INT303 Big Data Analytics

Lec 1 Intro

- Data Science Process
  1. Ask Questions
  2. Data Collection
  3. Data Exploration
  4. Data Modelling
  5. Data Analysis
  6. Visualization and Presentation of Results

Lec 2 Data

- Data
  - Gather Online Data
    - API (Application Programming Interface)
    - RSS (RIch Site Summary)
    - Web Scraping
  - Data Storage
    - Tabular Data
    - Structured Data
    - Semistructured Data
  - Data Types
    - Atomic Types: Numeric, Boolean, Strings
    - Compound: Data and time, Lists, Dictionaries
  - Data Format
    - Textual
    - Temporal
    - Geolocation
- Data Cleaning
  - Missing Data
    - Types: MCAR, MAR, NMAR
    - Handling Approaches
      - Deletion
      - **Imputation**
  - Messy Data
- Data Descriptive Statistics
  - Sampling
    - Numerical: mean, median, ...
    - Categorical: range, variance, STD, ...

Lec 3 Data Grammar

- Exploratory Data Analysis (EDA)
  1. Store Data
  2. Clean Data
  3. Explore Global Properties
  4. Explore Group Properties
- Pandas Grammar: review slides
- Data Concerns: Standardization and Normalization
  - Scale training and test sets separately

Lec 4 Data Gathering

- Web Scraping using Beautiful Soup
  1. Inspect Data Source
  2. Scrape HTML Content
  3. Parse HTML with BS
- Gathering Data from APIs

Lec 5 Data Visualization

- Types
  - Distribution: histogram, scatter plot
  - Relationship: scatter plot
  - Comparison: bar/line/pie chart, multiple histogram, boxplot
  - Composition: pie chart, stacked area graph

Lec 6 Infrastructure

- Large-scale Computing
- Storage Infrastructure: Distributed File System
  - Chunk servers
  - Master node
  - Client library for file access
- MapReduce
  - Pros
    - Easy parallel programming
    - Invisible management of hardware and software failures
    - Easy management of very-large-scale data
  - Steps
    1. Map
    2. Group by key
    3. Reduce
- Spark: extended MapReduce
  - **Resilient Distributed Dataset (RDD)**

Lec 7 Feature Engineering

- Numerical Features
  - Imputation
    - Strategies: using mean, median, mode, or a model (KNN, ...)
  - Binarization
    - 0 or 1
  - Binning
    - Fixed-width binning, Adaptative/Quantile binning
  - Log Transformation
    - Smoothing long-tailed data
  - Scaling
    - MinMax Scaling, Standard (Z) Scaling
  - Interaction Features
    - Create (nonlinear) feature combinations --> increase the complexity
- Categorical Features
  - One-Hot Encoding
    - Sparse, memory-friendly
  - Feature Hashing
    - Lower sparsely and higher compression
    - May introduce collisions
  - Bin-counting
    - Using global statistic
    - May introduce collisions
- Temporal Features
  - Time binning
  - Trendlines
- Spatial Features
  - GPS-coordinates: projection
  - Street Addressed: Geocoding
  - ZipCodes, cities, ...: Centroid coordinate
- Textual Features
  - NLP: Text Vectorization
- Feature Selection
  - Reduces model complexity and training time
  - Filter Methods: find correlation
  - Wrapper Methods: optimize the best subset (e.g., Stepwise Regression)
  - Embedded Methods: deature selection as part of training (e.g., decision trees, trees ensembles)

Lec 8 Dimension Reduction

- Principal Component Analysis (PCA)
  - Steps
    1. Standardize each predictors
    2. Calculate eigevectors
    3. Matrix Multiplication
- Singular Value Decomposition (SVD)
- PCA for Regression
  - Pros
    - Reducing dimensionality in high dimensional settings
    - Visualizing how predictive the features can be
    - Reducing multicollinearity, thus may improve computational time
  - Cons
    - Direct interpretation of coefficient in PCR is completely lost
    - Will often not improve predictive ability of a model
- PCA for Imputation
  - Iterative PCA
- **PCA for Matrix Completion (Recommender Systems)**

Lec 9-10 Model Selection and Cross Validation, Models

- Model Selection
  - Goal: choose the model that generalizes the best
  - Exhaustive Search
    - O(2^n)
  - Greedy Algorithms
    - O(n^2)
  - Fine-tuning Hyper-parameters
  - Regularization
- Cross Validation
  - Using a single validation set in model selection: may result in overfitting
- Lec 10 Models
  - Linear Regression
  - Logistic Regression
  - Decision Tree
  - KNN

Lec 11 Trees, Tree Ensembles

- Decision Tree
  - Learning approach
    1. Start with an empty tree
    2. Choose optimal predictor and threshold
    3. Recurse on each node until stopping condition
  - Pruning
    - Reduce the complexity, avoid overfitting
  - Pros
    - Easy to explain
    - Mirror human decision-making
    - Interpretable
    - Can easily handle qualitative predictors
  - Cons
    - Not that accurate
    - Non-robust
- Random samples; trees are grown independently; can caught in local optima
- Bagging
  - Bootstrap Aggregating: generate multiple samples of training data and average the predictions
  - Pros
    - High expressiveness
    - Low variance
  - Cons
    - Not longer interpretable
  - Metric: out-of-bag error
- Random Forests
  - Modified bagging -- randomly select a set of predictors (different from "averaging" in bagging), and then average the predictions
- Random samples; trees are grown independently; use random subset to decor relate the trees

Lec 12 Boosting

- Original data ONLY, no random samples; trees are grown successively using a slow learning approach
- Gradient Boosting
  - Gradient descent with MSE
  - A method for iteratively building a complex regression model T by adding simple models. Each new simple model added to the ensemble compensates for the weakness of the current ensemble
  - Steps
    1. Find a simple model, compute the residuals
    2. Fit a simple model to the residuals
    3. Update the model
    4. Compute residuals
    5. Repeat steps until the stopping condition met
- AdaBoost
  - Gradient descent with expotential loss
  - Iteratively train simple models that focuses on the points misclassified by the previous model
  - Steps
    1. Choose an initial distribution
    2. At the i_th step, fit a simple classifier on the weighted training data
    3. Update the weights
    4. Update the model
  - Issue
    - Increasing the number of trees can lead to overfitting

Lec 13 Recommedation Systems

- Content-based Recommendations
  - Main idea: Recommend items to customer x similar to previous items rated highly by x
  - Item profile and user profile
  - Pros
    - No need for data on other users
    - Able to recommend to users with unique tastes
    - Able to recommend new & unpopular items
    - Able to provide explanations
  - Cons
    - Finding the appropriate features is hard
    - Overspecialization
    - Unable to recommend for new users
- Collaborative Filtering
  - Main idea: Consider user x, find set N of other users whose ratings are similar to x's ratings
  - User-user CF
  - Item-item CF
    - Works better because items are simpler, users have multiple tastes
  - Pros
    - Works for any kind of item, no feature selection needed
  - Cons
    - Cold start, need enough users
    - The user/ratings matrix is sparse, hard to find users that have rated the same items
    - New items, esoteric items
    - Popular bias -- tends to recommend popular items
- Hybrid Methods
  - Add content-based methods to collaborative filtering
    - Item profiles for new item problem
    - Demographics to deal with new user problem
  - Implement two or more different recommenders and combine predictions -- using a linear model
- Evaluation: RMSE
  - Narrow focus on accuracy, miss prediction diversity, prediction context, order of predictions
  - Alternative: precision at top k -- percentage of predictions in the user's top k withheld ratings