JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Lecture 4

BU.330.775 Machine Learning

Minghong Xu, PhD.
Associate Professor

Reflections

- Parameters vs hyperparameters
- Gradient descent
- Learning rate/step size, adaptive learning rate
- Batch GD vs stochastic GD vs mini-batch GD
- Underfitting vs appropriate-fitting vs overfitting, regularization
- Bias and variance, generalization error
- L1 regularization vs L2 regularization
- Type I error vs Type II error, confusion matrix
- Performance measures

Gradient Descent Implementations

- https://github.com/tech-quantum/techquantum-demos/blob/master/Python/Implementation%20of%20Gradient%20Descent%20In%20Python.ipynb
- https://github.com/mattnedrich/GradientDescentExample

Today's Agenda

- Supervised machine learning models, part II
  - K-nearest neighbors
  - Boosting with tree structures
    - Adaptive boosting
    - Gradient boosting
- Competition! with breast cancer dataset

Two Task Types (Recap)

- Classification: predict a **class** label
  - Binary: two classes
  - Multiclass

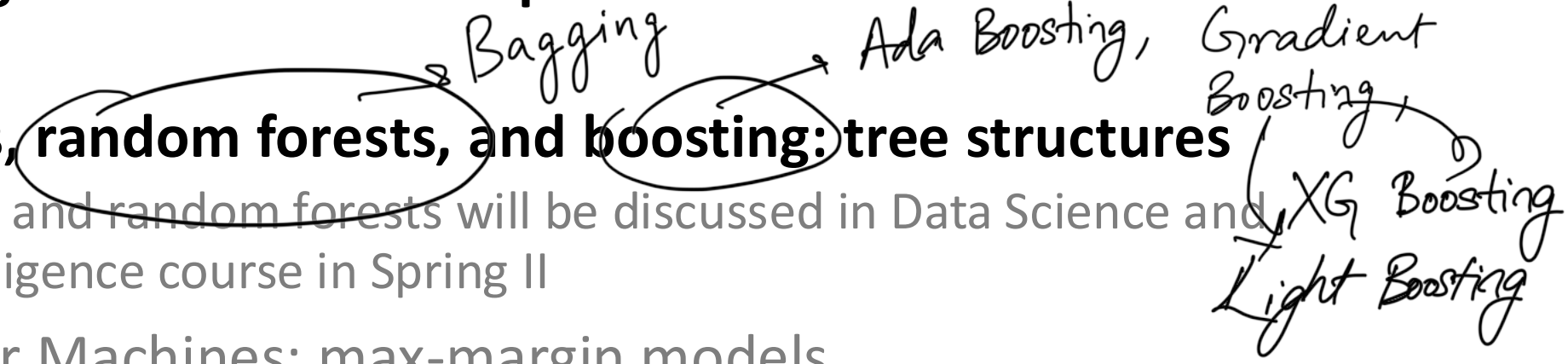
- Regression: predict a continuous number

Use Cases

- Classification
  - Fraud detection
  - Image recognition
  - Medical diagnostics
  - Customer retention
  - Personalized advertising
- Regression
  - Product sales prediction
  - Weather forecasting
  - Market forecasting
  - Population growth prediction

Major Algorithms (Recap)

- Linear regression: linear relationship
- Logistic regression: modeling the probability
- **K-nearest neighbors: similar data points tend to have similar labels or values**
- **Decision trees, random forests, and boosting: tree structures**
  - Decision trees and random forests will be discussed in Data Science and Business Intelligence course in Spring II
- Support Vector Machines: max-margin models
  - Topic of Data Science and Business Intelligence course in Spring II
- Neural networks: deep learning
  - Topic of AI Essentials for Business course in Spring I

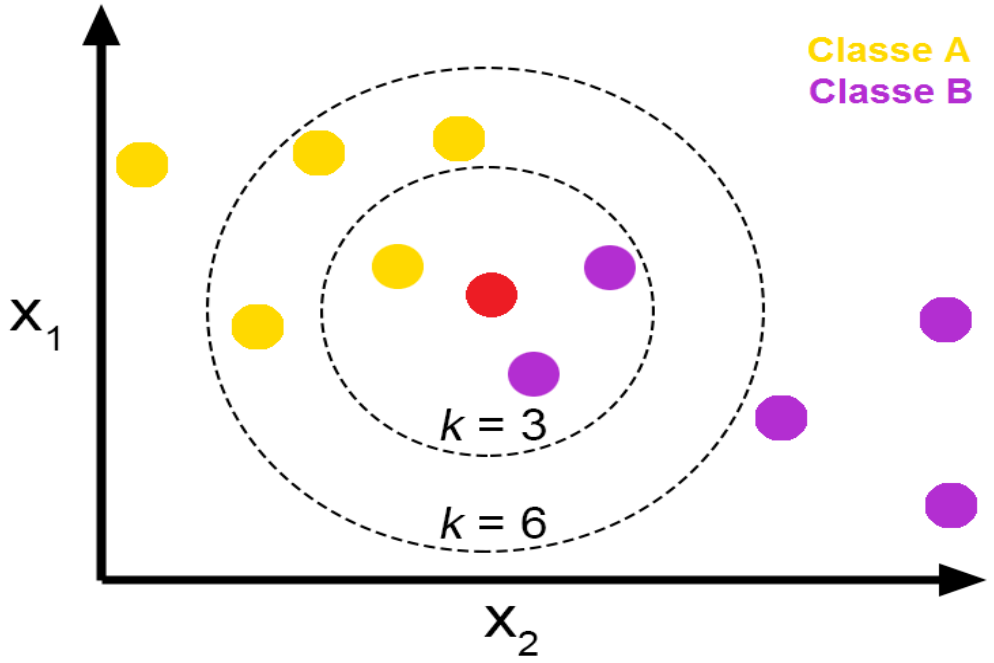


JOHNS HOPKINS
CAREY BUSINESS SCHOOL

K-Nearest Neighbors (KNN)

Intuition

- Store the training dataset in a space
- Find the “closest” data points, a.k.a, “nearest neighbors”
- Use voting to assign a label



Use Cases

- Simplest machine learning algorithm
  - No parameters to learn, thus no training needed
- Recommendation systems
  - Match customers with similar behavior
- Credit scoring
  - Evaluate the creditworthiness of a borrower by comparing their profile with those of previous borrowers

Discussion

KNN

choice of K can have huge impact of results

- Need a distance measure, such as Euclidean distance
- *Any issue?*
- Measure similarity of every pair of data points (a&b, a&c, b&c, etc.), can be computationally expensive
- Prediction can be slow when you have very large dataset, or many features
- Some speedup can be implemented
  - Pre-process and store the data structure on the training set
  - As in scikit-learn package

KNN

① distance measures like Euclidean distemo or Manhattan distance have impact on model performance

Euclidean distemo

① More sensitive if data is not scaled as it squares the distance

② have stronger impact of outlier

③ performs better dense and continuous data.

Manhattan distance.

① less sensitive to scaling as it computes just the difference.

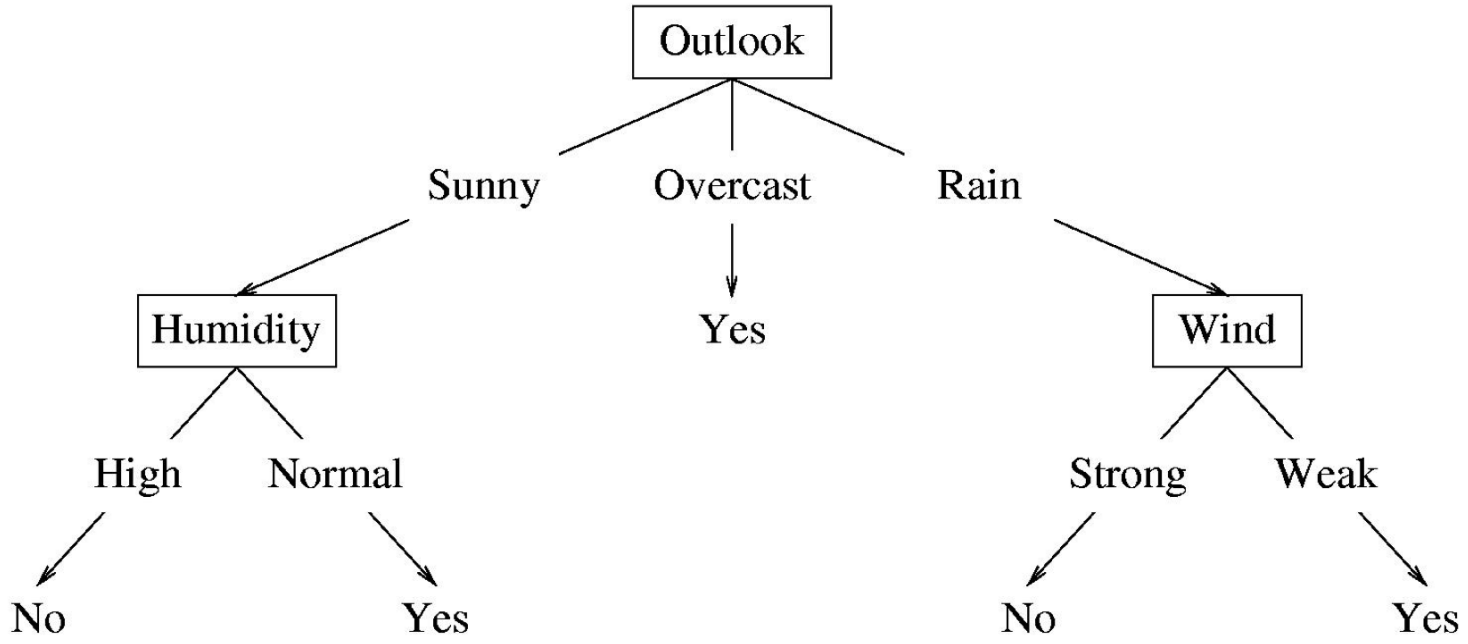
② less impacted by outliers.

③ performs better for binary and sparse data.

Boosting

Decision Trees

- A hierarchy of **if/else** questions, leading to a decision
- Each internal node represents a question
- Terminal node (leaf) contains the answer



Entropy and Tree Training

- High level training idea: search over all possible tests and find the one that is most informative about the target variable
  - Details in Data Science course

- Entropy: measure the average information content
  - Originated from thermodynamics
  - You will see cross-entropy, a common loss function used for deep learning

Strengths and Weaknesses

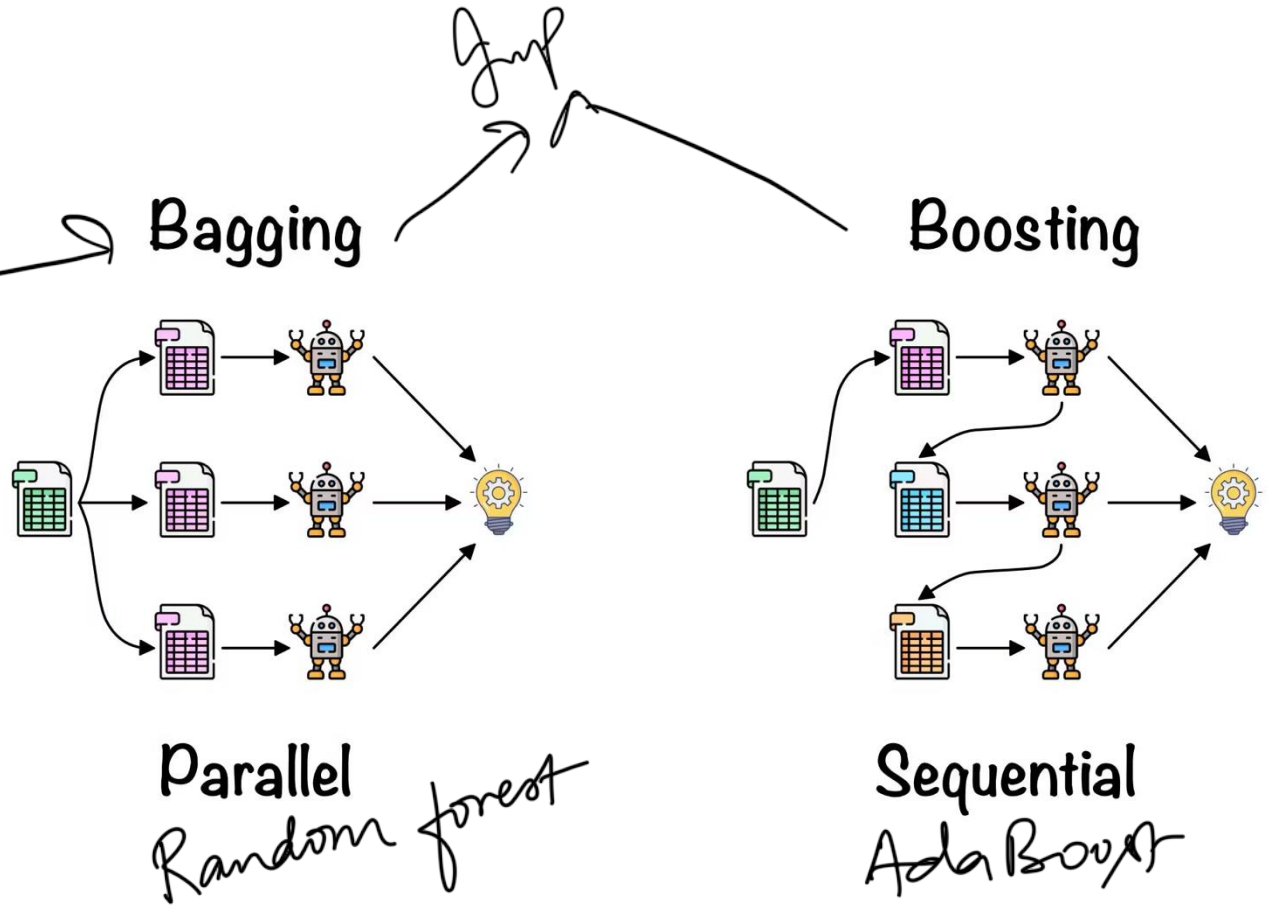
- Easily visualized and understood, white box models
- Invariant to scaling of the data
  - Each feature is processed separately
  - Possible splits of data not depend on scaling
- No preprocessing, such as normalization or standardization, needed
  - Works well when features are on completely different scales
- Poor generalization

K-neighbors and Tree Regression (Optional)

- Both KNN and tree structures can be adapted to regression tasks
- K-neighbors regression
  - Prediction: average or mean of the relevant neighbors
- Decision tree regression
  - Prediction: average value of the associated leaf

Ensemble

- Combine multiple machine learning models to create more powerful models
- Bootstrap aggregating (bagging) and random forest will be covered in Data Science course
- Boosting: train models sequentially, each on the errors of previous models



https://encord.com/blog/what-is-ensemble-learning/

XG Boost and Light Boost advanced version of Gradient Boosting,

Bagging

1. aims to reduce variance and overfitting by creating multiple subsets of training data through bootstrap sampling (sampling with replacement)
2. Final prediction is made by aggregating the prediction of all
3. Ex- Random forest

Boosting

1. Focuses on converting weaks learners into strong ones by training models sequentially Each model attempts to correct errors of its predecessors by giving more weights to misclassified instances
2. final model is weighted Sum of all individual models.
3. Ada Boost, Gradient Boost XG Boost, Light Boost.

Boosting

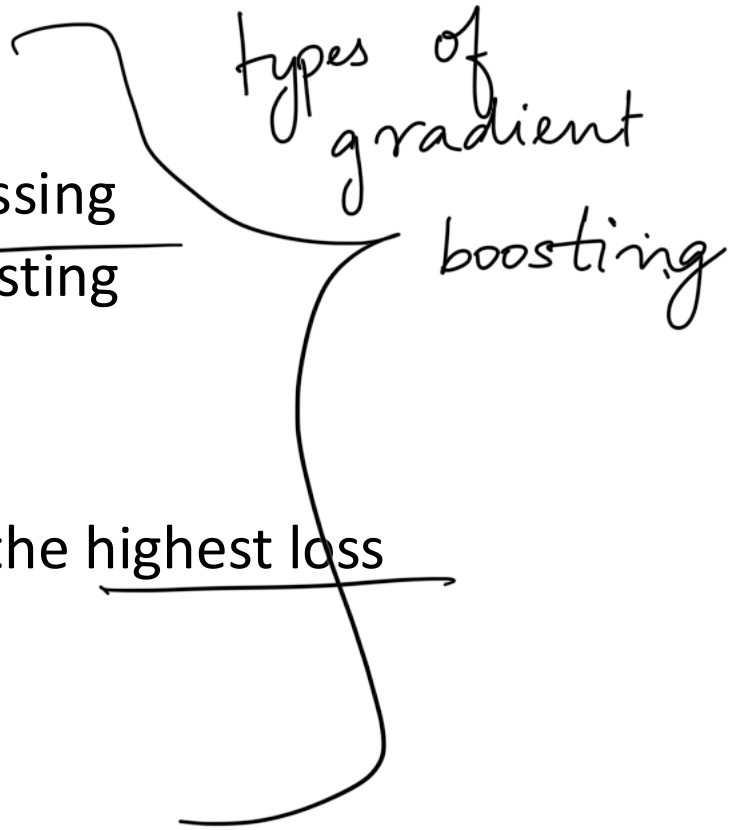
- Adaptive Boosting, or AdaBoost
  - Process: higher weights assigned to misclassified samples
  - Weighting:
    - Each model is given a weight based on its accuracy
    - Samples misclassified are weighted more heavily for the next learner
  - Final Prediction: weighted vote of all the weak learners
- Gradient Boosting → advanced implementation XGBoost & Light Boost.
  - Process: minimizes loss function by adding models sequentially
  - Each new model tries to correct the residual errors (differences between the predicted and actual values) from the previous model
  - Use gradient descent to optimize the loss function
  - Final Prediction: cumulative sum of the predictions of all models

Comparison

Aspect	Adaptive Boosting	Gradient Boosting
Error Handling	Misclassified samples are weighted more	Focuses on minimizing residual errors
Sample Weighting	Adjusts sample weights	No sample weighting
Model Weighting	Assigns weights to models based on accuracy	No model weighting
Loss Function	Penalizes misclassified samples exponentially	More flexible and allows for different loss functions depending on the task
Learning Rate	No learning rate	Has a learning rate
Use Cases	Binary classification	Both classification and regression
Interpretability	Generally simpler but less flexible	More complex and flexible
Overfitting Risk	Low risk	High risk

Implementations

- XGBoost: Extreme Gradient Boosting
  - Include regularization, tree pruning, and parallel processing
  - Faster and more scalable than traditional Gradient Boosting
  - Suitable for large datasets
- LightGBM: Light Gradient Boosting Machine
  - Leaf-wise strategy: grow tree by splitting the leaf with the highest loss reduction rather than depth-wise
- CatBoost: Categorical Boosting
  - Optimized for data with many categorical variables



Competition

- **Pre-model thinking**: Why you chose the models and why they are appropriate for the problem
- **Model explanation**: Explain your data preprocessing and modeling approach
- **After-model interpretation**: Evaluate your model's performance

- **Evaluation Criteria**: model performance (30%) and presentation quality (70%)
  - How are you convinced by the presentation
- **Evaluation Link**: https://forms.gle/hcsn5F9SfdW2q3yY7

Next Week

- Unsupervised Learning, Part I
- Dimension reduction and feature engineering