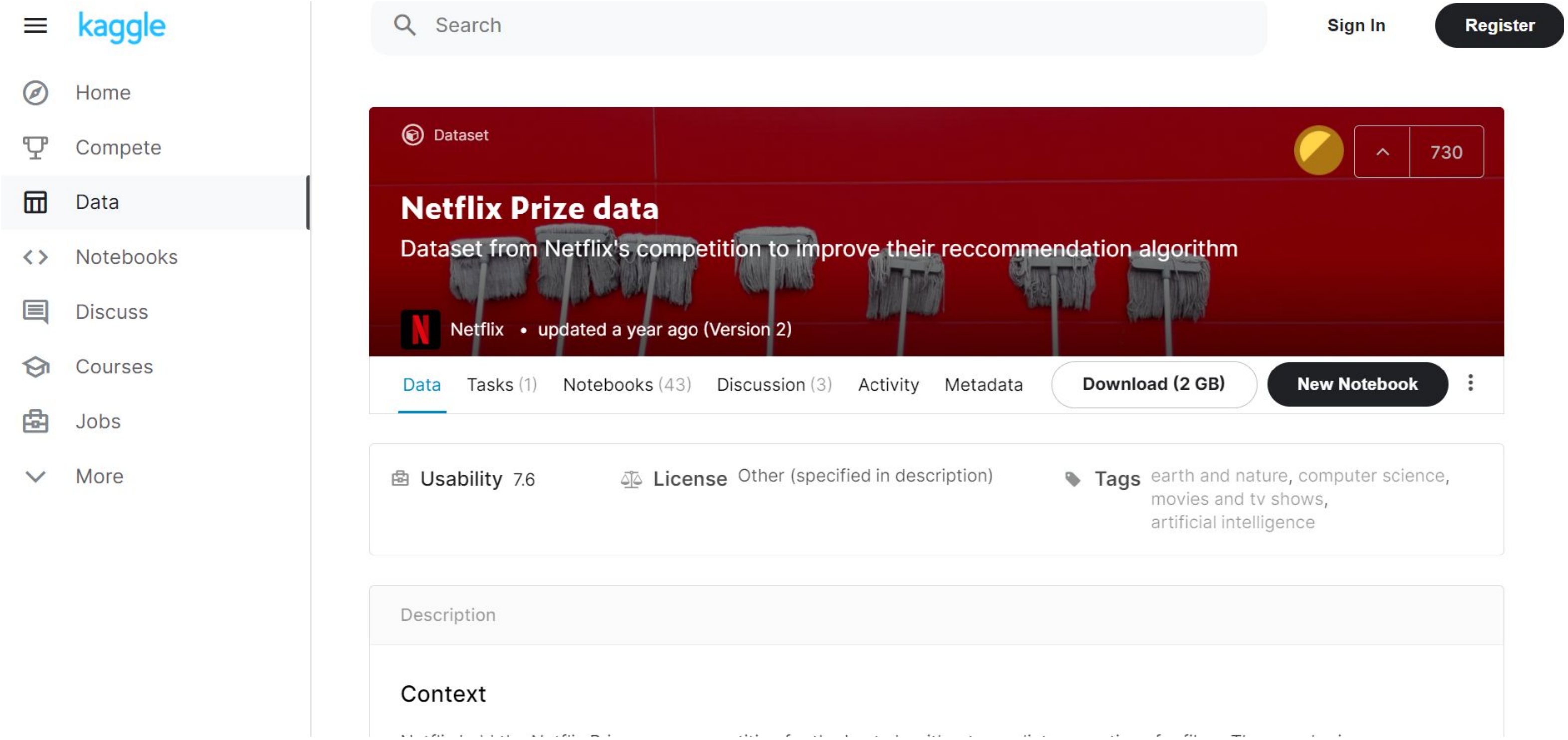
Week 6

Agenda

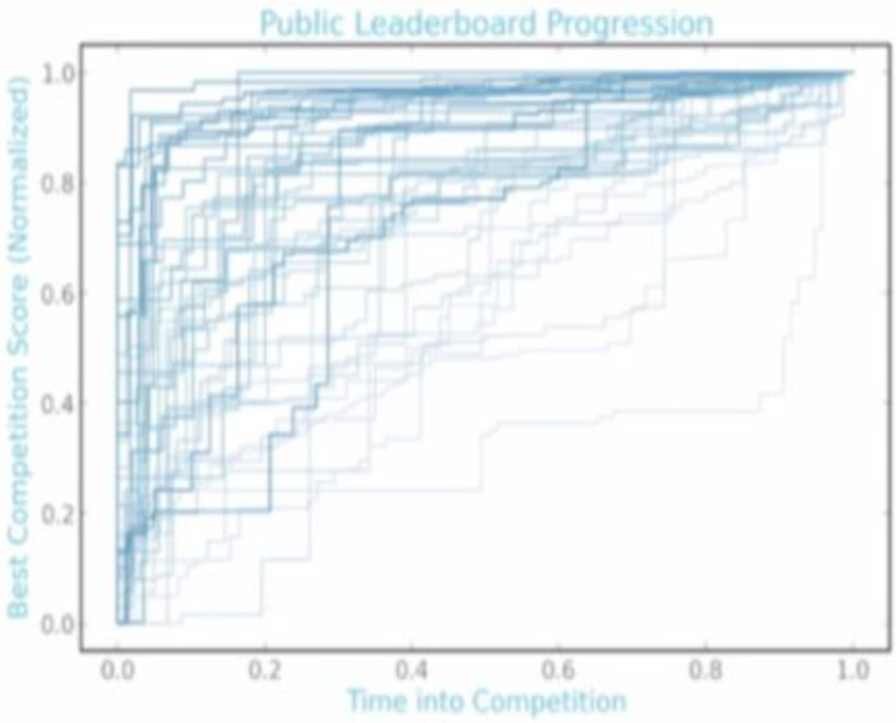
1. Project 2 intro: breakout
2. Introduction Final Project
3. Gradient Descent and Regularization review
4. Breakout: Regularization

Kaggle


kaggle
Home
Compete
Data
Notebooks
Discuss
Courses
Jobs
More
Search
Sign In
Register
Dataset
730
Netflix Prize data
Dataset from Netflix's competition to improve their reccommendation algorithm
Netflix • updated a year ago (Version 2)
Data
Tasks (1)
Notebooks (43)
Discussion (3)
Activity
Metadata
Download (2 GB)
New Notebook
Usability 7.6
License Other (specified in description)
Tags earth and nature, computer science, movies and tv shows, artificial intelligence
Description
Context

Kaggle

- ML domain independent?
- What makes Kaggle different from "real life"?



Final Project

FINAL PROJECTS

- Baseline presentation due: July 07 in class
- Final Notebook due: Sunday before last class, Sunday April 11
- Presentations: last class of the semester
- Groups of 2-4
- You pick your groups.
- Use the signup sheet shared in chat and slack

https://docs.google.com/spreadsheets/d/1roNqnPyklmxp-u9XaxSMnc-MLuR_QfqtwOma52Z42K4/edit?usp=sharing

RANDOM ACTS OF PIZZA

- https://www.kaggle.com/c/random-acts-of-pizza
- People post pizza requests on Reddit
- Build 2-class classifier
- Classify whether post will get pizza
- Practice mining features from text

HOME PRICE PREDICTION

- https://www.kaggle.com/c/house-prices-advanced-regression-techniqu
- Predict the sale price of a property (like the Zestimate)
- Regression
- Feature engineering

FOREST COVER PREDICTION

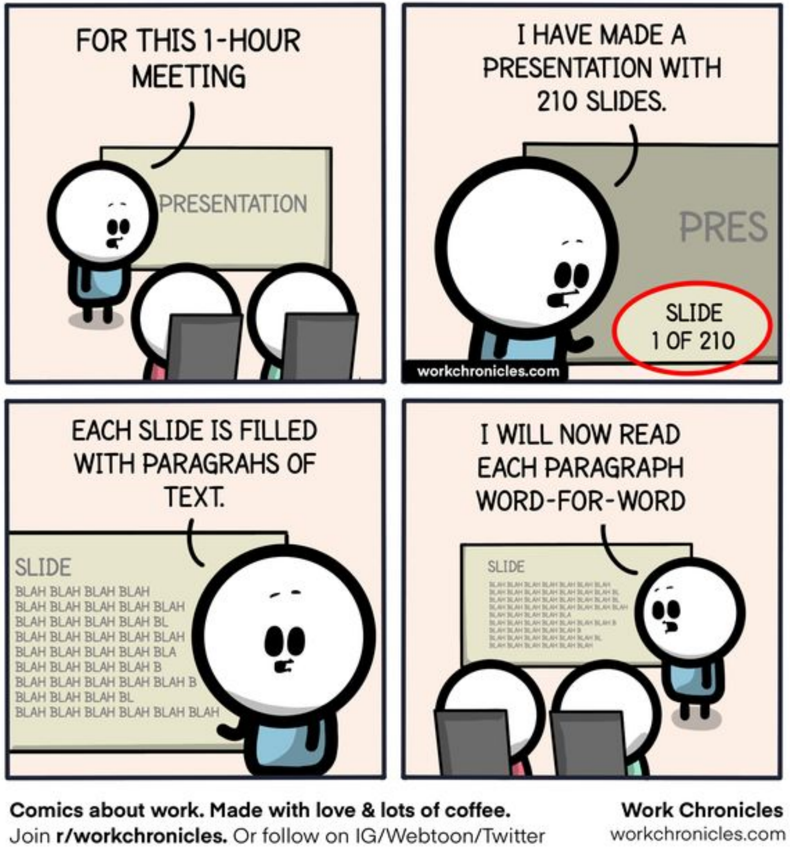
- https://www.kaggle.com/c/forest-cover-type-prediction
- Classify canopy type in forest (e.g. Spruce)
- Multi-class classification: 8 classes
- Practice trying different algorithms

FACIAL KEYPOINTS DETECTION

- https://www.kaggle.com/c/facial-keypoints-detection
- Determine x,y of keypoints in image (e.g. left eye corner)
- 30 regression outputs (x,y of 15 labels)
- Practice Convolutional Neural Networks

Baseline presentation guidelines

- make it a short **10 min presentation**
- slides or python notebooks
- introduce your group, data, problem
- first ideas on approach
- EDA
- maybe first baseline results



Gradient Descent Review

Gradient Descent

- Pseudocode:
  - Choose an initial vector of parameters α, β.
  - Choose learning rate R.
  - Repeat until an approximate minimum is obtained (randomly shuffl
    examples in training set).
  - For each example i:

$$\alpha <- \alpha - R\frac{\partial}{\partial\alpha}J(\alpha,\beta)$$

$$\beta <- \beta - R\frac{\partial}{\partial\beta}J(\alpha,\beta)$$

Algorithm:

- In short: how does it work?
- What does convergence mean?
- What is the benefit of having a convex cost function?
- Why might feature scaling be important?
- What is alpha? How might we set alpha?
- What is R? How do you initialize it?

GD Performance

- What is batch gradient descent algorithm?
- What is stochastic gradient descent algorithm (SGD)?
- What is mini-batch?

- Idea: Choose α and β so that $\alpha + \beta X_i$ is as close to Y_i for training data.
- Specifically:

$$\min_{\alpha,\beta} \sum_{i=1}^{N}(Y_i - (\alpha + \beta X_i))^2$$

Gradient Descent continued

Sigmoid Function

- Logistic (sigmoid) function: $g(z) = \frac{e^z}{e^z+1} = \frac{1}{1+e^{-z}}$
- In logistic regression: $z = \alpha + \beta X + \ldots$
- Transforms: $[-\infty, +\infty] \rightarrow [0, 1]$
- Constrains output of our model between 0 and 1

- **What is the Sigmoid function?**
- **What is the use of it?**

Gradient Descent

Examining the Cost Function

- Logistic regression cost function:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} Y_i \cdot \log \hat{Y}_i + (1 - Y_i)\log(1 - \hat{Y}_i)$$

- Can rewrite single part as two different components:

$$\mathrm{Cost}(\hat{Y}_i, Y_i) = \begin{cases} -\log(\hat{Y}_i) & \text{if } Y_i = 1 \\ -\log(1 - \hat{Y}_i) & \text{if } Y_i = 0 \end{cases}$$

Logistic Regression: Gradient Descent

- Benefit: leads to getting predicted cost values closer to actual values
  - Cost function:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} Y_i \cdot \log \hat{Y}_i + (1 - Y_i)\log(1 - \hat{Y}_i)$$

- Use the update rule:

$$\theta <- \theta - R\frac{\partial}{\partial\theta}J(\theta)$$

- Benefit: derivative is very simple:

$$\frac{\partial}{\partial\theta}J(\theta) = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)X_i$$

Bias and Variance Errors

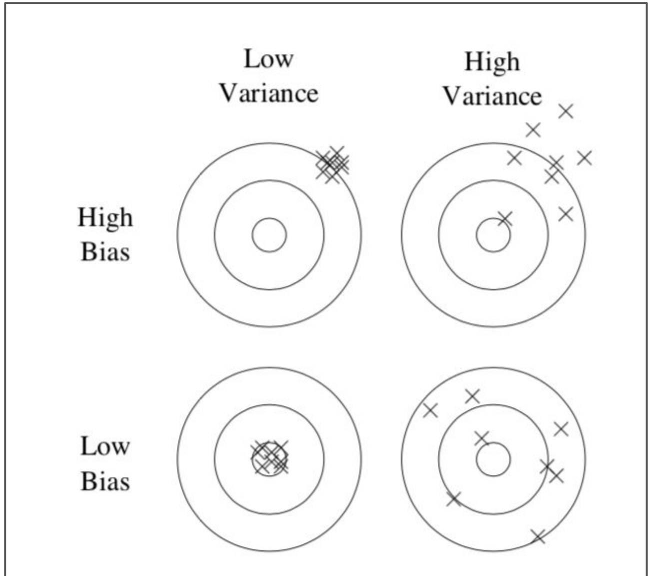
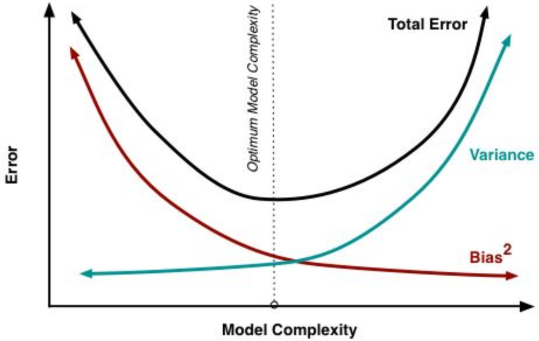


Figure 1: Bias and variance in dart-throwing.

Bias / Variance Tradeoff

1. What do bias and variance intuitively refer to?
2. Why is there typically a tradeoff?

Discuss when boosting or bagging may reduce variance and when bias



Regularization

6.3 *Weight Regularization*

In our learning objective, Eq (**??**), we had a term correspond to the zero/one loss on the training data, plus a **regularizer** whose goal was to ensure that the learned function didn't get too "crazy." (Or, more formally, to ensure that the function did not overfit.) If you replace to zero/one loss with a surrogate loss, you obtain the following objective:

$$\min_{\boldsymbol{w},b} \quad \sum_n \ell(y_n, \boldsymbol{w} \cdot \boldsymbol{x}_n + b) + \lambda R(\boldsymbol{w}, b) \tag{6.8}$$

The question is: what should $R(\boldsymbol{w}, b)$ look like?

From the discussion of surrogate loss function, we would like to ensure that R is convex. Otherwise, we will be back to the point where optimization becomes difficult. Beyond that, a common desire is that the components of the weight vector (i.e., the w_ds) should be small (close to zero). This is a form of **inductive bias**.

Breakout: Read 6.3 and discuss:

1. L1 and L2 are the most used for regularization. What are L1 and L2 Norms?
2. What effect do they have? What do they accomplish?
3. What is Elastic Net?

Regularization

Working With the Penalized Cost Function

- Penalized cost function:

$$J(\alpha, \beta) = \frac{1}{2N} \sum_{i=1}^{N} (Y_i - \theta_0 + \theta_1 X_i + \ldots + \theta_k X_i^k)^2 \overbrace{\boxed{+ \underbrace{\boxed{\lambda}}_{\text{Regularization parameter}} \sum_{j=1}^{k} \theta_j^2}}^{\text{Penalty}}$$

- The larger the θ parameter is, the higher the cost will be.

Recall the Async:
Is that L1 or L2 regularization?