MODULE 6
Different Approaches to Feature Selection

LESSON 10
Sequential Feature Selection

Keywords: Forward, Backward, Sequential, Floating

Sequential Methods

- In these methods, features are either sequentially added or deleted from feature subsets, at each step evaluating the feature subset chosen.
- These methods either start from an empty set and expand the feature subset at every iteration, or start from the set containing all features and contract the feature subset at each iteration.
- At each step, the significant feature is determined to add to the subset or to delete from the subset.
- Unlike the exhaustive search and branch and bound method, this method is not guaranteed to produce the optimal solution since all subsets are not examined.
- These methods suffer from the nesting effect.

Sequential Forward Selection(SFS)

- This method adds one feature at a time.
- At every iteration, one feature is added. The most significant feature is selected at each iteration.
- This is the **bottom-up approach** since the search starts with an empty set and keeps adding features.
- Once a feature is chosen, it cannot be deleted. This is called the **nesting effect**.
- Consider after the kth iteration, there will be k features in the subset. At this point, the significance of a feature i (which is not among the k features already selected) S_i is

  $$S_i = J(F_k \bigcup f_i) - J(F_k)$$

  where F_k is the set of k features already selected and f_i is the new feature being selected.

- At every stage, the most significant feature is selected to be added to the subset.
- This is the feature j which has the maximum value of significance.
- To illustrate this method, let us take a set of patterns with 4 features and two classes. The patterns in the training set are given below :

  Training data
  Class 1 :
  1-4,5-8,1-4,9-12 Pattern 1 : (5,7,3,12); Pattern 2 : (1,4,2,10)
  Pattern 3 : (3,8,1,8); Pattern 4 : (2,4,1,11)
  2 2.24 3 2.83 Class 2 :
  5-8, 1-4,9-12,5-8 Pattern 5 : (6,2,10,5); Pattern 6 : (4,3,8,5)
  Pattern 7 : (7,1,12,9); Pattern 8 : (8,4,11,9)

  Let us also have a validation set as follows :

  Validation data
  Class 1 :

  Pattern 1 : (3,7,3,9); Pattern 2 : (4,6,1,12)
  Pattern 3 : (2,5,3,10); Pattern 4 : (3,7,1,12)

  Class 2 :

  Pattern 5 : (7,2,12,6) ; Pattern 6 : (5,1,10,8)
  Pattern 7 : (6,3,11,6) ; Pattern 8 : (7,4,9,7)

  Now we have a feature set which is a null set. Let F be the feature set.

  Now we consider one feature at a time to find the most significant feature.

  Let us classify the training set using the validation set using the first feature only. Here 6 out of 8 patterns are classified correctly.

Let us consider only feature 2. Here also 6 out of 8 patterns are classified correctly.

Let us now consider only feature 3. Here 3 out of 8 patterns are classified correctly.

Considering feature 4 only, 5 out of 8 patterns are classified correctly.

Since feature 3 gives the maximum number of correct classifications, feature 3 is taken as the first feature selected.

We then have $F = \{3\}$

We now have to try out the feature sets $\{3, 1\}, \{3, 2\}$ and $\{3, 4\}$ and find the feature set giving the most number of correct classifications.

Then keeping these two features, the third feature is found and so on.

Sequential Backward Selection(SBS)

- This method starts with all the features as the feature subset to be considered.
- At every iteration, one feature is discarded.
- The feature to be discarded is the least significant feature.
- This is the **top-down** approach since it starts with the complete set of features and keeps discarding features at each iteration.
- Once a feature is discarded, it cannot be selected again. Hence this method also suffers from the **nesting effect**.
- At every stage, the feature j which has the minimum value of significance is selected.

- To illustrate this, consider the example given in the previous section. We start with the entire feature set. So the feature set considered is $F = \{1, 2, 3, 4\}$. From this feature set, one feature is removed at a time to get {2,3,4},{1,3,4} and {1,2,4}. Each of these feature sets is used to classify the training patterns and the feature set which misclassifies the most number of patterns is considered and the concerned feature is removed. This feature is the least significant feature. We now have three features in the feature subset. From this the least significant feature is removed to get a feature set of 2 features. If we are interested in retaining 2 out of features, we stop at this point.

Plus l-take away r Selection

- This method was introduced to take care of the **nesting effect** which has been described in the previous methods.
- In this method, at every iteration, l features are added and r features are deleted.
- The l features to be added are the most significant features like in forward selection.
- The r features deleted are the least significant features like in the backward selection.
- Choosing the values of l and r is a difficult task and may have to be done by trial and error.
- To take care of the problem of what values to use for l and r, the floating search methods were introduced.
- For example, if there are 10 features in the patterns, then if l is 3 and r is 1, then starting with the feature set $F = \phi$, first 3 features are added. For each feature added, the most significant feature is chosen. At the end of this, F will have three features. The next step is to exclude one of the features chosen. The least significant feature of the three features is removed and at this stage F has two features. Again three features are added, choosing the most significant features. Then again one feature is removed and so on. Choosing the values of l and

r arbitrary may not give the optimal feature subset. Choosing l and r to get the optimal subset is a difficult problem.

Sequential Floating Search

- In this method, there is no necessity to specify the l and r values.
- The number of forward steps and backward steps to be taken is decided dynamically so that the criterion function is maximized.
- The size of the subset of features keeps on increasing and decreasing till the search is stopped. Hence it is called the 'floating' search method.

Sequential Floating Forward Search(SFFS)

- This method starts with an empty set. To this set is added the most significant feature. At every stage, after adding a feature, the least significant feature is removed conditionally. This means that if the feature subset after removing the feature is not the best subset found so far of that size, the feature is added back again.
- The algorithm is given below.

  1. k=0
  2. Add the most significant feature to the subset and set k=k+1
  3. Conditionally remove the last significant feature. If the subset of size k-1 resulting from the removal is the best subset of size k-1, then let k=k-1 and go to Step 3 else return the feature removed and go to Step 2.

- If the total number of features d is large and the size of the subset to be found f is small, it is good to use the SFFS method since it is started from the empty set.
- As an example, consider a dataset with 10 features. First we start with $F = \phi$. The most significant feature is found. If it is 5, then $F = \{5\}$. Then according to Step 3, we need to conditionally remove 5 and check if the criterion improves. In this case, let there be no improvement by removing 5 so it is retained. Then again the most significant feature is

found. Let it be 7. Then $F = \{5, 7\}$. Let the least significant feature in 5 and 7 be 7. If there is no improvement in criterion value by removing 7, it is retained. Then the next most significant feature is found. Let it be 2. Then $F = \{5, 7, 2\}$. Now if the least significant feature in F is 7, we need to check if the criterion improves when 7 is removed. Let there be an improvement in F, then $F = \{5, 2\}$. The next most significant feature is found and added to F and so on. This process is repeated till we have the number of features we want.

Sequential Floating Backward Search

- This method starts with all the features being considered. One feature is removed at each stage and the most significant feature is added. If the subset gives the best performance of all subsets of its size, the feature added is retained otherwise it is discarded.

- The algorithm is given below :

  1. k=d
  2. Remove the least significant feature from the subset of size k. Let k=k-1.
  3. Conditionally add the most significant feature from the features not in the current subset. If the subset generated is the best subset of size k+1 found so far, retain the feature, let k=k+1 and go to Step 3. Else remove the conditionally added feature and go to Step 2.

- This method is suitable when only a few features need to be removed.

- As an example, if we take a dataset having 10 features, we start with the feature set $F = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. We now have to find the least significant feature. Let it be feature 9. F then becomes $F = \{1, 2, 3, 4, 5, 6, 7, 8, 10\}$. Then we find the most significant feature in the features absent in F. Here it is only feature 9. Then we try to see if the criterion improves if 9 is added. Here let the criterion not improve and 9 is not added again. Then from F the least significant feature is found. Let it be 4. Then 4 is removed from F giving $F =$

$\{1, 2, 3, 5, 6, 7, 8, 10\}$. Then we find the most significant feature among 4 and 9 and see if the criterion improves by including that feature. Let it not improve and we again find the least significant feature in F. Let it be feature 1. So feature 1 is removed to give $F = \{2, 3, 5, 6, 7, 8, 10\}$. If the most significant feature among 9,4 and 1 is 4 and adding 4 to F gives a better criterion function then 4 is again added to F to give $F = \{2, 3, 4, 5, 6, 7, 8, 10\}$. This process is continued to get the feature subset of the size required by us.

Max-Min approach

- This method considers two features at a time.
- At each iteration, it includes one feature into the feature subset depending on the absolute difference between the criterion of using the new feature with the feature subset and the criterion of the feature subset alone.
- Since only two features are considered at a time, it is not so computationally intensive.
- Since it does the calculations in two dimensions and not in multi- dimensional space, the results obtained are unsatisfactory.
- At a point in time, if F' is the subset of features so far selected, it is now necessary to find the feature f_j to be included.
- The difference between the criterion value of using the feature f_j along with F' and the criterion value of using only F' is found. The absolute value of this measure is called $\delta J(f_j, F')$.
- Therefore we get

  $\delta J(f_j, F') =\mid J(f_j, F') - J(F') \mid$

- The feature f_j chosen as the next feature is one which satisfies

  $max_{\forall f_j} min_{\forall F'} \delta J(f_j, F')$

- As an example, suppose we have three features f_1, f_2 and f_3. Considering f_1, if the criterion function value difference between using only f_1 and other features are

  Δ J(f_1, f_2)=20; Δ J(f_1, f_3)=25; Δ J(f_1, f_4)=30;

  Similarly, if the criterion function value difference between using only f_2 and other features are

  Δ J(f_2, f_1)=5; Δ J(f_2, f_3)=40; Δ J(f_2, f_4)=35

  If the criterion function value difference between using only f_3 and other features are

  Δ J(f_3, f_1)=60; Δ J(f_3, f_2)=10; Δ J(f_3, f_4)=25

  If the criterion function value difference between using only f_4 and other features are

  Δ J(f_4, f_1)=20; Δ J(f_4, f_2)=70; Δ J(f_4, f_3)=15

  Considering f_1 and other features, the minimum value of Δ is Δ J(f_2, f_1)=20.
  Considering f_2 and other features, the minimum value of Δ is Δ J(f_2, f_1)=5.
  Considering f_3 and other features, the minimum value of Δ is Δ J(f_3, f_2)=10.
  Considering f_4 and other features, the minimum value of Δ is Δ J(f_4, f_3)=15.
  The maximum of these four values is for Δ J(f_2, f_1). Therefore the feature chosen is f_1.

Stochastic Search Techniques

- These methods have one or more candidate solutions.
- Each candidate solution gives the feature subset selected.
- The candidate solution is a binary string where each element represents one feature.

- A 0 in the candidate solution represents the absence of a feature and a 1 represents the presence of a feature in the selected subset.
- Each string has an evaluation which depends on the performance of the subset of features on a training and validation sets. This evaluation is called the fitness function.
- In the Genetic algorithm, the initial population is formed at random.
- The next generation is formed by using the operators of selection, crossover and mutation on the population.
- A string with a higher fitness has a higher probability of being selected for the next generation.
- Crossover involves taking two strings, finding a crossover point and exchanging the elements of the two strings which occur to the right of the crossover point.
- Mutation involves choosing a bit and changing it from 0 to 1 or from 1 to 0.
- As the iterations increase, strings with higher average fitness will be formed.
- The string with the highest fitness will be the final solution.
- Some of the other stochastic search techniques are
    1. Simulated Annealing
    2. Tabu search

Assignment

1. Consider the problem of selecting 3 out of 5 features. How many feature subsets are examined by the exhaustive search?
2. Give an example feature selection problem where the branch and bound is as inefficient as the exhaustive search.

3. Consider a two-class document classification problem where the number of features is 50,000. If it is known that only 1000 features are adequate, which out of SFS, SBS do you choose. Why?

4. Consider feature selection starting with three features $f_1,\ f_2,\ f_3$. It is known that some of the feature subsets are ranked as follows:
   $f_3 > f_2 > f_1$: which means f_3 is better than f_2 which in turn is better than f_1. Similarly,
   $\{f_1,\ f_2\} > \{f_3,\ f_2\} > \{f_3,\ f_1\}$. Show how SFFS selects the best two-feature subset. What happens if we use SFS?

References

V. Susheela Devi and M. Narasimha Murty (2011) *Pattern Recognition: An Introduction*, Universities Press, Hyderabad.

A.K. Jain and D. Zongker (1997) Feature selection : evaluation, application and small sample performance, *IEEE Trans. PAMI*, Vol. 19, pp. 153-157.

L. Kuncheva and L.C. Jain (1999) Nearest neighbour classifier : simultaneous editing and feature selection, *Pattern Recognition Letters*, Vol. 20, pp. 1149-1156.

S. Lacoste-Julien, F. Sha, and M. Jordan(2009) DiscLDA: Discriminative learning for dimensionality reduction and classication, *Advances in Neural Information Processing Systems*, 21, pages 897904. **P.M. Narendra and K. Fukunaga** (1977) A branch and bound algorithm for feature subset selection, *IEEE Trans. Computers*, Vol. 26, No. 9, pp. 917-922.

P. Pudil, J. Novovicova and J. Kittler (1994) Floating search methods in feature selection, *Pattern Recognition Letters*, Vol. 15, pp. 1119-1125.

W. Siedlecki and J. Sklansky (1989) A note on genetic algorithms for large-scale feature selection, *Pattern Recognition Letters*, Vol. 10, pp. 335-347.

P. Somol, P. Pudil, J. Novovicova and P. Paclik (1999) Adaptive floating search methods in feature selection, *Pattern Recognition Letters*, Vol. 20, pp. 1157-1163.

Bin Yu and Baozong Yuan (1993) A more efficient branch and bound algorithm for feature selection, *Pattern Recognition*, Vol. 26, No. 6, pp. 883-889.