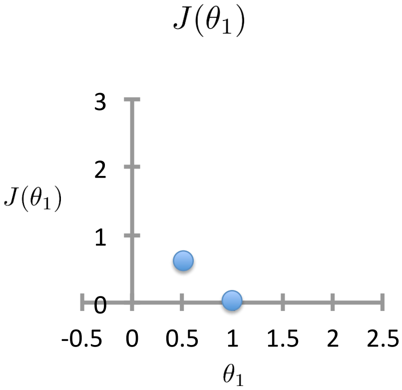


Intuiton behind cost function (residual sum of squares RSS)

Assume $x \in \mathbb{R}$, $\theta_0 = 0$.

$h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x = \theta_1 x$

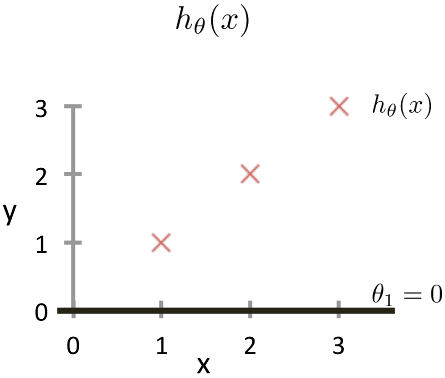


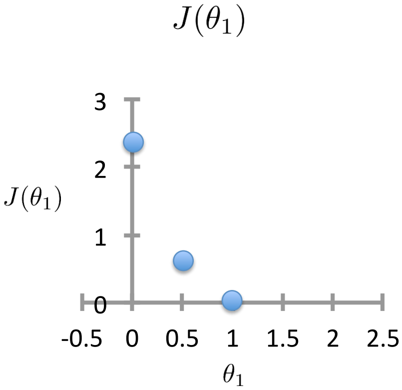


Intuiton behind cost function (residual sum of squares RSS)

Assume $x \in \mathbb{R}$, $\theta_0 = 0$.

$h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x = \theta_1 x$



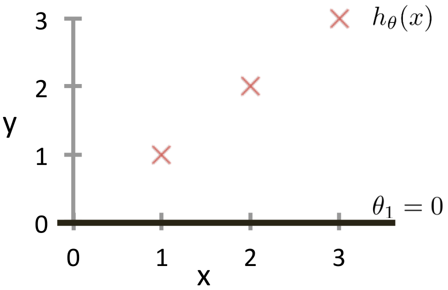


Intuiton behind cost function (residual sum of squares RSS)

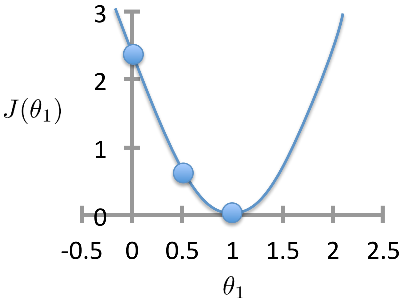
Assume $x \in \mathbb{R}$, $\theta_0 = 0$.

$h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x = \theta_1 x$

$h_\theta(x)$

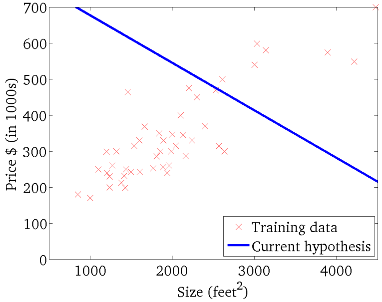


$J(\theta_1)$

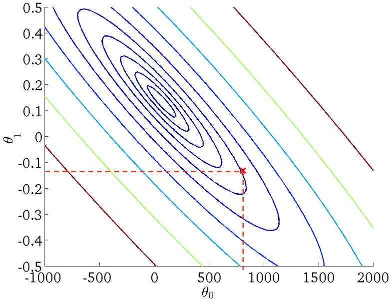


Intuiton behind cost function (residual sum of squares)

$h_\theta(x)$

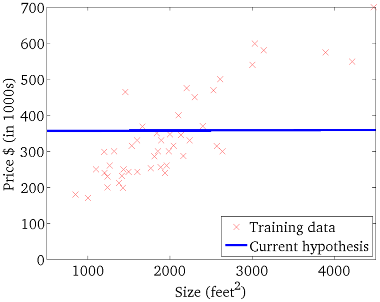


$J(\theta_0, \theta_1)$

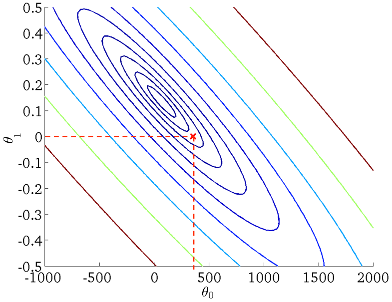


Intuiton behind cost function (residual sum of squares)

$h_\theta(x)$

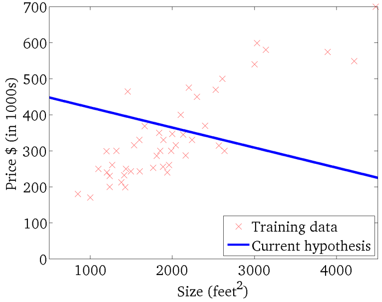


$J(\theta_0, \theta_1)$

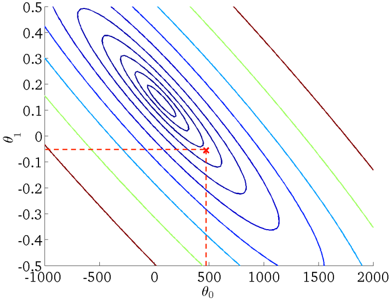


Intuiton behind cost function (residual sum of squares)

$h_\theta(x)$

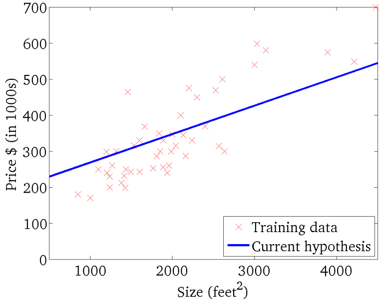


$J(\theta_0, \theta_1)$

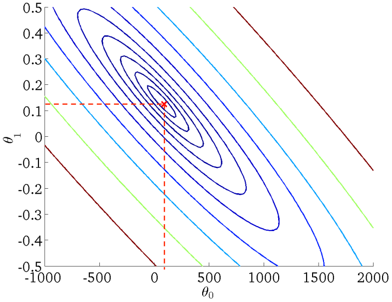


Intuiton behind cost function (residual sum of squares)

$h_\theta(x)$



$J(\theta_0, \theta_1)$



How do we minimize the RSS?

Numerical optimization

Algorithm 2 Gradient Descent (J)

$t \leftarrow 0$
Initialize $\theta^{(0)}$
repeat
    $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta \nabla J(\boldsymbol{\theta}^{(t)})$
    $t \leftarrow t + 1$
until convergence
Return final value of $\boldsymbol{\theta}$

Need to compute the gradient for the linear regression cost function (residual sum of squares RSS)

How do we minimize the RSS?

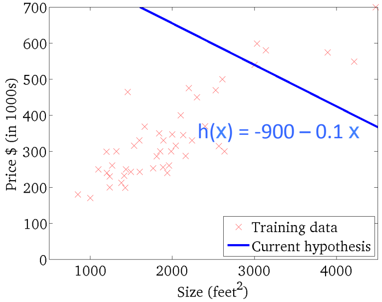
Numerical optimization

Algorithm 2 Gradient Descent (J)

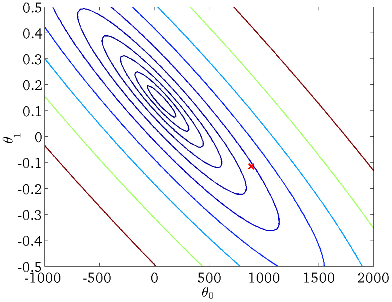
$t \leftarrow 0$
Initialize $\theta^{(0)}$
repeat
    $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta \sum_n (h_{\boldsymbol{\theta}^{(t)}}(\boldsymbol{x}_n) - y_n)\boldsymbol{x}_n$
    $t \leftarrow t + 1$
until convergence
Return final value of $\boldsymbol{\theta}$

Gradient descent

$h_\theta(x)$

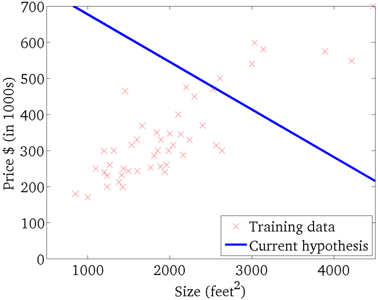


$J(\theta_0, \theta_1)$

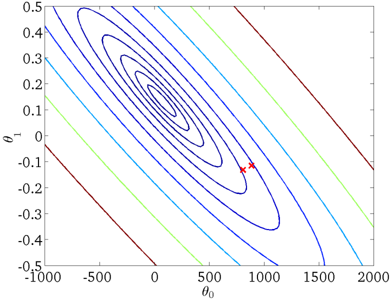


Gradient descent

$h_\theta(x)$

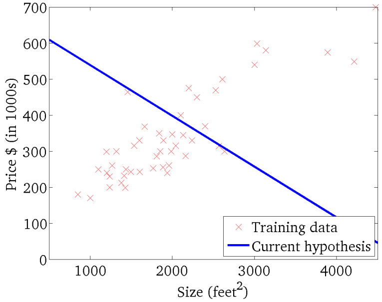


$J(\theta_0, \theta_1)$

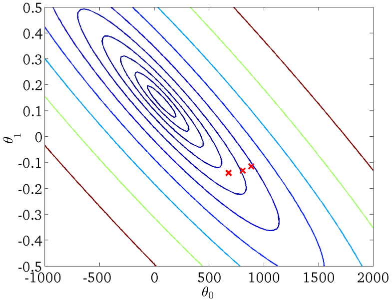


Gradient descent

$h_\theta(x)$

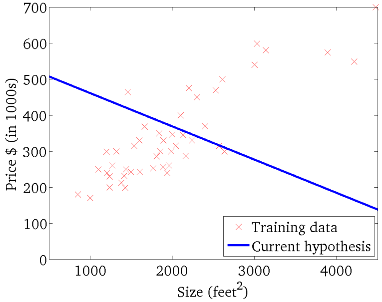


$J(\theta_0, \theta_1)$

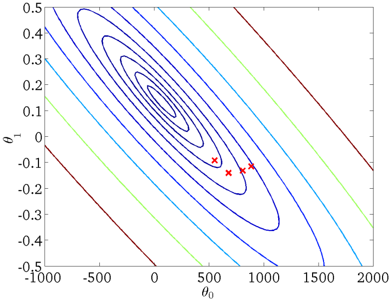


Gradient descent

$h_\theta(x)$

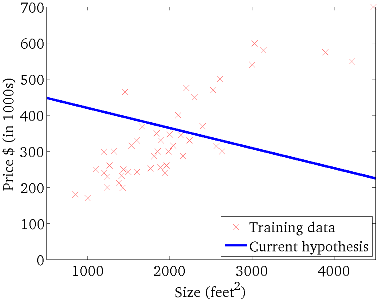


$J(\theta_0, \theta_1)$

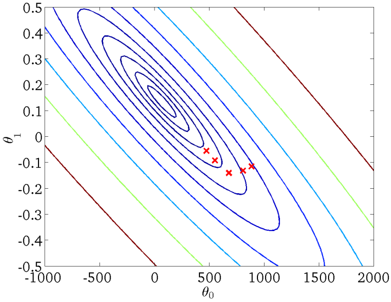


Gradient descent

$h_\theta(x)$

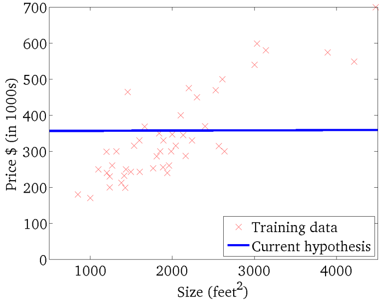


$J(\theta_0, \theta_1)$

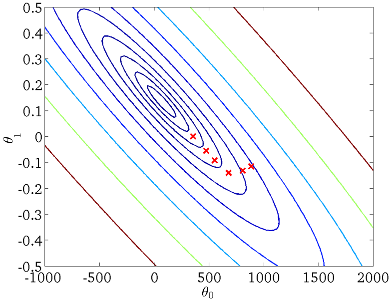


Gradient descent

$h_\theta(x)$

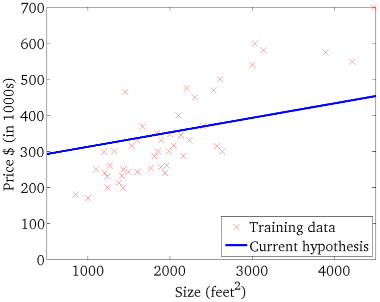


$J(\theta_0, \theta_1)$

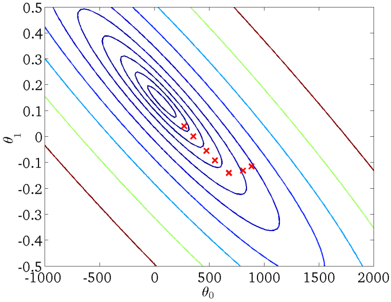


Gradient descent

$h_\theta(x)$

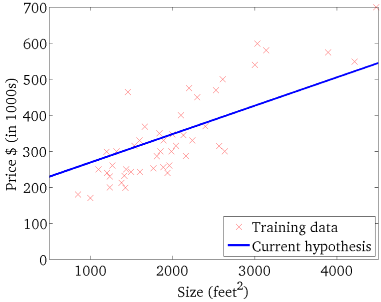


$J(\theta_0, \theta_1)$

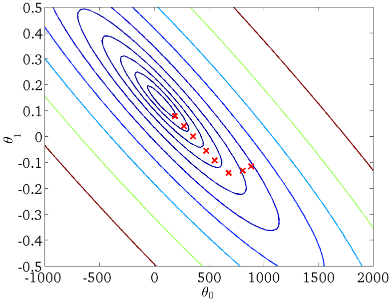


Gradient descent

$h_\theta(x)$

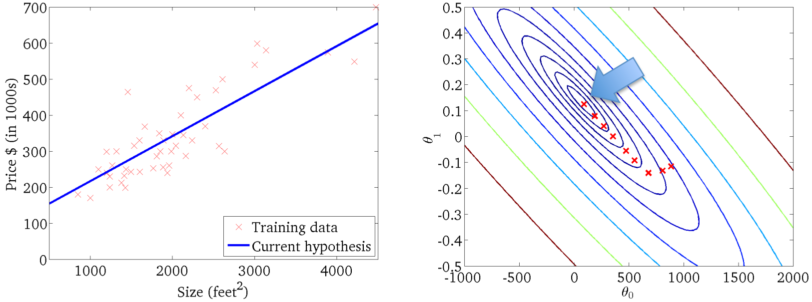


$J(\theta_0, \theta_1)$



Gradient descent

$h_\theta(x)$ $J(\theta_0, \theta_1)$



How do we minimize the cost function (residual sum of squares)?

Numerical optimization

Gradient descent

Analytical solution

Can compute minimum in closed form for linear regression!

A simple case: $\boldsymbol{x}$ is just one-dimensional ($D=1$)

Residual sum of squares (RSS)

$$J(\boldsymbol{\theta}) = \sum_n [y_n - h_{\boldsymbol{\theta}}(\boldsymbol{x}_n)]^2 = \sum_n [y_n - (\theta_0 + \theta_1 x_n)]^2$$

A simple case: $\boldsymbol{x}$ is just one-dimensional ($D=1$)

Residual sum of squares (RSS)

$$J(\boldsymbol{\theta}) = \sum_n [y_n - h_{\boldsymbol{\theta}}(\boldsymbol{x}_n)]^2 = \sum_n [y_n - (\theta_0 + \theta_1 x_n)]^2$$

Identify stationary points by taking derivative with respect to parameters and setting to zero

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_0} = 0 \Rightarrow -2 \sum_n [y_n - (\theta_0 + \theta_1 x_n)] = 0$$

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_1} = 0 \Rightarrow -2 \sum_n [y_n - (\theta_0 + \theta_1 x_n)] x_n = 0$$

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_0} = 0 \Rightarrow -2\sum_n [y_n - (\theta_0 + \theta_1 x_n)] = 0$$

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_1} = 0 \Rightarrow -2\sum_n [y_n - (\theta_0 + \theta_1 x_n)]x_n = 0$$

Simplify these expressions to get "Normal Equations"

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_0} = 0 \Rightarrow -2\sum_n [y_n - (\theta_0 + \theta_1 x_n)] = 0$$

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_1} = 0 \Rightarrow -2\sum_n [y_n - (\theta_0 + \theta_1 x_n)]x_n = 0$$

Simplify these expressions to get "Normal Equations"

$$\sum y_n = N\theta_0 + \theta_1 \sum x_n$$

$$\sum x_n y_n = \theta_0 \sum x_n + \theta_1 \sum x_n^2$$

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_0} = 0 \Rightarrow -2\sum_n [y_n - (\theta_0 + \theta_1 x_n)] = 0$$

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_1} = 0 \Rightarrow -2\sum_n [y_n - (\theta_0 + \theta_1 x_n)]x_n = 0$$

Simplify these expressions to get "Normal Equations"

$$\sum y_n = N\theta_0 + \theta_1 \sum x_n$$

$$\sum x_n y_n = \theta_0 \sum x_n + \theta_1 \sum x_n^2$$

We have two equations and two unknowns! Do some algebra to get:

$$\theta_1 = \frac{\sum (x_n - \bar{x})(y_n - \bar{y})}{\sum (x_i - \bar{x})^2} \qquad \text{and} \qquad \theta_0 = \bar{y} - \theta_1 \bar{x}$$

where $\bar{x} = \frac{1}{n}\sum_n x_n$ and $\bar{y} = \frac{1}{n}\sum_n y_n$.

Why is minimizing J sensible?

Probabilistic interpretation

- Noisy observation model

$$Y = \theta_0 + \theta_1 X + \eta$$

where $\eta \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian random variable

Why is minimizing J sensible?

Probabilistic interpretation

- Noisy observation model

$$Y = \theta_0 + \theta_1 X + \eta$$

  where $\eta \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian random variable
- Likelihood of one training sample (x_n, y_n)

$$p(y_n | x_n; \boldsymbol{\theta}) = \mathcal{N}(\theta_0 + \theta_1 x_n, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{[y_n - (\theta_0 + \theta_1 x_n)]^2}{2\sigma^2}}$$

Probabilistic interpretation (cont'd)

Log-likelihood of the training data $\mathcal{D}$ (assuming i.i.d)

$$\begin{aligned}
\mathcal{LL}(\boldsymbol{\theta}) &= \log P(\mathcal{D}) \\
&= \log \prod_{n=1}^{\mathsf{N}} p(y_n|x_n) = \sum_n \log p(y_n|x_n)
\end{aligned}$$

Probabilistic interpretation (cont'd)

Log-likelihood of the training data $\mathcal{D}$ (assuming i.i.d)

$$
\begin{aligned}
\mathcal{LL}(\boldsymbol{\theta}) &= \log P(\mathcal{D}) \\
&= \log \prod_{n=1}^{\mathsf{N}} p(y_n|x_n) = \sum_n \log p(y_n|x_n) \\
&= \sum_n \left\{ -\frac{[y_n - (\theta_0 + \theta_1 x_n)]^2}{2\sigma^2} - \log \sqrt{2\pi}\sigma \right\}
\end{aligned}
$$

Probabilistic interpretation (cont'd)

Log-likelihood of the training data $\mathcal{D}$ (assuming i.i.d)

$$
\begin{aligned}
\mathcal{LL}(\boldsymbol{\theta}) &= \log P(\mathcal{D}) \\
&= \log \prod_{n=1}^{\mathsf{N}} p(y_n|x_n) = \sum_n \log p(y_n|x_n) \\
&= \sum_n \left\{ -\frac{[y_n - (\theta_0 + \theta_1 x_n)]^2}{2\sigma^2} - \log \sqrt{2\pi}\sigma \right\} \\
&= -\frac{1}{2\sigma^2} \sum_n [y_n - (\theta_0 + \theta_1 x_n)]^2 - \frac{\mathsf{N}}{2} \log \sigma^2 - \mathsf{N} \log \sqrt{2\pi}
\end{aligned}
$$

Probabilistic interpretation (cont'd)

Log-likelihood of the training data $\mathcal{D}$ (assuming i.i.d)

$$
\begin{aligned}
\mathcal{LL}(\boldsymbol{\theta}) &= \log P(\mathcal{D}) \\
&= \log \prod_{n=1}^{\mathsf{N}} p(y_n|x_n) = \sum_n \log p(y_n|x_n) \\
&= \sum_n \left\{ -\frac{[y_n - (\theta_0 + \theta_1 x_n)]^2}{2\sigma^2} - \log \sqrt{2\pi}\sigma \right\} \\
&= -\frac{1}{2\sigma^2} \sum_n [y_n - (\theta_0 + \theta_1 x_n)]^2 - \frac{\mathsf{N}}{2} \log \sigma^2 - \mathsf{N} \log \sqrt{2\pi} \\
&= -\frac{1}{2} \left\{ \frac{1}{\sigma^2} \sum_n [y_n - (\theta_0 + \theta_1 x_n)]^2 + \mathsf{N} \log \sigma^2 \right\} + \text{const}
\end{aligned}
$$

What is the relationship between minimizing J and maximizing the log-likelihood?

Maximum likelihood estimation

Estimating σ, θ_0 and θ_1 can be done in two steps

- Maximize over θ_0 and θ_1

$$\max \; \log P(\mathcal{D}) \Leftrightarrow \min \; \sum_n [y_n - (\theta_0 + \theta_1 x_n)]^2 \leftarrow \text{That is } J(\boldsymbol{\theta})!$$

Maximum likelihood estimation

Estimating σ, θ_0 and θ_1 can be done in two steps

- Maximize over θ_0 and θ_1

$$\max \; \log P(\mathcal{D}) \Leftrightarrow \min \; \sum_n [y_n - (\theta_0 + \theta_1 x_n)]^2 \leftarrow \text{That is } J(\boldsymbol{\theta})!$$

- Maximize over $s = \sigma^2$ (we could estimate σ directly)

$$\frac{\partial \log P(\mathcal{D})}{\partial s} = -\frac{1}{2}\left\{ -\frac{1}{s^2} \sum_n [y_n - (\theta_0 + \theta_1 x_n)]^2 + \mathsf{N}\frac{1}{s} \right\} = 0$$

Maximum likelihood estimation

Estimating σ, θ_0 and θ_1 can be done in two steps

- Maximize over θ_0 and θ_1

$$\max \log P(\mathcal{D}) \Leftrightarrow \min \sum_n [y_n - (\theta_0 + \theta_1 x_n)]^2 \leftarrow \text{That is } J(\boldsymbol{\theta})!$$

- Maximize over $s = \sigma^2$ (we could estimate σ directly)

$$\frac{\partial \log P(\mathcal{D})}{\partial s} = -\frac{1}{2}\left\{-\frac{1}{s^2}\sum_n [y_n - (\theta_0 + \theta_1 x_n)]^2 + \mathsf{N}\frac{1}{s}\right\} = 0$$

$$\rightarrow \sigma^{*2} = s^* = \frac{1}{\mathsf{N}}\sum_n [y_n - (\theta_0 + \theta_1 x_n)]^2$$

Summary

- Use of linear models for classification and regression.
- Learning is a problem of optimization.
  - The objective function is convex.
  - Numerical methods and sometimes analytical solutions.
- Next class: linear regression for multi-dimensional inputs and going beyond linearity.