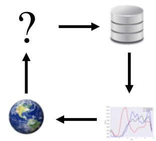
Data Science 100
Final Review (Part 1)

Slides by:
Joseph E. Gonzalez, Deb Nolan, & Fernando Perez
jegonzal@berkeley.edu

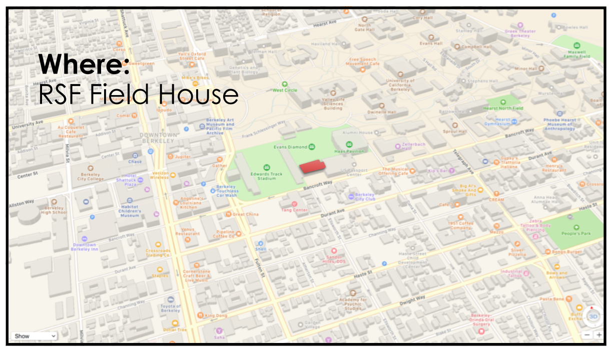


Logistics

When:
<u>*8:00AM*</u> – 11:00AM Thursday, May 10th

- That is so early!
  - We agree!
- Set an alarm
  - Set a second alarm
  - Call a friend and ask them to set an alarm
  - Go to bed at a reasonable hour

Where:
RSF Field House



What to Bring

- Cal ID Card
- Pencils and Erasers
- A two page study guide (more on this in a moment)
- No food or drink is allowed in RSF Fieldhouse

How to make a **Study Guide**

- We don't call it a cheat sheet. Why?
  - **Cheating is bad** … Don't cheat.
  - **Goal:** after you make it you don't need it
- You could just miniaturize all the lectures but this would not help you study.
- Go over lectures, HWS, projections, sections, and labs
  - Try to explain the material to your friends (real and imagined)
  - Write big concepts, technical ideas, terminology, & definitions.
  - Think about how things are arranged.
- You should be able to explain everything on your guide

What is the format?

- Same format as the midterm: *largely* ***multiple choice*** *and* ***very short answer***
- You **will not** need to write long programs
- You **will need** to read Python, SQL, and Regular expressions (find bugs, explain what they do, match with output …)
- For Python APIs and Regex syntax we will provide a **reference sheet** (same as midterm).

What is covered on the final?

- Everything!

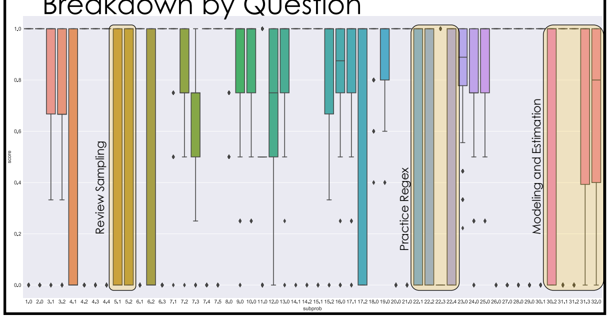
  … except Apache Spark ☹ [which I really like]

  … but you should know MapReduce concepts ☺
- This includes material before the midterm. (Review the midterm!)
- This exam review covers material up to the midterm
- Thursday will cover material after the midterm

Material Before the Midterm

- Data Sampling and Collection
- Pandas Indexes, DataFrames Series, Pivot Tables, Group By, and Merge
- Exploratory Data Analysis and Data Cleaning
- Data Visualization and plotting
- Web technologies (http and requests)
- Regular Expressions
- SQL
- Modeling and Estimation (Loss functions)
- Gradient Descent

Breakdown by Question



Sampling the Population

Data Collection and Sampling

- **Census:** the *complete* ***population of interest***
  - Important to identify the population of interest

Probability Samples:

- **Simple Random Sample (SRS):** a random subset where every subset has equal chance of being chosen
- **Stratified Sample:** population is partition into strata and a SRS is taken within each strata
  - Samples from each strata don't need to be the same size
- **Cluster Sample:** divide population into groups, take an SRS of groups, and elements from each group are selected
  - Often take all elements (one-stage) may sample within groups (two-stage)

Non Probability Samples

- **Administrative Sample**: *data collected to support an administrative purpose and not for research*
  - Bigger isn't always better → bias still an issue at scale
- **Voluntary Sample:** self-selected participation
  - Sensitive to self selection bias
- **Convenience Sample:** the data you have …
  - often administrative

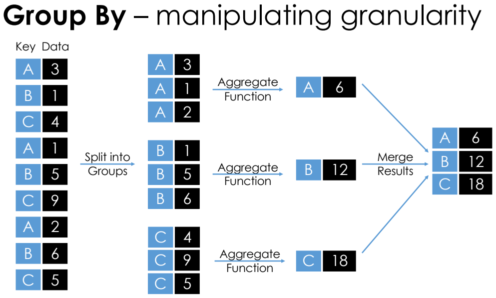
Code

Python + Numpy + Pandas + Seaborn + SQL + Regex +HTTP

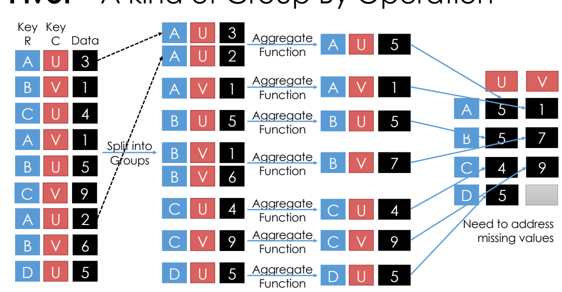
Pandas

- Review *column selection* and *Boolean slicing on rows*
- Review ***groupby***, **merge**, and **pivot_table**:
  - *df.groupby(['state', 'gender'])[['age', 'height']].mean()*
  - dfA.merge(dfB, on='key', how='outer')
  - df.pivot_table(index, columns, values, aggfunc, fill_value)
- Understand rough usage of basic plotting commands
  - plot, bar, histogram …
  - sns.distplot

Group By – manipulating granularity



Pivot – A kind of Group By Operation



Joining data across tables

- Joins are a way to connect data across multiple tables

Purchases.csv

OrderNum	ProdID	Quantity
1	**42**	3
1	**999**	2
2	**42**	1

⋈ Join

Products.csv

ProdID	Cost
42	3.14
999	2.72

=

Joined Table

OrderNum	ProdID	Quantity	Cost
1	**42**	3	3.14
1	**999**	2	2.72
2	**42**	1	3.14

EDA & Data Visualization

Kinds of Data

Note that categorical data can also be numbers and quantitative data may be stored as strings.

Quantitative Data

Numbers with meaning ratios or intervals.

Examples:
- Price
- Quantity
- Temperature
- Date
- …

Categorical Data

Ordinal

Categories with orders but no consistent meaning if magnitudes or intervals

Examples:
- Preferences
- Level of education
- …

Nominal

Categories with no specific ordering.

Examples:
- Political Affiliation
- CalD number
- …

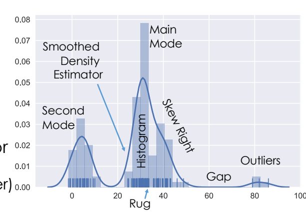
Visualizing Univariate Relationships

- **Quantitative Data**
  - Histograms, Box Plots, Rug Plots, Smoothed Interpolations (KDE – Kernel Density Estimators)
  - Look for symmetry, skew, spread, modes, gaps, outliers…
- **Nominal & Ordinal Data**
  - Bar plots (sorted by frequency or ordinal dimension)
  - Look for skew, frequent and rare categories, or invalid categories
  - Consider grouping categories and repeating analysis

Histograms, Rug Plots, and KDE Interpolation

Describes distribution of data – relative prevalence of values

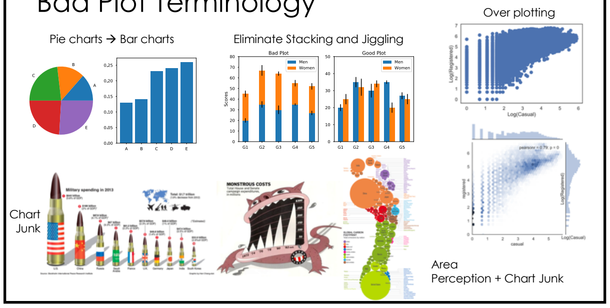
- Histogram
  - relative frequency of values
  - Tradeoff of bin sizes
- Rug Plot
  - Shows the actual data locations
- Smoothed density estimator
  - Tradeoff of "bandwidth" parameter (more on this later)



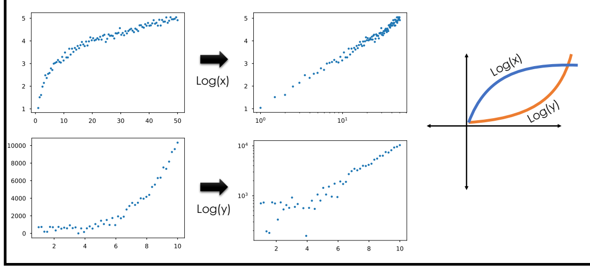
Techniques of Visualization

- **Scale:** ranges of values and how they are presented
  - Units, starting points, zoom, …
- **Conditioning:** breakdown visualization across dimensions for comparison (e.g., separate lines for males and females)
- **Perception**
  - **Length**: encode relative magnitude (best for comparison)
  - **Color**: encode conditioning and additional dimensions and
- **Transformations:** to linearize relationships highlight important trends
  - Symmetrize distribution
  - Linearize relationships (e.g., Tukey Mosteller Bulge)
- Things to avoid stacking, jiggling, chart junk, and over plotting

Bad Plot Terminology



Linearizing Relationships

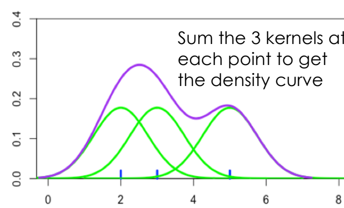


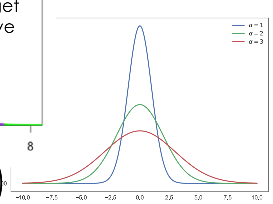
Dealing with Big Data

- **Big n** (many rows)
  - Aggregation & Smoothing – compute summaries over groups/regions
    - Sliding windows, kernel density smoothing
  - Set transparency or use contour plots to avoid over-plotting
- **Big p** (many columns)
  - Create new hybrid columns that summarize multiple columns
    - **Example**: total sources of revenue instead of revenue by product
  - Use dimensionality reduction techniques to automatically derive columns that preserve the relationships between records (e.g., distances)
    - PCA – not required to know PCA for the exam.

Kernel Density Estimator

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} K_\alpha(x - x_i)$$





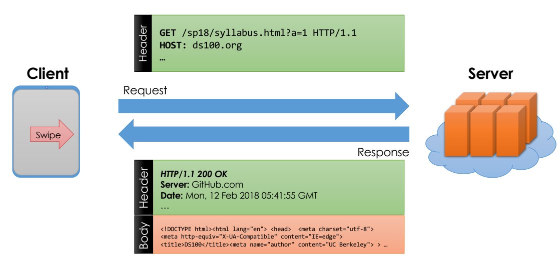
$$K_\alpha(r) = \frac{1}{\sqrt{2\pi\alpha^2}} \exp\left(-\frac{r^2}{2\alpha^2}\right)$$

Gaussian Kernels

Web Technologies

XML/JSON/HTTP/REST

Request – Response Protocol



Request Types (Main Types)

- Know differences between put and get
- **GET** – *get information*
  - Parameters passed in URI (limited to ~2000 characters)
    - `/app/user_info.json?username=mejoeyg&version=now`
    - Request body is typically ignored
  - Should not have side-effects (e.g., update user info)
  - Can be cached in on server, network, or in browser (bookmarks)
- **POST** – *send information*
  - Parameters passed in URI and BODY
  - May and typically will have side-effects
  - Often used with web forms.

HTML/XML/JSON

- Most services will exchange data in HTML, XML, or JSON
- Nested data formats (review JSON notebook)
  - Understand how JSON objects map to python objects (HWs)
    - JSON List → Python List
    - JSON Dictionary → Python Dictionary
    - JSON Literal → Python Literal
- Review basic XML formatting requirements:
  - Well nested tags, no spaces, case sensitive,
- Be able to read XML and JSON and identify basic bugs

String Manipulation and Regular Expressions

Regex Reference Sheet

^ match beginning of string (unless used for negation [^ …])

$ match end of string character

? match preceding character or subexpression at most once

+ match preceding character or subexpression one or more times

***** match preceding character or subexpression zero or more times

. matches any character **except newline**

[ ] match any single character inside
- match a range of characters [a-c]

( ) used to create sub-expressions

\b match boundary between words

\w match a "word" character (letters, digits, underscore). \W is the complement

\s match a whitespace character including tabs and newlines. \S is the complement

\d match a digit. \D is the complement

You should know these.

Greedy Matching

- **Greedy matching: *** and **+** match as many characters as possible using the preceding subexpression in the regular expression before going to the next subexpression.
- Example
  - `<.*>` matches <body>text</body>
- ? The modifier suffix makes * and **+** non-greedy.
  - `<.*?>` matches <body>text</body>

Suggested Practice

- https://www.w3resource.com/python-exercises/re/
- Try running regular expression on the midterm through:
  - https://regex101.com/
  - Don't forget to switch to python mode.
- `r”\d\d/\d\d/\d{4}”`
  - Dates
- `r”\w*’\w”`
  - Don’t
- `r”[Aa]naly[zs]e”`
  - Analyze Analyse

SQL

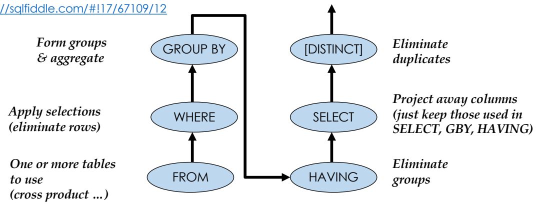
Relational Terminology

- *Database*: Set of Relations (i.e., one or more tables)
- *Attribute* (*Column*)
- *Tuple* (*Record, Row*)
- *Relation* (*Table*):
  - *Schema*: the set of column names, their types, and any constraints
  - *Instance*: data satisfying the schema
- *Schema of database* is set of schemas of its relations

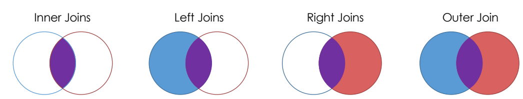
Conceptual SQL Evaluation

```
SELECT     [DISTINCT]  target-list
FROM       relation-list
WHERE      qualification
GROUP BY   grouping-list
HAVING     group-qualification
```

Try Queries Here
http://sqlfiddle.com/#!17/67109/12



Kinds of Joins



Review the slides and syntax for each join type

```
SELECT r.sid, b.bid, b.bname
FROM Reserves3 r FULL JOIN Boats2 b
ON r.bid = b.bid
```

Reserves3

sid	bid	day
22	101	1996-10-10
95	103	1996-11-12
38	42	2010-08-21

Boats2

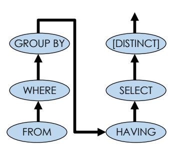
bid	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

Result:

sid	bid	bname
22	101	Interlake
95	103	Clipper
38	**(null)**	**(null)**
(null)	104	Marine
(null)	102	Interlake

Putting it all together

```
SELECT c.name, AVG(g.grade) AS avg_g, COUNT(*) AS size
  FROM grades AS g, classes AS c
 WHERE g.class_id = c.class_id AND
       g.year = "2006"
 GROUP BY g.class_id
HAVING COUNT(*) > 2
 ORDER BY avg_g DESC
```



What does this compute?

Modeling and Estimation

Summary of Model Estimation

1. **Define the Model:** simplified representation of the world
   - Use domain knowledge but … ***keep it simple!***
   - Introduce **parameters** for the unknown quantities
2. **Define the Loss Function:** measures how well a particular instance of the model "fits" the data
   - We introduced L^2, L^1, and Huber losses for each record
   - Take the average loss over the entire dataset
3. **Minimize the Loss Function:** find the parameter values that minimize the loss on the data
   - Analytically using calculus
   - Numerically using gradient descent

Linear Models

One of the most widely used tools in machine learning and data science

Model

$$\hat{y} = f_\theta(x) = \sum_{j=1}^{d} \theta_j \phi_j(x)$$

Linear in the Parameters

Feature Functions

Loss Minimization

$$\hat{\theta} = \arg\min \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} \theta_j \phi_j(x_i) \right)^2$$

Squared Loss

We will return to solving this soon!

Linear Models and Feature Functions

$$\hat{y} = f_\theta(x) = \sum_{j=1}^{d} \theta_j \phi_j(x)$$

Linear in the Parameters

Feature Functions

Designing the feature functions is a big part of machine learning and data science.

Feature Functions

- capture domain knowledge
- substantial contribute to expressivity (and complexity)

Linear Models and Feature Functions

$$\hat{y} = f_\theta(x) = \sum_{j=1}^{d} \theta_j \phi_j(x)$$

Linear in the Parameters

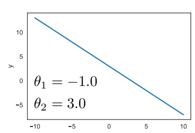
Feature Functions

For Example: Domain: $x \in \mathbb{R}$ Model: $f_\theta(x) = \theta_1 x + \theta_2$

Features:

$\phi_1(x) = x$

$\phi_2(x) = 1$



Adding a **"constant" feature function** $\phi_2(x) = 1$ is a common method to introduce an **offset** (also sometimes called **bias**) term.

Linear Models and Feature Functions

$$\hat{y} = f_\theta(x) = \sum_{j=1}^{d} \theta_j \phi_j(x)$$

Linear in the Parameters

Feature Functions

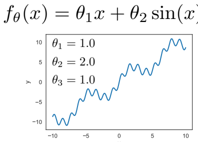
For Example: $x \in \mathbb{R}$ $f_\theta(x) = \theta_1 x + \theta_2 \sin(x) + \theta_3 \sin(5x)$

Features:

$\phi_1(x) = x$

$\phi_2(x) = \sin(x)$

$\phi_3(x) = \sin(5x)$



← This is a linear model!

Linear in the parameters

Linear Models and Feature Functions

$$\hat{y} = f_\theta(x) = \sum_{j=1}^{d} \theta_j \phi_j(x)$$

Linear in the Parameters

Feature Functions

For Example: $x \in \mathbb{R}^2$

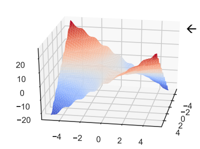
$$f_\theta(x) = \theta_1 x_1 x_2 + \theta_2 \cos(x_2 x_1) + \theta_3 \mathbb{I}[x_1 > x_2]$$

Features:

$\phi_1(x) = x_1 x_2$

$\phi_2(x) = \cos(x_2 x_1)$

$\phi_3(x) = \mathbb{I}[x_1 > x_2]$



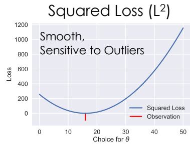
← This is a linear model!

Linear in the parameters

Loss Functions

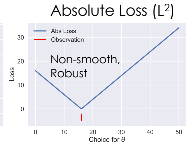
- **Loss function**: a function that characterizes the cost, error, or loss resulting from a choice of model and parameters.

Squared Loss (L^2)



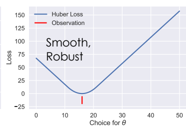
$$L(\theta, y) = (y - \theta)^2$$

Absolute Loss (L^2)



$$L(\theta, y) = |y - \theta|$$

Huber Loss



$$L_\alpha(\theta, y) = \begin{cases} \frac{1}{2}(y-\theta)^2 & |y-\theta| < \alpha \\ \alpha\left(|y-\theta| - \frac{\alpha}{2}\right) & \text{otherwise} \end{cases}$$

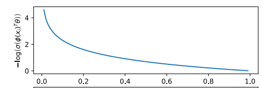
The Average Cross Entropy Loss

This came after the Midterm but is reviewed here for fun

$$L(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log\left(\sigma\left(\phi(x_i)^T\theta\right)\right) + (1-y_i)\log\left(1-\sigma\left(\phi(x_i)^T\theta\right)\right)\right)$$

- If $y_i = 1$

$$-\log\left(\sigma\left(\phi(x_i)^T\theta\right)\right)$$

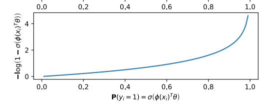


- If $y_i = 0$

Prb $y_i = 1$

$$-\log\left(1-\sigma\left(\phi(x_i)^T\theta\right)\right)$$

Prb $y_i = 0$

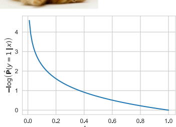


- Average **cross entropy loss**

$$\arg\min_\theta \frac{1}{n}\sum_{i=1}^{n}\sum_{k=0}^{K-1} -\mathbf{P}(y_i = k \mid x_i)\log\left(\hat{\mathbf{P}}_\theta(y_i = k \mid x_i)\right)$$

Cute Cat Example

x =  , y = 1 "cute"



	Cute	Not Cute
Observed Probability	$\mathbf{P}(y=1 \mid x)$ = 1.0	$\mathbf{P}(y=0 \mid x)$ = 0.0
Predicted Probability	$\hat{\mathbf{P}}_\theta(y=1 \mid x)$ = 0.8	$\hat{\mathbf{P}}_\theta(y=0 \mid x)$ = 0.2
Cross Ent. $-\mathbf{P}\log\hat{\mathbf{P}}_\theta$	-1.0 Log(0.8) ≈ 0.22	-0.0 Log(0.2) = 0.0

Also called the log loss because it is the log of the predicted probability for the true class

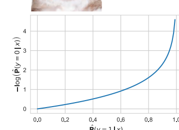
---

- Average **cross entropy loss**

$$\arg\min_\theta \frac{1}{n}\sum_{i=1}^{n}\sum_{k=0}^{K-1} -\mathbf{P}(y_i = k \mid x_i)\log\left(\hat{\mathbf{P}}_\theta(y_i = k \mid x_i)\right)$$

Cute Cat Example

x =  , y = 0 "not cute"



	Cute	Not Cute
Observed Probability	$\mathbf{P}(y=1 \mid x)$ = 0.0	$\mathbf{P}(y=0 \mid x)$ = 1.0
Predicted Probability	$\hat{\mathbf{P}}_\theta(y=1 \mid x)$ = 0.7	$\hat{\mathbf{P}}_\theta(y=0 \mid x)$ = 0.3
Cross Ent. $-\mathbf{P}\log\hat{\mathbf{P}}_\theta$	-0.0 Log(0.7) = 0.0	-1.0 Log(0.3) ≈ 1.20

Also called the log loss because it is the log of the predicted probability for the true class

Example: Minimizing Average L^2 Loss

Average Loss (L^2)

1. $$L_\mathcal{D}(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta)^2$$

Derivative of the Average Loss (L^2)

2. $$\frac{\partial}{\partial\theta}L_\mathcal{D}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\theta}(y_i - \theta)^2$$

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i - \theta)$$

Set derivative = 0 and solve for θ…

3. $$0 = -\frac{2}{n}\sum_{i=1}^{n}(y_i - \theta)$$

$$0 = \left(\sum_{i=1}^{n} y_i\right) - n\theta$$

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Essential Calculus: The Chain Rule

- How do I compute the derivative of composed functions?

$$\frac{\partial}{\partial\theta}h(\theta) = \frac{\partial}{\partial\theta}f(g(\theta))$$

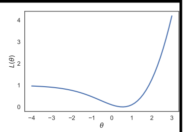
$$= \left(\frac{\partial}{\partial u}f(u)\Big|_{u=g(\theta)}\right)\frac{\partial}{\partial\theta}g(\theta)$$

Derivative of f evaluated at g(θ)

Derivative of g(θ)

Know how to calculate derivatives of logs, exponents, and exponentials.

Exercise of Calculus

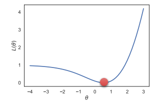


- Minimize: $L(\theta) = (1 - \log(1 + \exp(\theta)))^2$
- Take the derivative:

$$\begin{aligned}\frac{\partial}{\partial\theta}L(\theta) &= \frac{\partial}{\partial\theta}(1 - \log(1+\exp(\theta)))^2 \\ &= 2(1-\log(1+\exp(\theta)))\frac{\partial}{\partial\theta}(1-\log(1+\exp(\theta))) \\ &= 2(1-\log(1+\exp(\theta)))(-1)\frac{\partial}{\partial\theta}\log(1+\exp(\theta)) \\ &= 2(1-\log(1+\exp(\theta)))\frac{-1}{1+\exp(\theta)}\frac{\partial}{\partial\theta}(1+\exp(\theta)) \\ &= 2(1-\log(1+\exp(\theta)))\frac{-1}{1+\exp(\theta)}\exp(\theta)\end{aligned}$$

- Take the derivative:

$$\begin{aligned}\frac{\partial}{\partial\theta}L(\theta) &= 2(1-\log(1+\exp(\theta)))\frac{-1}{1+\exp(\theta)}\exp(\theta) \\ &= -2(1-\log(1+\exp(\theta)))\frac{\exp(\theta)}{1+\exp(\theta)}\end{aligned}$$



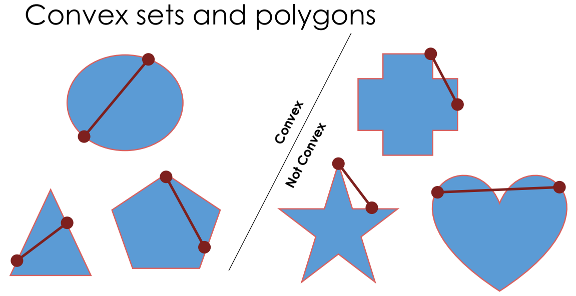
- Set derivative equal to zero and solve for parameter

$$-2(1-\log(1+\exp(\theta)))\frac{\exp(\theta)}{1+\exp(\theta)} = 0 \;\Rightarrow\; 1-\log(1+\exp(\theta)) = 0$$

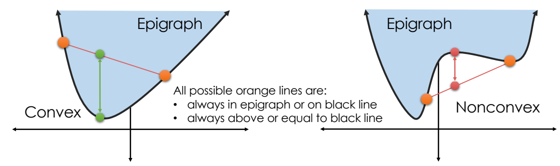
Solving for parameters

$$\begin{aligned}\log(1+\exp(\theta)) &= 1 \\ 1+\exp(\theta) &= \exp(1) \\ \exp(\theta) &= \exp(1) - 1 \\ \theta &= \log(\exp(1)-1) \approx 0.541\end{aligned}$$

Convex sets and polygons



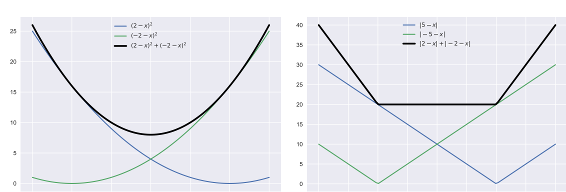
Formal Definition of Convex Functions



- A function f is convex if and only if:

$$tf(a) + (1-t)f(b) \geq f(ta + (1-t)b)$$
$$\forall a,\ \forall b,\ t \in [0,1]$$

Sum of Convex Functions is Convex



Bonus material (not covered in lecture) but useful for studying

Formal Proof

- Suppose you have two convex functions *f* and *g*:

$$tf(a) + (1-t)f(b) \geq f(ta - (1-t)a)$$
$$tg(a) + (1-t)g(b) \geq g(ta - (1-t)a)$$
$$\forall a,\ \forall b,\ t \in [0,1]$$

- We would like to show:

$$th(a) + (1-t)h(b) \geq h(ta - (1-t)a)$$

- Where: $h(x) = f(x) + g(x)$

Bonus material (not covered in lecture) but useful for studying

- We would like to show:

$$th(a) + (1-t)h(b) \geq h\left(ta - (1-t)a\right)$$

  - Where: $h(x) = f(x) + g(x)$

- Starting on the left side

Substituting definition of h:

$$th(a) + (1-t)h(b) = t\left(f(a) + g(a)\right) + (1-t)\left(f(b) + g(b)\right)$$

Re-arranging terms: $= [tf(a) + (1-t)f(b)] + [tg(a) + (1-t)g(b)]$

Convexity in f $\geq f\left(ta + (1-t)b\right) + [tg(a) + (1-t)g(b)]$

Convexity in g $\geq f\left(ta + (1-t)b\right) + g\left(ta + (1-t)b\right)$

Definition of h $= h\left(ta + (1-t)b\right)$

□

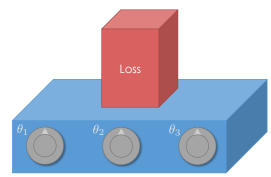
Bonus material (not covered in lecture) but useful for studying

Minimizing the Loss

- Calculus techniques can be applied generally …
- Guaranteed to minimize the loss when **loss** is convex in the parameters
- May not always have an analytic solution …

Gradient Descent

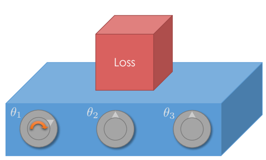
Intuition



Goal: Minimize the loss by turning the knobs.

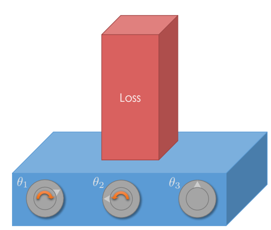
Try the loss game (its free)!

Intuition



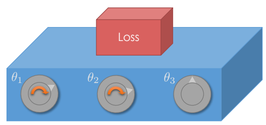
Try the loss game (you can't lose)!

Intuition



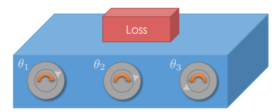
Try the loss game (your loss will be minimal)!

Intuition

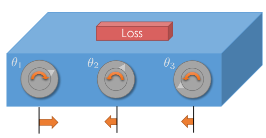


Try the loss game (victory without loss)!

Intuition



Intuition

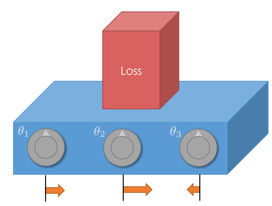


What if we knew which way to turn the knob and an idea of how far?

This is the Gradient!

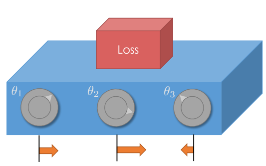
Try the loss game (its free)!

Intuition



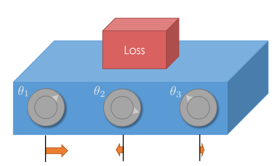
Try the loss game (its free)!

Intuition



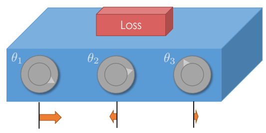
Try the loss game (its free)!

Intuition



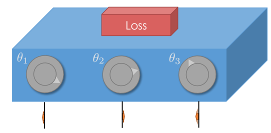
Try the loss game (its free)!

Intuition



Try the loss game (its free)!

Intuition



This is the Gradient descent algorithm!

Try the loss game (its free)!

Quick Review: **Gradients**

Loss function

$$f : \mathbb{R}^p \to \mathbb{R}$$

For Example:

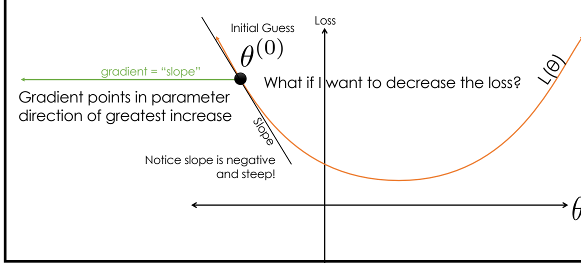
$$f(\theta_1, \theta_2, \theta_3) = a\theta_1 + b\theta_2 + c\theta_2\theta_3^2$$

- Gradient: $g : \mathbb{R}^p \to \mathbb{R}^p$

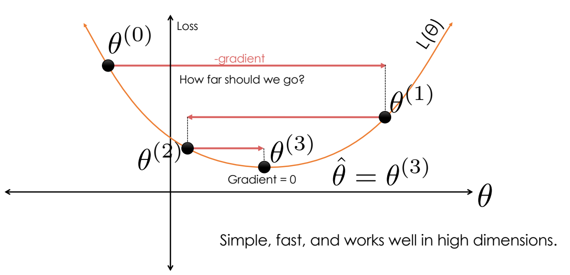
$$\nabla_\theta f(\theta_1, \theta_2, \theta_3) = [a, b + c\theta_3^2, 2c\theta_2\theta_3]$$

$$g(\theta) = \nabla_\theta f(\theta) = \left[\frac{\partial}{\partial \theta_1} f(\theta)|_\theta, \ldots, \frac{\partial}{\partial \theta_3} f(\theta)|_\theta\right]$$

Gradient Descent Intuition



Gradient Descent Intuition



Simple, fast, and works well in high dimensions.

The Gradient Descent Algorithm

$\theta^{(0)} \leftarrow$ initial vector (random, zeros …)

For τ from 0 to convergence:

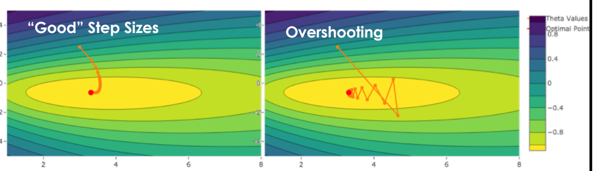
$$\theta^{(\tau+1)} \leftarrow \theta^{(\tau)} - \rho(\tau) \left(\nabla_\theta \mathbf{L}(\theta) \Big|_{\theta=\theta^{(\tau)}} \right)$$

(Evaluated at $\theta=\theta^{(\tau)}$)

- ***ρ(τ)*** is the step size (learning rate)
  - typically $1/\tau$
- Converges when gradient is ≈ 0 (or we run out of patience)

Gradient Descent Solution Paths

- Orange line is path taken by gradient descent
  - Contours are from loss on two parameter model



Stochastic Gradient Descent

This came after the Midterm but is reviewed here because it makes sense.

- For many learning problems the gradient is a sum:

$$\nabla_\theta \mathbf{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\sigma\left(\phi(x_i)^T\theta\right) - y_i\right)\phi(x_i)$$

  - For large n this can be **expensive to compute**
- What if we approximated the gradient by looking at a few random points:

$$\nabla_\theta \mathbf{L}(\theta) \approx \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\left(\sigma\left(\phi(x_i)^T\theta\right) - y_i\right)\phi(x_i)$$

- What if we approximated the gradient by looking at a few random points:

$$\nabla_\theta \mathbf{L}(\theta) \approx \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\left(\sigma\left(\phi(x_i)^T\theta\right) - y_i\right)\phi(x_i)$$

Batch Size

Random sample of records

- This is a reasonable estimator for the gradient
  - Unbiased …
- Often batch size is one! (why is this helpful)
  - Fast to compute!
- A key ingredient in the recent success of deep learning

Stochastic Gradient Descent

$\theta^{(0)} \leftarrow$ initial vector (random, zeros …)

For τ from 0 to convergence:

$$\mathcal{B} \sim \text{ Random subset of indices}$$

$$\theta^{(\tau+1)} \leftarrow \theta^{(\tau)} - \rho(\tau)\left(\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\nabla_\theta \mathbf{L}_i(\theta)\Big|_{\theta=\theta^{(\tau)}}\right)$$

Decomposable Loss $\mathbf{L}(\theta) = \sum_{i=1}^{n}\mathbf{L}_i(\theta) = \sum_{i=1}^{n}\mathbf{L}(\theta, x_i, y_i)$

Loss can be written as a sum of the loss on each record.

Gradient Descent

$\theta^{(0)} \leftarrow$ initial vector (random, zeros …)

For τ from 0 to convergence:

$$\theta^{(\tau+1)} \leftarrow \theta^{(\tau)} - \rho(\tau)\left(\frac{1}{n}\sum_{i=1}^{n}\nabla_\theta \mathbf{L}_i(\theta)\Big|_{\theta=\theta^{(\tau)}}\right)$$

Stochastic Gradient Descent

$\theta^{(0)} \leftarrow$ initial vector (random, zeros …)

Very Similar Algorithms

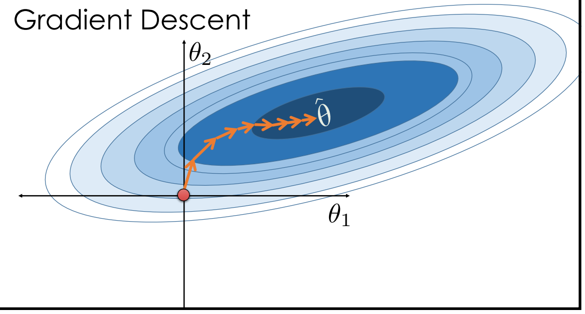
For τ from 0 to convergence:

$$\mathcal{B} \sim \text{ Random subset of indices}$$

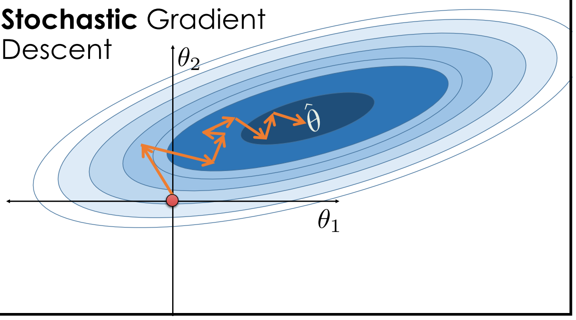
$$\theta^{(\tau+1)} \leftarrow \theta^{(\tau)} - \rho(\tau)\left(\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\nabla_\theta \mathbf{L}_i(\theta)\Big|_{\theta=\theta^{(\tau)}}\right)$$

Assuming Decomposable Loss Functions

Gradient Descent



Stochastic Gradient Descent



Basics of Random Variables

(Right after the midterm)

Characterizing Random Variables

- **Probability Mass Function** (*Discrete Distribution*)
  - The probability a variable will take on a particular value
- **Probability Density Function** (*Continuous Distributions*)
  - The probability a variable takes on a range of values.
  - Not covered … *here there be dragons*
- **Expectation**
  - The average value the variable takes (the mean)
- **Variance**
  - The spread of the variable about the mean

Summary | Expected Value and Linearity of Expectation

- Expected Value

$$\mathbf{E}[X] = \sum_{x \in \mathcal{X}} x\mathbf{P}(x)$$

- Linearity of Expectation

$$\mathbf{E}[aX + Y + b] = a\mathbf{E}[X] + \mathbf{E}[Y] + b$$

  - independence **not** required
- If X and Y are independent then $\mathbf{E}[XY] = \mathbf{E}[X]\mathbf{E}[Y]$

The Variance

$$\mathbf{Var}[X] = \mathbf{E}\left[(X - \mathbf{E}[X])^2\right] = \sum_{x \in \mathcal{X}} (x - \mathbf{E}[X])^2 \mathbf{P}(x)$$

$$= \mathbf{E}[X^2] - \mathbf{E}[X]^2$$

- Properties of Variance:

$$\mathbf{Var}[aX + b] = a^2\mathbf{Var}[X] + 0$$

  - If X and Y are independent:

$$\mathbf{Var}[X + Y] = \mathbf{Var}[X] + \mathbf{Var}[Y]$$

$$= \mathbf{E}[X^2] - \mathbf{E}[X]^2$$

- Properties of Variance:

$$\mathbf{Var}[aX + b] = a^2\mathbf{Var}[X] + 0$$

  - If X and Y are independent:

$$\mathbf{Var}[X + Y] = \mathbf{Var}[X] + \mathbf{Var}[Y]$$

- Standard Deviation (easier to interpret units)

$$\mathbf{SD}[X] = \sqrt{\mathbf{Var}[X]}$$

  - Useful identity

$$\mathbf{SD}[aX + b] = |a|\,\mathbf{SD}[X]$$

Binary Random Variable (Bernoulli)

- Takes on two values (e.g., (0,1), (heads, tails)…)

$$X \sim \mathbf{Bernoulli}(p)$$

- Characterized by probability *p*

Value	1	0
Chance	p	1-p

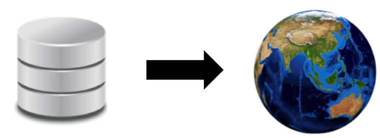
- Expected Value:

$$\mathbf{E}[X] = 1 * p + 0 * (1-p) = p$$

- Variance

$$\mathbf{Var}[X] = (1-p)^2 * p + (0-p)^2(1-p) = p(1-p)$$

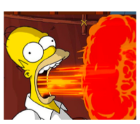
Generalization



The focus of the next few lectures.

A Simple Example

- I like to eat shishito peppers
- Usually they are not too spicy …
  - but occasionally you get unlucky (or lucky)



- Supposed we **sample *n* peppers** at random from the **population of all shishito peppers**
  - can we do this in practice?
  - Difficult! Maybe cluster sample farms?
- *What can our sample tell us about the population?*

Formalizing the Shishito Peppers

- **Population:** all shishito peppers
- **Generation Process:** simple random sample
- **Sample:** we have a sample of *n* shishito peppers
- **Random Variables:** we define a set of *n* random variables

$$X_1, X_2, \ldots X_n \sim \mathbf{Bernoulli}(p^*)$$

- Where $X_i = 1$ if the i^{th} pepper is spicy and 0 otherwise.

Population Parameter
(We don't know it.)
Remember star is for the universe.

- **Random Variables:** we define a set of *n* random variables

$$X_1, X_2, \ldots X_n \sim \mathbf{Bernoulli}(p^*)$$

- Where $X_i = 1$ if the i^{th} pepper is spicy and 0 otherwise.

Population Parameter
(We don't know it.)
Remember star is for the universe.

- **Sample Mean:** Is a random variable

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

- **Expected Value** of the sample mean:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \boxed{X_1, X_2, \ldots X_n \sim \mathbf{Bernoulli}(p^*)}$$

- **Expected Value** of the sample mean:

$$\mathbf{E}[\bar{X}] = \mathbf{E}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbf{E}[X_i]$$

Linearity of expectation

$$= \frac{1}{n}\sum_{i=1}^{n}\mu = \mu$$

Let μ be the expected value for all X_i

$$= p^*$$

For the shishito peppers setting we have $\mu = p^*$

The expected value of the **sample mean** is the **population mean!**

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \boxed{X_1, X_2, \ldots X_n \sim \mathbf{Bernoulli}(p^*)}$$

- **Expected Value** of the sample mean:

$$\mathbf{E}\left[\bar{X}\right] = \frac{1}{n}\sum_{i=1}^{n} \mu = \mu$$

- The **sample mean** is an **unbiased estimator** of the population mean

$$\textbf{Bias} = \mathbf{E}\left[\bar{X}\right] - \mu = 0$$

Sample Mean is a Random Variable

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

- Expected Value:

$$\mathbf{E}\left[\bar{X}\right] = \mathbf{E}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n}\sum_{i=1}^{n} \mu = \mu$$

- Variance:

$$\mathbf{Var}\left[\bar{X}\right] = \mathbf{Var}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right]$$

- Variance:

$$\mathbf{Var}\left[\bar{X}\right] = \mathbf{Var}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n^2}\mathbf{Var}\left[\sum_{i=1}^{n} X_i\right] \quad \text{Property of the Variance}$$

If the X_i are independent! $= \frac{1}{n^2}\sum_{i=1}^{n}\mathbf{Var}\left[X_i\right]$

- In the shishito peppers example are the X_i independent?
  - Depends on the sampling strategy
- Random with replacement (after tasking)→ Yes!
- Random **without** replacement → No!
  - Correction factor is small for large populations

- Variance:

$$\mathbf{Var}\left[\bar{X}\right] = \mathbf{Var}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n^2}\mathbf{Var}\left[\sum_{i=1}^{n} X_i\right] \quad \text{Property of the Variance}$$

If the X_i are independent! $= \frac{1}{n^2}\sum_{i=1}^{n}\mathbf{Var}\left[X_i\right]$

Define the variance of X_i as σ $= \frac{1}{n^2}\sum_{i=1}^{n}\sigma^2 = \frac{\sigma^2}{n}$

For shishito peppers with replacement $= \frac{p^*(1-p^*)}{n}$

The **variance** of the **sample mean** decreases at a **rate of one over the sample size**

Summary of Sample Mean Statistics

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

- Expected Value:

$$\mathbf{E}\left[\bar{X}\right] = \mathbf{E}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n}\sum_{i=1}^{n} \mu = \mu$$

- Variance:

$$\mathbf{Var}\left[\bar{X}\right] = \mathbf{Var}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{\sigma^2}{n} \quad \text{Assuming } X_i \text{ are independent}$$

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

- Expected Value:

$$\mathbf{E}\left[\bar{X}\right] = \mathbf{E}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n}\sum_{i=1}^{n} \mu = \mu$$

- Variance:

$$\mathbf{Var}\left[\bar{X}\right] = \mathbf{Var}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{\sigma^2}{n} \quad \text{Assuming } X_i \text{ are independent}$$

- Standard Error:

$$\mathbf{SE}\left(\bar{X}\right) = \sqrt{\mathbf{Var}\left[\bar{X}\right]} = \frac{\sigma}{\sqrt{n}}$$

← Square root law

Good Luck!