Ensemble Learning

4/10/17
Ensemble Learning

Hypothesis Space:
• Supervised learning (data has labels)
• Classification (labels are discrete)
  • Also regression, but the algorithms differ.
• The type of mapping that can be learned depends on the base classifiers.

Key idea: Train lots of classifiers and have them vote.

Base-classifier requirements:
• Must be better than random guessing.
• Must be (relatively) uncorrelated.
A first try at ensemble learning...

We’ve learned lots of methods for classification:

• Neural networks
• Decision trees
• Naïve Bayes
• K-nearest neighbors
• Support vector machines

We could train one of each and let them vote.

Problems:

• We’d like to vote over more models.
• Some of these are quite slow to train.
• Errors may be correlated.
A better approach...

Train lots of variations on the same model, and pick a simple one, like decision trees.

Problem: re-running the decision tree algorithm on the same data set will give the same classifier.

Solutions:
1. Change the data set.
   • Bagging
2. Change the learning algorithm.
   • Boosting

Note: we’ll use decision trees in all our examples, and they’re the most popular, but the same ideas apply with other base-learners.
Bagging (Bootstrap Aggregating)

Key idea: change the data set by sampling with replacement.

- Train a strong classifier on each sample.
  - For example: a deep decision tree.
- Voting reduces over-fitting.
  - Different trees will over-fit in different ways.
Boosting

Key idea: change the algorithm by restricting its complexity and/or randomizing.

- Train lots of weak classifiers.
  - For example: shallow decision trees (stumps).
- Randomize some part of the algorithm.
  - For example: the sequence of features to split on.
- Voting increases accuracy.
  - Different stumps will make different errors.
What is this accomplishing?

Simple models often have high bias.
  • They can’t fit the data precisely.
  • They may under-fit the data.

Complex models often have high variance.
  • Small perturbations in the data can drastically change the model.
  • They may over-fit the data.

Boosting and bagging are trying to find a sweet-spot in the bias/variance tradeoff.
Ensembles and Bias/Variance

Bagging fits complex models to resamples of the data set.
- Each model will be over-fit to its sample.
- The models will have high-variance.
- Taking lots of samples and voting reduces the overall variance.

Boosting fits simple models to the whole data set.
- Each model will be under-fit to the data set.
- The models will have high bias.
- As long as the biases are uncorrelated, voting reduces the overall bias.
Ada-Boost Algorithm

Training:
assign equal weight to all data points
repeat num_classifiers times:
    train a classifier on the weighted data set
    assign a weight to the new classifier to minimize (weighted) error
    compute weighted error of the ensemble
    increase weight of misclassified points
    decrease weight of correctly classified points

Prediction:
for each classifier in the ensemble:
    predict(classifier, test_point)
return plurality label according to weighted vote
Random Forest Algorithm

Training:
repeat num_classifiers times:
  resample = bootstrap(data set)
  for max_depth iterations:
    choose a random feature
    choose the best split on that feature
    add tree to ensemble

Prediction:
for each tree in the ensemble:
  predict(tree, test_point)
return plurality vote over the predictions

Different from the reading.
Discussion: Extending to Regression

How can we extend ensemble learning to regression?

1. Suppose we had a base-learner like linear regression. How can we do boosting or bagging?

2. Suppose we used decision trees as in our earlier examples. How can we extend decision trees to do regression?

Hint: think about how we extended K-nearest neighbors to do a type of regression.