MODULE 16
Combination of Classifiers

LESSON 39
Schemes for Combining Classifiers

Keywords: Majority Vote, Weighted, Stacking

Methods for Combining Classifiers

- Once the individual classifiers is used to classify a new pattern, these decisions have to be combined in some way to obtain the final classification of the pattern.
- There are a number of ways of doing this and these are discussed below.

Majority Vote

- The class corresponding to the classification label for the new pattern chosen by most of the classifiers is chosen to be the class label of the pattern.
- One way of doing this is as follows. If there are C classes, we have a array which gives the vote of each classifier. Initially, this is initialized to 0. This means $vote(i) = 0, i = 1, .., C$. Everytime a classifier chooses a class label k for the pattern, the vote of that class is incremented. That means

  $$vote(k) = vote(k) + 1$$

  Finally, class i which has the highest value in $vote$ is chosen as the class label of the new pattern.

Weighted Majority Vote

- In this method, each classifier is given a weightage. The weightage given to all the classifiers in the majority vote was equal.
- $vote(i), i = 1, .., C$ is calculated just as in the previous method but with a weightage given to each classifier. If $wt(i)$ is the weightage given to the i_{th} classifier, then when the i_{th} class label is chosen by a classifier, then

  $$vote(i) = vote(i) + wt(i)$$

- The question to be asked here is how to determine the weightage given to each classifier. Listed below is a number of ways of determining this weightage.

    1. The performance of the classifiers on the training data is determined. The weights given to each classifier is proportional to the classification accuracy obtained by using the classifier on the training data.
    2. The Naive Bayes algorithm is used to determine the weights for each classifier by using likelihood combination.
    3. In ADABOOST as discussed earlier, the weight given to each classifier depends on the number of patterns classified correctly and the weight given to each pattern.
    4. The posterior probability is computed for each classifier which is used to determine the weights. If we have a training data D, then the likelihood $P(D \mid cl_i)$ is obtained by applying classifier cl_i to the training data D. The posterior probability is got by multiplying the prior probability of the classifier i, $P(cl_i)$ and the likelihood $P(D \mid cl_i)$.

Class Probabilities

- If we have a data point P, we have to find out the probability that P belongs to each of the classes for each classifier.
- This means that for every classifier cl_i we have to find

  $Pr(P = k \mid cl_i)$, for k=1,..,C

  which is the class probability estimate.
- For every class, the class probability estimates from the different classes are added up.
- For Class k the class probability estimate is

$$\frac{1}{n}\Sigma_{i=1}^{i=n} Pr(P = k \mid cl_i)$$

where n is the number of classifiers used.

- The class label which has the largest value of class probability estimate is the label give to P.
- Consider the case where we have 5 classifiers used on a two class data. To classify a point P we find the class probability estimate for every classifier for every class. Let the values of these probability estimates be as follows :

	Class 1	Class 2
Classifier 1	0.4	0.6
Classifier 2	0.55	0.45
Classifier 3	0.3	0.7
Classifier 4	0.6	0.4
Classifier 5	0.7	0.3

The total class probability estimate for Class 1 is $\frac{1}{5} * (0.4 + 0.55 + 0.3 + 0.6 + 0.7) = 0.51$
The total class probability estimate for Class 2 is $\frac{1}{5} * (0.6 + 0.45 + 0.7 + 0.4 + 0.3) = 0.49$

The point P will therefore be classified as belonging to Class 1.

Stacking

- In this method, a classification is done of the second level patterns which consists of class labels produced at the first level.

- Each pattern in the training set is left out of the training data and is classified by using all the other patterns by all the classifiers. These class labels obtained form second level patterns which are used to classify a new pattern P.

- To illustrate stacking, consider Figure 1.

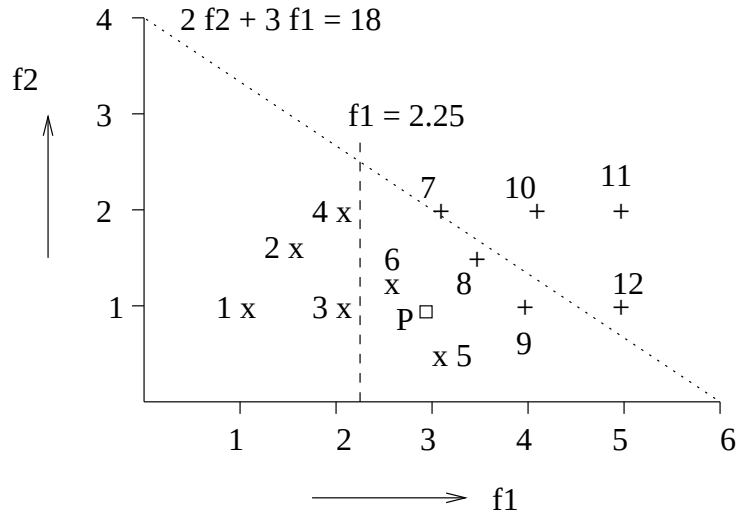

Figure 1: A two-class problem

We will consider a number of classification algorithms to classify a new pattern P which is also shown in the Figure 1.

The patterns are as follows :

Class 'x' :

Pattern 1 : (1,1) ; Pattern 2 : (1.5,1.5)
Pattern 3 : ((2,1); Pattern 4 : (2,2)
Pattern 5 : (3,1.5); Pattern 6 : (2.5,1.25)

Class '+' :

Pattern 7 : (3,2) ; Pattern 8 : (3.5,1.5)
Pattern 9 : (4,1) ; Pattern 10 : (4,2)
Pattern 11 : (5,2) ; Pattern 12 : ((5,1)

Centroid of Class 'x' = (2,1.375)
Centroid of Class '+' = (4.08,1.58)

Classifier 1:

The first classifier is that closest pattern to the pattern to be classified is found and the class label of this closest pattern will be the class of the new pattern.

The classification of the 12 patterns in the dataset if they are removed and classified using the other patterns is

Pattern :	1	2	3	4	5	6	7	8	9	10	11	12
Class label :	x	x	x	x	x	x	+	+	+	+	+	+

Classifier 2:

The second classifier is that the class label of the new pattern is the class label of the closest centroid.

The classification of the 12 patterns in the dataset if they are removed and classified using the other patterns is

Pattern :	1	2	3	4	5	6	7	8	9	10	11	12
Class label :	x	x	x	x	x	x	+	+	+	+	+	+

Classifier 3 :

The third classifier is that if the x co-ordinate of the new pattern $x \leq 2.25$ it belongs to Class 'x' and it belongs to Class '+' otherwise.

The classification of the 12 patterns in the dataset if they are removed and classified using the other patterns is

Pattern :	1	2	3	4	5	6	7	8	9	10	11	12
Class label :	x	x	x	x	+	+	+	+	+	+	+	+

Classifier 4 :

The fourth classifier is that if the co-ordinate of the new pattern falls to the left of $2y + 3x = 18$, it is classified as belonging to Class 'x' and if it falls to the right, it is classified as belonging to Class '+'.

The classification of the 12 patterns in the dataset if they are removed and classified using the other patterns is

Pattern :	1	2	3	4	5	6	7	8	9	10	11	12
Class label :	x	x	x	x	x	x	+	x	x	+	+	+

At the second level, each pattern is replaced by its class label using each of the classifiers with the actual class label as the class label again. This is shown below.

Pattern	feature 1	feature 2	feature 3	feature 4	Class
1	x	x	x	x	x
2	x	x	x	x	x
3	x	x	x	x	x
4	x	x	x	x	x
5	x	x	+	x	x
6	x	x	+	x	x
7	+	+	+	+	+
8	+	+	+	x	+
9	+	+	+	x	+
10	+	+	+	+	+
11	+	+	+	+	+
12	+	+	+	+	+

Now if we consider point P = (3,1), we classify this pattern using all the four classifiers to get the second level pattern.

The pattern representing point P is (xx+x)

Now if we find the nearest neighbour among the training patterns at the second level, we can classify P as belonging to Class 'x'.

Assignment

1. Consider the following two-dimensional training data corresponding to a two-class problem:
   $X_1 = (0.5, 1, X)^t; X_2 = ((1, 1, X)^t; X_3 = (0.5, 0.5, X)^t; X_4 = (1, 0.5, X)^t;$
   $X_5 = (2, 2.5, X)^t; X_6 = (2, 2, X)^2; X_7 = (4, 1.25, O)^t; X_8 = (5, 1.25, O)^t;$
   $X_9 = (4, 0.5, O)^t; X_{10} = (5, 0.5, O)^t;$
   If different classifiers are formed using different subsets of the training set,for the test pattern $P = (3, 2)^t$, what is the class assigned by NNC if either or both training patterns X_5 and X_6 are in the subset. What happens if majority of the subsets (classifiers) do not have both X_5 and X_6?

2. Consider the following training set :

   $X_1 = (1, 1, X)^t; X_2 = (2, 1, X)^t; X_3 = (3.3, 1, X)^t; X_4 = (1, 2, X)^t;$
   $X_5 = (2, 2, X)^t; X_6 = (5, 1, O)^t; X_7 = (6, 1, O)^t; X_8 = (5, 2, O)^t;$
   $X_9 = (6, 2, O)^t; X_{10} = (5, 3, O)^t$
   We have the disjoint subsets, S_1 = {1,2}, S_2 = {4,5}, S_3 = {3}, S_4 = {6,7}, S_5 = {8,10}, S_6 = {9}.
   Consider classifiers obtained by leaving out (a) S_1 and S_4, (b) S_1 and S_5, and (c) S_1 and S_6. What is the class label assigned to the test pattern (4,2) if NNC is used as the classifier in all the three cases?

3. Consider the data provided in problem 2. How do you learn the AdaBoost classifier using the following weak learners in that order?
   Classifier1: if $x \leq 3$ then class X, else O
   Classifier2: if $x \leq 5$ then class X, else O
   Classifier3: if $x + y \leq 3.5$ then class X, else O.
   How do you classify the test pattern (4,2) using the AdaBoost classifier?

4. Consider the dataset given in problem 2 and the test pattern $P = (4, 2)$. Classifier 1 : This method finds the centroid of the two classes. The distance from the test pattern P is found from the two centroids. Let this be $d(P, C_1)$ and $d(P, C_2)$. Then the probability that P belongs to class 1 will be

   $$Pr(P \in class1) = 1 - \frac{(d(P,C_1)}{d(P,C_1)+d(P,C_2)}$$

   Similarly,

   $$Pr(P \in class2) = 1 - \frac{(d(P,C_2)}{d(P,C_1)+d(P,C_2)}$$

   Classifier 2:

   If three closest neighbours of P is taken, it will be 3,6 and 8. Then Then $Pr(P \in class1) = \frac{1}{3} = 0.333$
   Then $Pr(P \in class2) = \frac{2}{3} = 0.667$
   Combine these probabilities to assign a label to the test pattern P.

5. Consider the dataset given in problem 2. Let h_1 be the hypothesis that if $x_1 \leq 3$, the pattern belongs to Class 'X' and the pattern belongs to Class 'O' otherwise. Let the second hypothesis h_2 be that if $x_1 \leq 5$, the pattern belongs to Class 'X' and the pattern belongs to Class 'O' otherwise. Let the prior probabilities $P(h_1)$ and $P(h_2)$ be 0.5 each. Compute the posterior probabilities.

6. Consider dataset given in problem 2. Let us consider different hypothesis for classification using this data. Let h_1 be according to the nearest neighbour. Let h_2 be according the majority vote for the closest three neighbours. Let h_3 be the classification according to the closest neighbour in the x direction. Let h_4 be the classification according to the nearest centroid. How do you use stacking based on these 4 hypotheses to classify (4,2)?

References

T.G. Dietterich(1997) Machine Learning Research : Four Current Directions, *AI Magazine*, 18(4), pages 97 - 136.

H. Drucker, C. Cortes, L.D. Jackel, Y. Lecun, V. Vapnik(1994) Boosting and other ensemble methods, *Neural Computation*, Vol. 6, No. 6, pages 1289-1301.

Yoav Freund and Robert E. Schapire(1999) A Short Introduction to Boosting, *Journal of Japanese Society for Artificial Intelligence*, 14(5), pages 771-780.

Lei Chin and Mohamed S. Kamel(2009) A generalized adaptive ensemble generation and aggregation approach for multiple classifier systems, *Pattern Recognition*, Vol. 42, Issue 5, pages 629-644.

Tin Kam Ho, Jonathan J. Hull and Sargur N. Srihari(1994) Decision Combination in Multiple Classifier Systems, *IEEE Trans on PAMI, Vol. 16, No. 1.*

Xiao Wang and Han Wang*(2006) Classification by evolutionary ensembles, Pattern Recognition, Vol. 39, Issue 4, pp. 595-607.*