**Lecture 4 – White board notes**

# **Ensemble Learning**

* Can use both supervised and unsupervised learning techniques

Ensemble learning

* Different types of classifiers
  + Same type (Homogeneous)
  + Different type (Heterogeneous)
* Combine their predictions

Coordinated constructed ensembles

* Boosting
  + Adaboost
* Stacking

Independently constructed ensembles

* Bootstrapping sampling
* Bagging
* Random infusion
  + Random forest
  + Random subspaces
    - Rotation forest

Intuition:

Suppose we have 5 completely independent classifiers and if accuracy is 70% for each classifier, then

When all 5 classifiers are correct we get the below using binomial distribution:

n = Number of classifiers = 5, p = 0.70, x = 5

The same procedure is repeated for x=3.

## **BAGGING (Bootstrap Aggregation)**

### Algorithm

* Given a standard training set *D* of size *n*
* For i = 1 .. M
  + Draw a sample of size *n\*<n* from *D* uniformly and with replacement
  + Learn classifier *Ci*
* Final classifier is a vote of *C1* .. *CM*

### Pictorial representation

Machine Learning

F1 Learning

Bootstrapping

F2 Learning

Fk Learning

Machine Learning

Machine Learning

Bootstrapping

Bootstrapping

With the above bagging approach, case 1 will have predictions from each classifier giving F1, F2,.. Fk

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Classifiers | | | | |
| Case 1 | F11 | F21 | F31 | .. | Fk1 |
| … |  |  |  |  |  |
| New Case X | F1X | F2X | F3X | .. | FkX |

For final predictions, If F1, F2,.. Fk are discrete output: Then use majority vote

If F1, F2,.. Fk are continuous output: Average

If F1, F2,.. Fk are probabilities: Average probability

### What is the value of k?

* Can be determined using scree plot as shown below

Hence 50 classifiers would be ideal in this case

## **RANDOM FORESTS**

### Algorithm

Each tree is constructed using the following algorithm:

* 1. Let the number of training cases be *N*, and the number of variables in the classifier be *M*.
  2. We are told the number *m* of input variables to be used to determine the decision at a node of the tree; *m* should be much less than *M*.
  3. Choose a training set for this tree by choosing *n* times with replacement from all *N* available training cases (i.e. take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes.
  4. For each node of the tree, randomly choose *m* variables on which to base the decision at that node. Calculate the best split based on these *m* variables in the training set.
  5. Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).

### Pictorial representation

m – Number of features

Machine Learning

F1 Learning

Bootstrapping

Choose m

F2 Learning

Fk Learning

Machine Learning

Machine Learning

Bootstrapping

Bootstrapping

Choose m

Choose m

For final predictions, If F1, F2,.. Fk are discrete output: Then use majority vote

If F1, F2,.. Fk are continuous output: Average

## **ROTATION FORESTS**

- Rotation forests have the goal of creating accurate and diverse ensemble members

* + N = Number of cases
  + M = Number of attributes
  + K = Number of subsets
  + Each subset = M/K features

1 – Iteration

PCA

Bootstrapping

PC1, PC2.. PCM/K

F1

ML

M/K features

M/K features

PC1, PC2.. PCM/K

PCA

Bootstrapping

M/K features

M/K features

The same procedure is repeated for N Iterations.

For final predictions, If F1, F2,.. Fk are discrete output: Then use majority vote

If F1, F2,.. Fk are continuous output: Average

## **BOOSTING - Adaboost**

* Convert weak learning algorithm to a strong one

|  |  |  |
| --- | --- | --- |
|  | Iteration 1 (t = 1) | |
|  | Weights | Result |
| Case 1 | 1/m | Correctly classified |
| Case 2 | 1/m | Misclassified |
| Case 3 | 1/m | Correctly classified |
| … | … |  |
| Case m | 1/m | Misclassified |
| Classifier 1 | | |

|  |  |
| --- | --- |
|  | Iteration 2(t =2) |
|  | Weights |
| Case 1 | Less weight |
| Case 2 | More weight |
| Case 3 | Less weight |
| … |  |
| Case m | More weight |
| Classifier 2 | |

|  |  |
| --- | --- |
|  | Iteration L (t= T) |
|  | Weights |
| Case 1 | Less weight |
| Case 2 | More weight |
| Case 3 | Less weight |
| … |  |
| Case m | More weight |
| Classifier L | |

Reweight

Classifiers are weighted as