CMSC 170 Lecture 9 Notes April 17-20, 2018

Classifiers

CLASSIFIERS

- Supervised Learning - Classification
- Given *labeled data*, train model to classify unseen data into fixed set of categories / labels / classes
- Uses the *training* set for learning, the *validation set* for testing during training, and the *test set* to evaluate the final model after training

NAIVE BAYES

- Classifier based on *Bayes* Theorem: P(C | x) = [P(C) * P(x | C)] / P(x)
- *Naive* assumption: every pair of features is independent
- Usually used for text classification (e.g. spam filter)

SUPPORT VECTOR MACHINES

- A linear classifier that looks for the separating hyperplane that gives the maximal margin
- *Support vectors*: the points in the dataset closest to the separating hyperplane
- Best separating hyperplane: one where the support vectors have the maximum margin / distance to the hyperplane.
- Uses the *kernel trick* to classify non-linearly separable data
- One of the best classifiers in practice

K-NEAREST NEIGHBORS

- A classifier that doesn't perform training; instead, it checks the training set during testing
- Given a test data point, we will consult K of its nearest neighbors (according to some distance metric, e.g. Manhattan or Euclidean distance)
- The K nearest neighbors will vote on what class that test data point will belong to (majority wins)
- We usually use an odd number for K to avoid ties

DECISION TREES

- Decision rules arranged in a tree structure to predict answer; decision rules are inferred from the data
- At each level, we select the feature that best separates the remaining data
- Metrics for choosing best split: Gini impurity, information gain
- Purity of split: e.g. 4-0 (perfect split), 3-3 (50-50 split)
- Prone to overfitting

RANDOM FORESTS

- Collection of decision trees; ensemble
- Each decision tree gives prediction; majority vote wins
- Instead of using the whole dataset in building the decision tree, at each split, we use randomly sampled data (with replacement), and find the best split for them
- Individual RF trees (weak) are generally worse than a decision tree, but the combination of these weak trees are generally better than a single decision tree

NEURAL NETWORKS

- Biologically-inspired classifier that uses an interconnected group of nodes
- Inspired by network of neurons in the brain
- *Input layer*: layer that accepts the data
- *Output layer*: layer that outputs the prediction
- *Hidden layer*: all the layers in between the input and output layers; they try to learn the connections of the different features of the data
- Each *connection* between nodes in successive layers have associated **weights** which are learned during training. Nodes also have an associated **bias** factor.
- Each layer has an associated activation or non-linearity function for its nodes. The weighted sum of the values (plus bias) from the previous layer's connected nodes is passed to the non-linearity function.
- Uses backpropagation algorithm in training

ENSEMBLE

- Uses a combination of classifiers to create the final answer; uses majority voting or weighted answer
- Bagging
    - Uses randomly sampled data
    - Each classifier can be trained independently / in parallel
    - All classifiers have equal weight; majority wins
    - *Example*: random forests
- Boosting
    - Uses randomly sampled data
    - New classifiers are trained based on the mistakes of previous classifier (sequential)
    - After training a classifier, the data points in the training set that are misclassified are given higher weight so that they have a higher chance of getting selected during random sampling
    - Next training focuses its training on the misclassified data points from previous model
    - Classifiers in the ensemble have associated weight, based on their training performance
    - Weighted voting