Linear Regression

Machine Learning Decal

Hosted by Machine Learning at Berkeley

Agenda

Background

Linear Algebra Perspective

Optimization via Gradient Descent

Probabilistic Perspective

Background

Where are we?

	Continuous	Discrete
Supervised	**Regression**	Classification
Unsupervised	Dimensionality Reduction	Clustering

- Suppose we have data

Where are we?

	Continuous	Discrete
Supervised	**Regression**	Classification
Unsupervised	Dimensionality Reduction	Clustering

- Suppose we have data
    - Want to model relationships and make **predictions**

Where are we?

	Continuous	Discrete
Supervised	**Regression**	Classification
Unsupervised	Dimensionality Reduction	Clustering

- Suppose we have data
    - Want to model relationships and make **predictions**
- The data has **continuous** labels (y)

Where are we?

	Continuous	Discrete
Supervised	**Regression**	Classification
Unsupervised	Dimensionality Reduction	Clustering

- Suppose we have data
  - Want to model relationships and make **predictions**
- The data has **continuous** labels (y)
  - i.e. prices, heights, miles per gallon, etc.

Where are we?

	Continuous	Discrete
Supervised	**Regression**	Classification
Unsupervised	Dimensionality Reduction	Clustering

- Suppose we have data
    - Want to model relationships and make **predictions**
- The data has **continuous** labels (y)
    - i.e. prices, heights, miles per gallon, etc.
- The data has a set of **explanatory** variables (x_i)

Where are we?

	Continuous	Discrete
Supervised	**Regression**	Classification
Unsupervised	Dimensionality Reduction	Clustering

- Suppose we have data
  - Want to model relationships and make **predictions**
- The data has **continuous** labels (y)
  - i.e. prices, heights, miles per gallon, etc.
- The data has a set of **explanatory** variables (x_i)
  - i.e. sales, weights, engine power, etc.

Where are we?

	Continuous	Discrete
Supervised	**Regression**	Classification
Unsupervised	Dimensionality Reduction	Clustering

- Suppose we have data
  - Want to model relationships and make **predictions**
- The data has **continuous** labels (y)
  - i.e. prices, heights, miles per gallon, etc.
- The data has a set of **explanatory** variables (x_i)
  - i.e. sales, weights, engine power, etc.
- How does a computer make predictions?

Why linear regression?

- Regression is one of the most commonly used methods by data scientists

Why linear regression?

- Regression is one of the most commonly used methods by data scientists
- It is simple, fast, interpretable, and **powerful**

Why linear regression?

- Regression is one of the most commonly used methods by data scientists
- It is simple, fast, interpretable, and **powerful**
- The techniques we use here are widely applicable

Why linear regression?

- Regression is one of the most commonly used methods by data scientists
- It is simple, fast, interpretable, and **powerful**
- The techniques we use here are widely applicable
- It is practical!

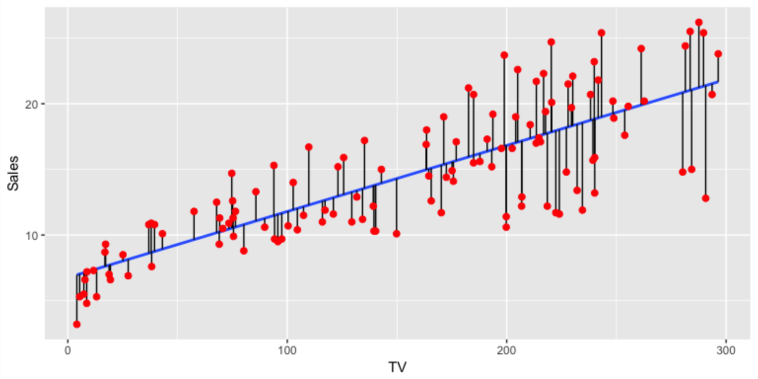
Why linear regression?

- Regression is one of the most commonly used methods by data scientists
- It is simple, fast, interpretable, and **powerful**
- The techniques we use here are widely applicable
- It is practical!
    - (Physics) Ohm's law, Hooke's law, Charles's law

Why linear regression?

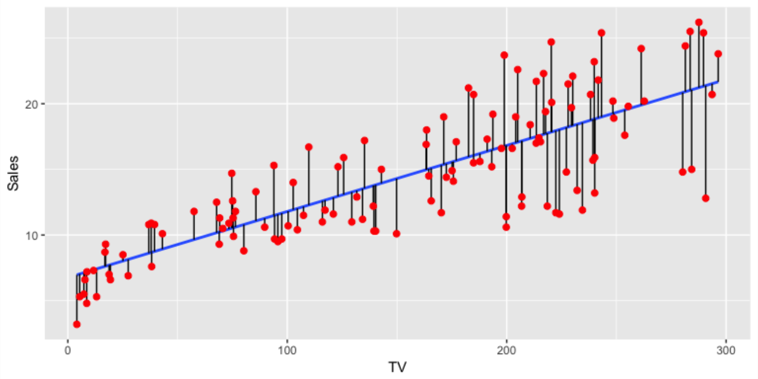
- Regression is one of the most commonly used methods by data scientists
- It is simple, fast, interpretable, and **powerful**
- The techniques we use here are widely applicable
- It is practical!
    - (Physics) Ohm's law, Hooke's law, Charles's law
    - (Economics) Okun's law

Example of Linear Regression



How exactly is this line calculated?

Example of Linear Regression



How exactly is this line calculated?

Minimizing the sum of the square of those errors!

Linear regression

- Suppose we have p predictor variables $x_1, x_2, \ldots, x_p$

Linear regression

- Suppose we have p predictor variables x_1, x_2, . . . , x_p
- We can approximate y as a linear function of the $x_i's$:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_p x_p$$

- Suppose we have p predictor variables $x_1, x_2, \ldots, x_p$
- We can approximate y as a linear function of the $x_i' s$:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_p x_p$$

- θ_i's are the **parameters** (also called **weights**) which we need to estimate

Linear regression

- Suppose we have p predictor variables $x_1, x_2, \ldots, x_p$
- We can approximate y as a linear function of the $x_i's$:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_p x_p$$

- θ_i's are the **parameters** (also called **weights**) which we need to estimate
- We introduce $x_0 = 1$ for simplicity so that:

$$h_\theta(x) = \sum_{i=1}^{p} \theta_i^T x_i = \theta^T x$$

Warm-up: predicting house prices

- Suppose we have the following data about houses:

Price	# of Square Feet	# of Bedrooms
221,900	1180	3
538,000	2570	3
⋮	⋮	⋮
1,225,000	5420	4

Warm-up: predicting house prices

- Suppose we have the following data about houses:

Price	**# of Square Feet**	**# of Bedrooms**
221,900	1180	3
538,000	2570	3
⋮	⋮	⋮
1,225,000	5420	4

- Let's predict the price of a house from the number of square feet it has

Warm-up: predicting house prices

- Suppose we have the following data about houses:

Price	# of Square Feet	# of Bedrooms
221,900	1180	3
538,000	2570	3
$\vdots$	$\vdots$	$\vdots$
1,225,000	5420	4

- Let's predict the price of a house from the number of square feet it has
- Our linear model has the form:

$$h_\theta(sqft) = \theta_0 + \theta_1 sqft$$

- We are given labeled data in (x, y) pairs

Brief Recap

- We are given labeled data in (x, y) pairs
- We can construct our model $h_\theta(x)$ to predict y

Brief Recap

- We are given labeled data in (x, y) pairs
- We can construct our model $h_\theta(x)$ to predict y
- TODO: how do we figure out how close $h_\theta(x)$ is to y?

Brief Recap

- We are given labeled data in (x, y) pairs
- We can construct our model $h_\theta(x)$ to predict y
- TODO: how do we figure out how close $h_\theta(x)$ is to y?
- TODO: how do we optimize $h_\theta(x)$ to be as close as possible to y?

Linear Algebra Perspective

How do we choose θ?

- **Goal**: have $h_\theta(x)$ be as close to y as possible

How do we choose θ?

- **Goal**: have $h_\theta(x)$ be as close to y as possible
- Idea: measure how different each prediction is via squared error

How do we choose θ?

- **Goal**: have $h_\theta(x)$ be as close to y as possible
- Idea: measure how different each prediction is via squared error
- We can express this mathematically:

$$L(\theta) = \sum_{i=1}^{n} (h_\theta(x^i) - y^i)^2$$

How do we choose θ?

- **Goal**: have $h_\theta(x)$ be as close to y as possible
- Idea: measure how different each prediction is via squared error
- We can express this mathematically:

$$L(\theta) = \sum_{i=1}^{n} (h_\theta(x^i) - y^i)^2$$

- $L(\theta)$ sums the squared **residuals**

How do we choose θ?

- **Goal**: have $h_\theta(x)$ be as close to y as possible
- Idea: measure how different each prediction is via squared error
- We can express this mathematically:

$$L(\theta) = \sum_{i=1}^{n} (h_\theta(x^i) - y^i)^2$$

- $L(\theta)$ sums the squared **residuals**
- To have an accurate model, we want to **minimize** $L(\theta)$

Matrix Notation

Recall:

$$L(\theta) = \sum_{i=1}^{n} (h_\theta(x^i) - y^i)^2$$

Matrix Notation

Recall:

$$L(\theta) = \sum_{i=1}^{n} (h_\theta(x^i) - y^i)^2$$

Substituting in:

$$L(\theta) = \sum_{i=1}^{n} (y^i - \theta^T x^i)^2$$

Matrix Notation

Recall:

$$L(\theta) = \sum_{i=1}^{n} (h_\theta(x^i) - y^i)^2$$

Substituting in:

$$L(\theta) = \sum_{i=1}^{n} (y^i - \theta^T x^i)^2$$

Supposing x^i was a row vector:

$$L(\theta) = \sum_{i=1}^{n} ||y^i - x^i\theta||^2$$

Matrix Notation

Recall:

$$L(\theta) = \sum_{i=1}^{n} (h_\theta(x^i) - y^i)^2$$

Substituting in:

$$L(\theta) = \sum_{i=1}^{n} (y^i - \theta^T x^i)^2$$

Supposing x^i was a row vector:

$$L(\theta) = \sum_{i=1}^{n} ||y^i - x^i\theta||^2$$

Now in matrix notation:

$$L(\theta) = ||Y - X\theta||^2$$

Recall:

$$L(\theta) = \sum_{i=1}^{n} (h_\theta(x^i) - y^i)^2$$

Substituting in:

$$L(\theta) = \sum_{i=1}^{n} (y^i - \theta^T x^i)^2$$

Supposing x^i was a row vector:

$$L(\theta) = \sum_{i=1}^{n} ||y^i - x^i\theta||^2$$

Now in matrix notation:

$$L(\theta) = ||Y - X\theta||^2$$

Matrix Notation

- The equation for a linear model:

$$h_\theta(x_i) = \theta_0 + \theta_1 x_{i1} + \cdots + \theta_p x_{ip}$$

Matrix Notation

- The equation for a linear model:

$$h_\theta(x_i) = \theta_0 + \theta_1 x_{i1} + \cdots + \theta_p x_{ip}$$

- The outcome, y, which we observe can be thought of as:

$$y_i = h_\theta(x_i) + \epsilon_i$$

where ϵ is some unobserved error

- The equation for a linear model:

$$h_\theta(x_i) = \theta_0 + \theta_1 x_{i1} + \cdots + \theta_p x_{ip}$$

- The outcome, y, which we observe can be thought of as:

$$y_i = h_\theta(x_i) + \epsilon_i$$

  where ϵ is some unobserved error

- We don't know the true θ is, so we estimate it with $\hat{\theta}$

Matrix Notation

- The equation for a linear model:

$$h_\theta(x_i) = \theta_0 + \theta_1 x_{i1} + \cdots + \theta_p x_{ip}$$

- The outcome, y, which we observe can be thought of as:

$$y_i = h_\theta(x_i) + \epsilon_i$$

where ϵ is some unobserved error

- We don't know the true θ is, so we estimate it with $\hat{\theta}$
- Our predictions for test points are then

$$\hat{y} = h_{\hat{\theta}}(x)$$

- We can rewrite linear regression as

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

Linear regression in matrix form

- We can rewrite linear regression as

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

- In compressed notation:

$$\vec{y} = X\vec{\theta} + \vec{\epsilon}$$

Linear regression in matrix form

- We can rewrite linear regression as

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

- In compressed notation:

$$\vec{y} = X\vec{\theta} + \vec{\epsilon}$$

- Here, we're using capital letters to represent matrices, and arrows to represent vectors

Estimating θ

- We want our estimate, $\hat{\theta}$ to be accurate

Estimating θ

- We want our estimate, $\hat{\theta}$ to be accurate
- We can be accurate by trying to minimize error

Estimating θ

- We want our estimate, $\hat{\theta}$ to be accurate
- We can be accurate by trying to minimize error
- We can be accurate by minimizing our residuals

$$e_i = y_i - \vec{\theta}^T x_i$$

Estimating θ

- We want our estimate, $\hat{\theta}$ to be accurate
- We can be accurate by trying to minimize error
- We can be accurate by minimizing our residuals

$$e_i = y_i - \vec{\theta}^T x_i$$

- More mathematically convenient to minimize squared residuals

Estimating θ

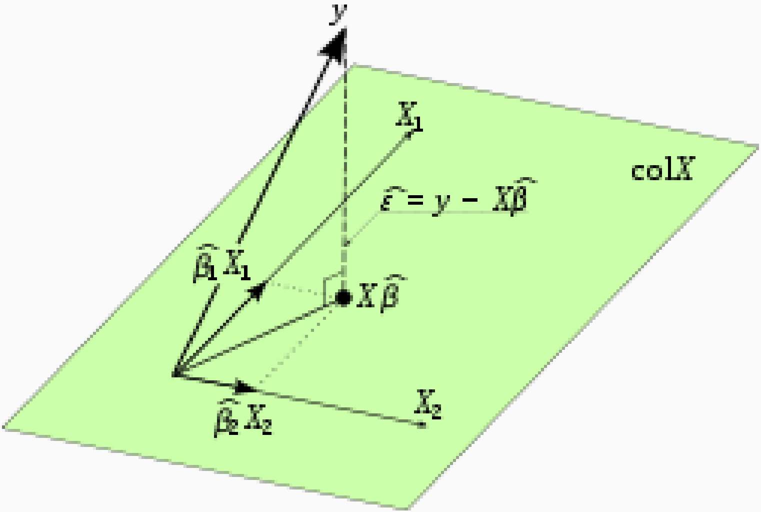
- We want our estimate, $\hat{\theta}$ to be accurate
- We can be accurate by trying to minimize error
- We can be accurate by minimizing our residuals

$$e_i = y_i - \vec{\theta}^T x_i$$

- More mathematically convenient to minimize squared residuals
- That is,

$$\hat{\theta} = argmin_{\vec{\theta}}\ L(\theta) = argmin_{\vec{\theta}}\ ||\vec{y} - X\vec{\theta}||_2^2$$

Geometric Interpretation



Projection of y on the features of X

Estimation (least squares) via vector calculus

$$\boxed{\hat{\theta} = (X^T X)^{-1} X^T \vec{y}}$$

Estimation (least squares) via vector calculus

$$\boxed{\hat{\theta} = (X^TX)^{-1}X^T\vec{y}}$$

$$\hat{\theta} = argmin_{\vec{\theta}}L(\theta) = argmin_{\vec{\theta}}||\vec{y} - X\vec{\theta}||_2^2$$

$$= argmin_{\vec{\theta}}(\vec{y} - X\vec{\theta})^T(\vec{y} - X\vec{\theta})$$

$$= argmin_{\vec{\theta}}\vec{y}^T\vec{y} - 2\vec{y}X\vec{\theta} + \vec{\theta}^TX^TX\vec{\theta}$$

How to minimize a function? Take the derivative!

$$\frac{\partial L}{\partial \vec{\theta}} = 2X^TX\vec{\theta} - 2X^T\vec{y} = 0$$

$$\hat{\theta} = (X^TX)^{-1}X^T\vec{y}$$

Model interpretation

- $\hat{\theta}$ is indeed a minimizer (the second derivative is negative)

Model interpretation

- $\hat{\theta}$ is indeed a minimizer (the second derivative is negative)
- Recall, once we have our estimate $\hat{\theta}$, we can predict new x's using:

$$\hat{y}_i = \hat{\theta}_0 + \hat{\theta}_1 x_{i1} + \cdots + \hat{\theta}_p x_{ip}$$

Model interpretation

- $\hat{\theta}$ is indeed a minimizer (the second derivative is negative)
- Recall, once we have our estimate $\hat{\theta}$, we can predict new x's using:

$$\hat{y}_i = \hat{\theta}_0 + \hat{\theta}_1 x_{i1} + \cdots + \hat{\theta}_p x_{ip}$$

- In matrix notation:

$$\hat{\vec{y}} = X\hat{\vec{\theta}} = X(X^T X)^{-1} X^T Y$$

Model interpretation

- $\hat{\theta}$ is indeed a minimizer (the second derivative is negative)
- Recall, once we have our estimate $\hat{\theta}$, we can predict new x's using:

$$\hat{y}_i = \hat{\theta}_0 + \hat{\theta}_1 x_{i1} + \cdots + \hat{\theta}_p x_{ip}$$

- In matrix notation:

$$\hat{\vec{y}} = X\hat{\vec{\theta}} = X(X^T X)^{-1} X^T Y$$

- For a one unit increase in x_{ik}, we expect y_i to, **on average** increase by $\hat{\theta}_k$

Optimization via Gradient Descent

How do we choose θ?

- **Goal**: have $h_\theta(x)$ be as close to y as possible

How do we choose θ?

- **Goal**: have $h_\theta(x)$ be as close to y as possible
- We've seen how to do this by solving some equations

How do we choose θ?

- **Goal**: have $h_\theta(x)$ be as close to y as possible
- We've seen how to do this by solving some equations
- Let's now look at a more general purpose algorithm to see if we can still reach the same answer (will allow us to optimize more complex models)

How do we choose θ?

- **Goal**: have $h_\theta(x)$ be as close to y as possible
- We've seen how to do this by solving some equations
- Let's now look at a more general purpose algorithm to see if we can still reach the same answer (will allow us to optimize more complex models)
- Remember our **cost function**:

$$L(\theta) = \frac{1}{2}\sum_{i=1}^{n}(h_\theta(x^i) - y^i)^2$$

How do we choose θ?

- **Goal**: have $h_\theta(x)$ be as close to y as possible
- We've seen how to do this by solving some equations
- Let's now look at a more general purpose algorithm to see if we can still reach the same answer (will allow us to optimize more complex models)
- Remember our **cost function**:

$$L(\theta) = \frac{1}{2}\sum_{i=1}^{n}(h_\theta(x^i) - y^i)^2$$

- $L(\theta)$ sums the squared **residuals**

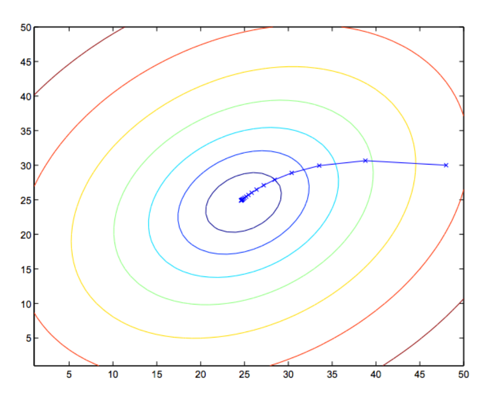
How do we choose θ?

- **Goal**: have $h_\theta(x)$ be as close to y as possible
- We've seen how to do this by solving some equations
- Let's now look at a more general purpose algorithm to see if we can still reach the same answer (will allow us to optimize more complex models)
- Remember our **cost function**:

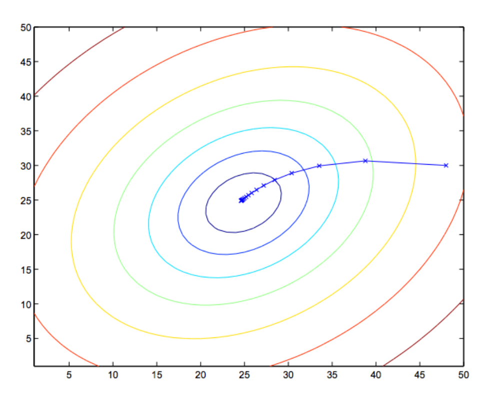
$$L(\theta) = \frac{1}{2}\sum_{i=1}^{n}(h_\theta(x^i) - y^i)^2$$

- $L(\theta)$ sums the squared **residuals**
- To have an accurate model, we want to **minimize** $L(\theta)$

- **Idea**: choose θ to minimize $L(\theta)$

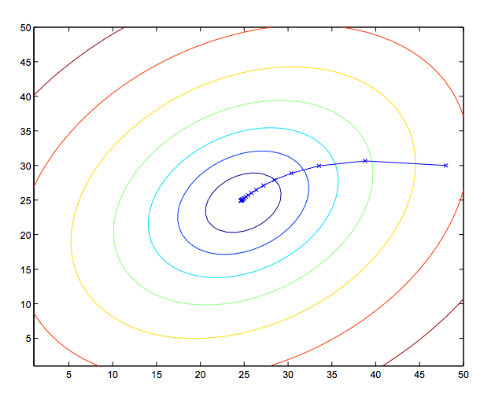


- **Idea**: choose θ to minimize $L(\theta)$
- We can use a search algorithm that follows the scheme:



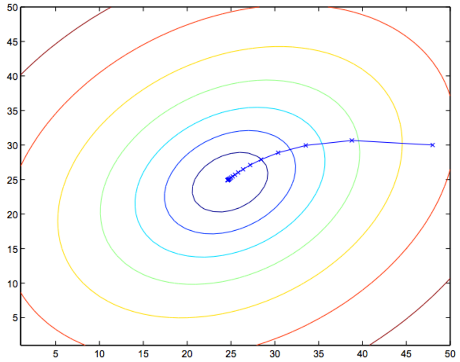
Minimizing the cost function

- **Idea**: choose θ to minimize $L(\theta)$
- We can use a search algorithm that follows the scheme:
  - Choose an initial guess for θ



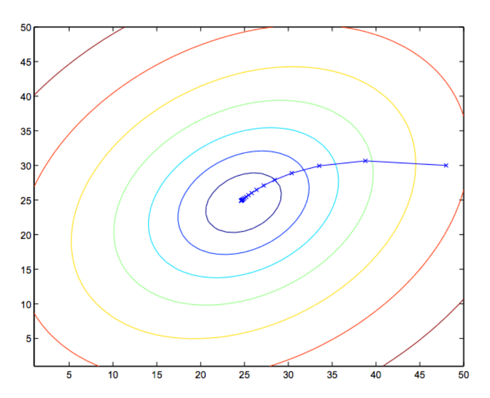
Minimizing the cost function

- **Idea**: choose θ to minimize $L(\theta)$
- We can use a search algorithm that follows the scheme:
  - Choose an initial guess for θ
  - Repeatedly update θ to make $L(\theta)$ smaller



Minimizing the cost function

- **Idea**: choose θ to minimize $L(\theta)$
- We can use a search algorithm that follows the scheme:
  - Choose an initial guess for θ
  - Repeatedly update θ to make $L(\theta)$ smaller
  - Keep doing this until $L(\theta)$ reaches its minimum



Updating θ to minimize $L(\theta)$

- **Note**: $L(\theta)$ is a convex quadratic function (has nice properties)

Updating θ to minimize $L(\theta)$

- **Note**: $L(\theta)$ is a convex quadratic function (has nice properties)
- From Math 53: the direction of greatest increase is the same direction of the gradient vector

Updating θ to minimize $L(\theta)$

- **Note**: $L(\theta)$ is a convex quadratic function (has nice properties)
- From Math 53: the direction of greatest increase is the same direction of the gradient vector
- **Idea**: let's update θ by traversing the opposite direction instead

Updating θ to minimize $L(\theta)$

- **Note**: $L(\theta)$ is a convex quadratic function (has nice properties)
- From Math 53: the direction of greatest increase is the same direction of the gradient vector
- **Idea**: let's update θ by traversing the opposite direction instead
- This scheme is known as **gradient descent**

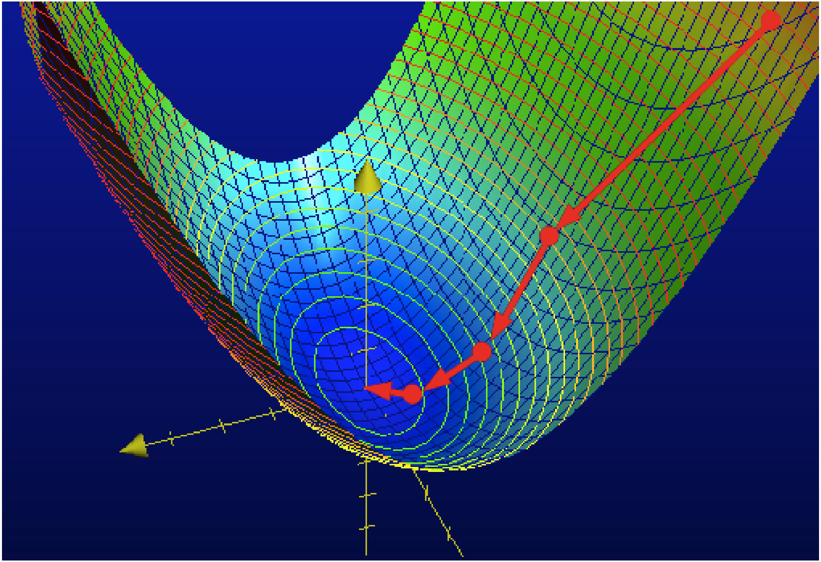
$$\theta \leftarrow \theta - \epsilon \nabla_{\theta} L(\theta)$$

- **Note**: $L(\theta)$ is a convex quadratic function (has nice properties)
- From Math 53: the direction of greatest increase is the same direction of the gradient vector
- **Idea**: let's update θ by traversing the opposite direction instead
- This scheme is known as **gradient descent**

$$\theta \leftarrow \theta - \epsilon \nabla_\theta L(\theta)$$

- ϵ is called the **learning rate**

Visualizing gradient descent



Deriving the update rule

- Let's start with the case where we only have one training example (x, y)

$$\begin{aligned}
\nabla_\theta L(\theta) &= \nabla_\theta \frac{1}{2}(h_\theta(x) - y)^2 \\
&= 2(\frac{1}{2})(h_\theta(x) - y)\nabla_\theta(h_\theta(x) - y) \\
&= (h_\theta(x) - y)\nabla_\theta(\theta^T x - y) \\
&= (h_\theta(x) - y)x
\end{aligned}$$

- Let's start with the case where we only have one training example (x, y)

$$
\begin{aligned}
\nabla_\theta L(\theta) &= \nabla_\theta \frac{1}{2}(h_\theta(x) - y)^2 \\
&= 2(\frac{1}{2})(h_\theta(x) - y)\nabla_\theta(h_\theta(x) - y) \\
&= (h_\theta(x) - y)\nabla_\theta(\theta^T x - y) \\
&= (h_\theta(x) - y)x
\end{aligned}
$$

- For a single training example, the update rule is:

$$
\theta \leftarrow \theta - \epsilon(y^1 - h_\theta(x^1))x^1
$$

Gradient descent for linear regression

- For n training examples:

$$\theta_j \leftarrow \theta_j - \epsilon \sum_{i}^{n} (h_\theta(x^i) - y^i) x_j^i$$

for $j = 1, \ldots, p$ and $i = 1, \ldots, n$

Gradient descent for linear regression

- For n training examples:

$$\theta_j \leftarrow \theta_j - \epsilon \sum_i^n (h_\theta(x^i) - y^i) x_j^i$$

for $j = 1, \ldots, p$ and $i = 1, \ldots, n$

- This rule is also called the **LMS** update rule ("least mean squares")

Gradient descent for linear regression

- For n training examples:

$$\theta_j \leftarrow \theta_j - \epsilon \sum_i^n (h_\theta(x^i) - y^i)x_j^i$$

  for $j = 1, \ldots, p$ and $i = 1, \ldots, n$

- This rule is also called the **LMS** update rule ("least mean squares")
- Size of update is proportional to the residual term $(y^i - h_\theta(x^i))$

Gradient descent for linear regression

- For n training examples:

$$\theta_j \leftarrow \theta_j - \epsilon \sum_{i}^{n} (h_\theta(x^i) - y^i) x_j^i$$

for $j = 1, \ldots, p$ and $i = 1, \ldots, n$

- This rule is also called the **LMS** update rule ("least mean squares")
- Size of update is proportional to the residual term $(y^i - h_\theta(x^i))$
- If the prediction $h_\theta(x^i)$ is close the actual y^i then the parameters θ shouldn't need much changing

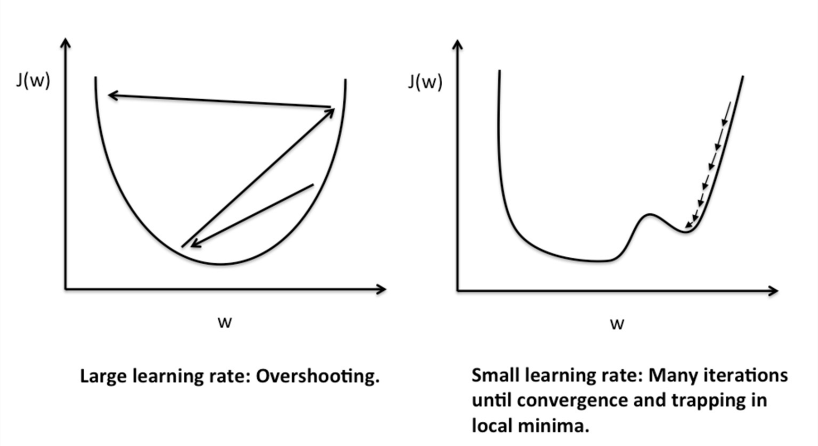
Stochastic gradient descent for linear regression

While $L(\theta)$ is not minimized:

For $i = i, \ldots, n$:

$$\theta_j \leftarrow \theta_j - \epsilon(h_\theta(x^i) - y^i)x_j^i \quad \text{(for each j)}$$

Choosing the learning rate



Linear regression can be solved in an easier way...

- SGD is the basis for many optimization algorithms

Linear regression can be solved in an easier way...

- SGD is the basis for many optimization algorithms
- Not necessary for linear regression, as there exists a closed form solution

Linear regression can be solved in an easier way...

- SGD is the basis for many optimization algorithms
- Not necessary for linear regression, as there exists a closed form solution
- Useful when derivatives are easy to calculate, but there is no closed form solution

Linear regression can be solved in an easier way...

- SGD is the basis for many optimization algorithms
- Not necessary for linear regression, as there exists a closed form solution
- Useful when derivatives are easy to calculate, but there is no closed form solution
- We can find the optimal θ by solving the **normal equations** as covered previously

Probabilistic Perspective

Assumptions

- Regression is a good summary of data, assuming the data has some key properties

Assumptions

- Regression is a good summary of data, assuming the data has some key properties
- We need to know what those assumptions are, where they come from, and what to do when they fall apart

Assumptions: what are they?

- Linearity

Assumptions: what are they?

- Linearity
- Normality of errors

$$\epsilon_i \sim N(0, \sigma^2)$$

Assumptions: what are they?

- Linearity
- Normality of errors

$$\epsilon_i \sim N(0, \sigma^2)$$

- Homoscedasticity (constant variance)

$$Var(\epsilon_i) = Var(\epsilon_j)$$

Assumptions: what are they?

- Linearity
- Normality of errors

$$\epsilon_i \sim N(0, \sigma^2)$$

- Homoscedasticity (constant variance)

$$Var(\epsilon_i) = Var(\epsilon_j)$$

- Independence of errors

$$\epsilon_i \perp\!\!\!\perp \epsilon_j \qquad \forall i \neq j$$

Data Generation Process

- The real world has natural processes. These processes can be collected/observed as data

Data Generation Process

- The real world has natural processes. These processes can be collected/observed as data
- We try to build a model that mimics the real world model as close as possible

Data Generation Process

- The real world has natural processes. These processes can be collected/observed as data
- We try to build a model that mimics the real world model as close as possible
- Problem: there are some factors we can directly observe, and others that we can't

Data Generation Process

- The real world has natural processes. These processes can be collected/observed as data
- We try to build a model that mimics the real world model as close as possible
- Problem: there are some factors we can directly observe, and others that we can't
- Problem: we don't know what model the world uses

Data Generation Process

- The real world has natural processes. These processes can be collected/observed as data
- We try to build a model that mimics the real world model as close as possible
- Problem: there are some factors we can directly observe, and others that we can't
- Problem: we don't know what model the world uses
- In order to reason this process more rigorously, we can construct a "toy universe" to analyze our models

Data Generation Process

In our toy universe, data is generated through a linear model:

$$Y = X\theta$$

Data Generation Process

In our toy universe, data is generated through a linear model:

$$Y = X\theta$$

We can observe these Y values, but our observations have some noise in them. So our collected data looks like this:

$$Y = X\theta + Z \text{ where } Z \sim \mathcal{N}(0, \sigma^2)$$

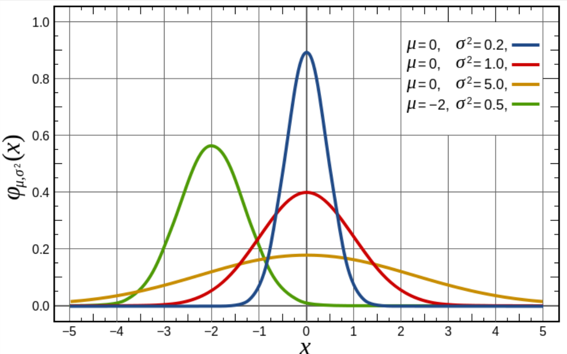
Data Generation Process

In our toy universe, data is generated through a linear model:

$$Y = X\theta$$

We can observe these Y values, but our observations have some noise in them. So our collected data looks like this:

$$Y = X\theta + Z \text{ where } Z \sim \mathcal{N}(0, \sigma^2)$$



Data Generation Process

So we have our noisy data:

$$Y = X\theta + Z$$

Data Generation Process

So we have our noisy data:

$$Y = X\theta + Z$$

- We want to find the θ that can best replicate the data we've already observed

Data Generation Process

So we have our noisy data:

$$Y = X\theta + Z$$

- We want to find the θ that can best replicate the data we've already observed
- Aka we want to **increase the likelihood** of the observed data being generated by the model

Data Generation Process

So we have our noisy data:

$$Y = X\theta + Z$$

- We want to find the θ that can best replicate the data we've already observed
- Aka we want to **increase the likelihood** of the observed data being generated by the model

Data Generation Process

So we have our noisy data:

$$Y = X\theta + Z$$

- We want to find the θ that can best replicate the data we've already observed
- Aka we want to **increase the likelihood** of the observed data being generated by the model

$$\hat{\theta} = argmax_{\theta} P(y_1, y_2, ..., y_n | \theta, x_1, x_2, ..., x_n)$$

Maximum Likelihood

$$\hat{\theta} = argmax_{\theta} P(y_1, y_2, ..., y_n | \theta, x_1, x_2, ..., x_n)$$

Maximum Likelihood

$$\hat{\theta} = argmax_{\theta} P(y_1, y_2, ..., y_n | \theta, x_1, x_2, ..., x_n)$$

$$\hat{\theta} = argmax_{\theta} \prod_i P(y_i | \theta, x_i) = argmax_{\theta} \sum_i \log P(y_i | \theta, x_i)$$

Maximum Likelihood

$$\hat{\theta} = argmax_{\theta} P(y_1, y_2, ..., y_n | \theta, x_1, x_2, ..., x_n)$$

$$\hat{\theta} = argmax_{\theta} \prod_i P(y_i | \theta, x_i) = argmax_{\theta} \sum_i \log P(y_i | \theta, x_i)$$

Remember that $y_i \sim \mathcal{N}(\theta^T x_i, \sigma^2)$:

Maximum Likelihood

$$\hat{\theta} = argmax_{\theta} P(y_1, y_2, ..., y_n | \theta, x_1, x_2, ..., x_n)$$

$$\hat{\theta} = argmax_{\theta} \prod_i P(y_i | \theta, x_i) = argmax_{\theta} \sum_i \log P(y_i | \theta, x_i)$$

Remember that $y_i \sim \mathcal{N}(\theta^T x_i, \sigma^2)$:

$$\hat{\theta} = argmax_{\theta} \sum_i \log \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - \theta x_i)^2}{2\sigma^2}}$$

Maximum Likelihood

$$\hat{\theta} = argmax_\theta P(y_1, y_2, ..., y_n|\theta, x_1, x_2, ..., x_n)$$

$$\hat{\theta} = argmax_\theta \prod_i P(y_i|\theta, x_i) = argmax_\theta \sum_i \log P(y_i|\theta, x_i)$$

Remember that $y_i \sim \mathcal{N}(\theta^T x_i, \sigma^2)$:

$$\hat{\theta} = argmax_\theta \sum_i \log \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - \theta x_i)^2}{2\sigma^2}}$$

$$\hat{\theta} = argmax_\theta \sum_i \log e^{-\frac{(y_i - \theta x_i)^2}{2\sigma^2}} = argmax_\theta \sum_i -\frac{(y_i - \theta x_i)^2}{2\sigma^2}$$

Maximum Likelihood

$$\hat{\theta} = argmax_{\theta} P(y_1, y_2, ..., y_n | \theta, x_1, x_2, ..., x_n)$$

$$\hat{\theta} = argmax_{\theta} \prod_i P(y_i | \theta, x_i) = argmax_{\theta} \sum_i \log P(y_i | \theta, x_i)$$

Remember that $y_i \sim \mathcal{N}(\theta^T x_i, \sigma^2)$:

$$\hat{\theta} = argmax_{\theta} \sum_i \log \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - \theta x_i)^2}{2\sigma^2}}$$

$$\hat{\theta} = argmax_{\theta} \sum_i \log e^{-\frac{(y_i - \theta x_i)^2}{2\sigma^2}} = argmax_{\theta} \sum_i -\frac{(y_i - \theta x_i)^2}{2\sigma^2}$$

$$\boxed{\hat{\theta} = argmin_{\theta} \sum_i (y_i - \theta^T x_i)^2}$$

Maximum Likelihood

$$\hat{\theta} = argmax_\theta P(y_1, y_2, ..., y_n | \theta, x_1, x_2, ..., x_n)$$

$$\hat{\theta} = argmax_\theta \prod_i P(y_i | \theta, x_i) = argmax_\theta \sum_i \log P(y_i | \theta, x_i)$$

Remember that $y_i \sim \mathcal{N}(\theta^T x_i, \sigma^2)$:

$$\hat{\theta} = argmax_\theta \sum_i \log \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - \theta x_i)^2}{2\sigma^2}}$$

$$\hat{\theta} = argmax_\theta \sum_i \log e^{-\frac{(y_i - \theta x_i)^2}{2\sigma^2}} = argmax_\theta \sum_i -\frac{(y_i - \theta x_i)^2}{2\sigma^2}$$

$$\boxed{\hat{\theta} = argmin_\theta \sum_i (y_i - \theta^T x_i)^2}$$

Assumptions: what do we do if they are not satisfied?

- If the data is nonlinear...

Assumptions: what do we do if they are not satisfied?

- If the data is nonlinear...
  - Try performing a transformation on the independent or dependent variables such as squaring it, taking the log or square root, or ...

Assumptions: what do we do if they are not satisfied?

- If the data is nonlinear...
  - Try performing a transformation on the independent or dependent variables such as squaring it, taking the log or square root, or ...
- If the errors are not normal...

Assumptions: what do we do if they are not satisfied?

- If the data is nonlinear...
  - Try performing a transformation on the independent or dependent variables such as squaring it, taking the log or square root, or ...
- If the errors are not normal...
  - Often, this isn't a big problem

Assumptions: what do we do if they are not satisfied?

- If the data is nonlinear...
  - Try performing a transformation on the independent or dependent variables such as squaring it, taking the log or square root, or ...
- If the errors are not normal...
  - Often, this isn't a big problem
  - Transformations help here too

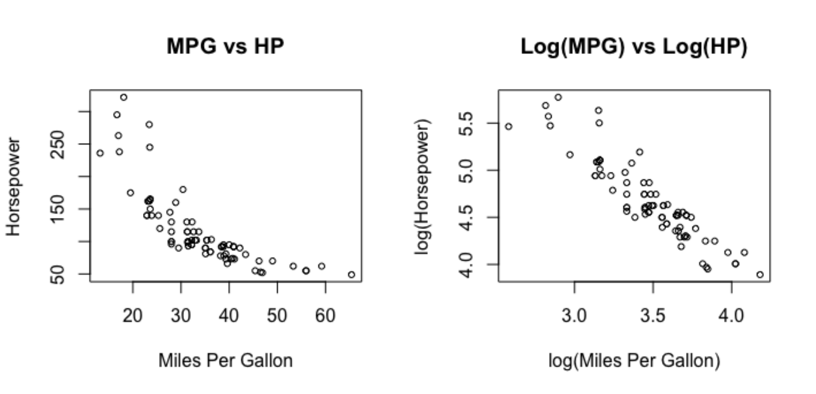
Assumptions: what do we do if they are not satisfied?

- If the data is nonlinear...
  - Try performing a transformation on the independent or dependent variables such as squaring it, taking the log or square root, or ...
- If the errors are not normal...
  - Often, this isn't a big problem
  - Transformations help here too
  - Maybe subsets of the data are more normal than the overall set

Assumptions: what do we do if they are not satisfied?

- If the data is nonlinear...
  - Try performing a transformation on the independent or dependent variables such as squaring it, taking the log or square root, or ...
- If the errors are not normal...
  - Often, this isn't a big problem
  - Transformations help here too
  - Maybe subsets of the data are more normal than the overall set
  - Outliers and/or high leverage points may contribute to this issue

Example of the beauty of a log transform



Lecture Recap

- Introduced linear regression

Lecture Recap

- Introduced linear regression
- Used linear algebra to derive a closed form solution

Lecture Recap

- Introduced linear regression
- Used linear algebra to derive a closed form solution
- Used gradient descent to iteratively optimize to a solution

Lecture Recap

- Introduced linear regression
- Used linear algebra to derive a closed form solution
- Used gradient descent to iteratively optimize to a solution
- Constructed a universe in which to better understand our model and the assumptions behind it

Questions?