

Class Specific Feature Selection Using Simulated Annealing

V. Susheela Devi^(✉)

Department of Computer Science and Automation,
Indian Institute of Science, Bangalore 560 012, India
`susheela@csa.iisc.ernet.in`

Abstract. This paper proposes a method of identifying features which are important for each class. This entails selecting the features specifically for each class. This is carried out by using the simulated annealing technique. The algorithm is run separately for each class resulting in the feature subset for that class. A test pattern is classified by running a classifier for each class and combining the result. The 1NN classifier is the classification algorithm used. Results have been reported on eight benchmark datasets from the UCI repository. The selected features, besides giving good classification accuracy, gives an idea of the important features for each class.

1 Introduction

Whenever the training data is large or the dimensionality is large, feature selection helps to take care of the space and time complexity especially when classifiers such as the nearest neighbour are used. Besides, by reducing the number of features, the ‘curse of dimensionality’ problem is taken care of. Another benefit of feature selection would be facilitating data visualization and data understanding [6].

Feature selection is usually carried out using the wrapper or filter methods [3]. The filter method carries out feature selection based on the properties of the dataset itself. The wrapper method generates feature subsets by a method of search which are evaluated using a classification algorithm and the best feature subset found. Searching for the best feature subset cannot be done by exhaustive enumeration because of the time and space complexity involved and a number of methods are available to reduce this search space [10]. These include the branch and bound technique, the sequential forward and backward search and the min-max approach.

The search can also be carried out using soft computing techniques such as neural networks or evolutionary algorithms. Genetic algorithms [7, 8], simulated annealing [4], tabu search [13] and particle swarm optimization [9] have all been used for feature selection.

Class-specific feature selection, finds a different set of features for each class. The features which help to identify instances of one class maynot be the ones used to identify the instances belonging to another class. For example, if the

class labels are different animals, the class of elephants can be best identified by the presence of the trunk and a giraffe by the length of its neck. So, for every class, a specific feature subset may be enough to identify instances of that class. It is therefore meaningful to carry out class specific feature selection. This means that the features which are important to classify a pattern belonging to one class are different from the features which are important to classify patterns belonging to another class.

The F-score or Fisher score measures the discrimination between features [12]. Given a training pattern $x_i, i = 1, \dots, d$, the F-score of a feature f is calculated as follows :

$$F(f) = \frac{\sum_{k=1}^c (\bar{x}_f^k - \bar{x}_f)^2}{\sum_{k=1}^c \frac{1}{n_k - 1} \sum_{i=1}^{n_k} (x_{i,f}^k - \bar{x}_f^k)^2}$$

\bar{x} : Average if f^{th} feature in dataset

\bar{x}_f^k : Average of all patterns of f^{th} feature belonging to class k

$x_{i,j}^k$: i^{th} element of f^{th} feature of k^{th} class

c : Number of classes

n_k : Number of elements of k^{th} class

The numerator indicates the inter class variance, and the denominator indicates the sum of variances separately within each class. For a feature f if only those features which give large values in the numerator and small values in the denominator are used, F_f value can be increased. This is the intention of carrying out class-specific feature selection.

A few papers have worked on class-specific feature selection. In [1], a separate feature set is found for each class. A separate classifier is then built for each class based on its own feature set. To find the feature subset, the features are ranked according to a separability index such that higher the separability index, the higher the rank of the feature. Higher ranked features are combined together to form the feature subset. A hypersphere classification algorithm is used.

In [5], class-specific feature selection is used in multiclass support vector machines. Here k binary classifiers are constructed, each classifier being trained with the examples of one class with a positive label and all the other samples with a negative label. A new observation P is assigned to the class j which produces the largest distance between P and the functional margin of the classifier.

In [2], the sequential backward selection is used to select features. The evaluation of the feature subsets is done using a Naive-Bayes Classifier and a validation set.

In this paper, the simulated annealing technique has been used to find the best feature subset separately for each class. In addition, most of the papers have used small datasets with small number of classes. In this paper, some large datasets with large number of features and classes have been used. The rest of the paper is organized as follows : Sect. 1 explains the simulated annealing

technique as used in the proposed method. Section 2 details the methodology. The implementation details are given in Sect. 3. Section 4 describes the results and Sect. 5 gives the Conclusion followed by the references.

2 Methodology

In the method proposed in this paper, the attempt is to find a feature subset for each class which is the best feature subset to separate this class from all the other classes. A test pattern is classified by using k different classifiers, one for each class using only the features chosen for that class. The notations used are given in Table 1.

Table 1. Notations used

Symbol	Description
T	Temperature
X_c	Current solution
J_c	Fitness evaluation of X_c
X_n	Neighbour of current solution
J_n	Fitness evaluation of X_n
α	Cooling rate
ϵ	A very small value say 0.01
p	Probability of accepting a worse solution
Δt	Difference between J_n and J_c

2.1 Feature Subset Selection for Each Class

To find the best feature subset, a simulated annealing(SA) algorithm has been used. The simulated annealing procedure is run k times if k is the number of classes. In each run i , the feature subset for the i^{th} class is found. When carrying out the evaluation of the current string in the SA, all the patterns of the i^{th} class are taken as belonging to one class and all the other patterns belong to the other class. This process results in a set of features which classify patterns as belonging to class i or not class i .

If there are d features, the current solution of the simulated annealing has d elements where each element is either 1 or 0. If the element j is 1, it means that the feature j is present in the feature subset and if the element j is 0, it means that the feature j is not present in the feature subset. Using the current feature subset, the classification of a verification set is carried out. When the feature subset for the i^{th} class is being found, the classification is correct if one of the following conditions are satisfied:

1. If the actual class of the pattern is i and the classified class is also i .
2. If the actual class of the pattern is not i and the predicted class is also not i .

This means that if the class is i and the predicted class is not i or if the class is not i and the predicted class is i , it is a misclassification. The current solution is evaluated by finding the classification accuracy of the verification set using the feature subset selected in the current solution. The SA is used to find the best feature subset using the classification accuracy as the evaluation criterion.

The algorithm is as given below:

Feature Subset Selection Algorithm

1. X_c is the current solution. $T=100, it1=5$.
2. Set the value of the elements of X_c to 0 or 1 at random.
3. Evaluate the fitness J_c of *Validation* using *Train* using only the feature subset which has a value 1 in X_c . Also $it=0$;
4. $it = it + 1$
5. Find a neighbour of X_c .
6. Let the neighbour of X_c be X_n . Evaluate X_n to get the evaluation J_n .
7. If $J_n < J_c$, $p = \exp\left(\frac{J_c - J_n}{T}\right)$
8. if $((J_n \geq J_c) \text{ or } (random(0,1) < p))$ set $X_c = X_n$ and $J_c = J_n$.
9. if $((it \bmod it1) == 0)$ $T = T * \alpha$
10. If $(T > \epsilon)$ Go to 4.
11. The string which gave the best fitness upto this is chosen as the best solution and this string gives the feature subset chosen.

The simulated annealing algorithm used to find the right feature subset for each class is shown above. X_c is a vector which has d values if the dimensionality is d . The elements of X_c are 0 or 1 which is set at random when the algorithm is started. X_c is evaluated by classifying the patterns in *Validation* by using the patterns in *Train* but only using the features which correspond to a 1 in X_c . A neighbour of X_c is then generated. This is done by choosing a small number k of elements in X_c and changing their value to 0 if they were earlier 1 and changing their value to 1 if they were earlier 0. Here k is a user-specified value and a good value for k would be 5% to 10% of d . This is the new candidate solution X_n . X_n is then evaluated. If X_n is better than X_c , then X_n is made the current solution X_c . If X_n is worse than X_c , X_n can still be accepted as the current solution with a probability p where $p = \exp\left(\frac{J_c - J_n}{T}\right)$. This process is repeated for a number of iterations and the solution with the best evaluation is chosen as the final solution. At the beginning, the value of T is large and p is large enough so that X_n maybe accepted as the current solution even if it is a worse solution. This procedure is done to avoid falling into a local minimum. But every few generations T is reduced by the cooling rate α which is around 0.95. If the value of α is closer to 1 then it takes longer for T to reduce and the algorithm will have more number of iterations. T is kept at one value for $it1$ iterations after which it is reduced using the cooling rate. As p comes down the probability of accepting a worse solution keeps coming down. This procedure is used to prevent the algorithm from converging to a local minimum. The algorithm is stopped when T reaches a very small value say 0.01.

2.2 Classification of a Test Pattern

When a test pattern is to be classified, k classifiers are used. The classifier for class 1 uses only the feature subset selected for Class 1 and classifies the pattern as either belonging to Class 1 or not. The same procedure is applied for each of the k classes. These results are combined to find the class label of the test pattern. If the algorithm assigns two(or more) classes to the test pattern, or if the pattern is not assigned to any class, then the closest neighbour of the pattern is found using all the features and the class label of the closest neighbour is assigned to the test pattern.

3 Implementation

The algorithm has been written using GCC in linux. It has been implemented for a number of datasets from the machine learning repository [11]. Table 2 shows the datasets on which the program has been implemented. For each of the datasets the feature subset has been found for every class. It can be seen that the first three datasets have a large number of classes. Dataset 1 and 2 have 10 classes whereas dataset 3 has 26 classes corresponding to the letters of the English alphabet. The number of instances indicated in Table 2 includes the training data, the verification data and the test data. The verification data is classified using the training data to evaluate the current solution of the simulated annealing procedure. The final feature subsets chosen for each class are used to classify the test patterns and the classification accuracy has been reported.

Table 2. Description of datasets used

Sl.No	Dataset	No.of classes	No. of instances	No. of features
1	Optical digit recog	10	5620	64
2	OCR	10	10003	192
3	Letter Recognition	26	20000	16
4	Breast Cancer	2	683	9
5	Iris	3	150	4
6	Wine	3	178	13
7	Glass	6	214	9
8	Seeds	3	210	7

Table 3 shows the division of the dataset in each case into the training set, verification set and test set.

Table 3. No. of patterns in the datasets

Sl.No	Dataset	No.train	No.verific	No.test
1	Optical digit recog	2523	1300	1797
2	OCR	4450	2220	3333
3	Letter recog	8000	5000	7000
4	Breast Cancer	298	153	232
5	Iris	60	30	60
6	Wine	80	40	58
7	Glass	96	46	72
8	Seeds	90	48	72

4 Results

For each dataset, the simulated annealing procedure was used to obtain the feature subset for each class. T was initially fixed at 100. The cooling rate α was taken to be 0.95. The value of ϵ was taken to be 0.01. For every class c , the feature subset which best classified patterns of class c and classified other patterns as not belonging to class c were found.

Table 4 shows the number of features chosen among the 64 features for each of the 10 classes for the Opt.Dig.Rec data. Table 5 shows the number of features selected among the 192 features for the OCR data for each of the 10 classes. Table 6 shows the number of features selected among the 16 features for the letter recognition for each of the 26 classes. The total number of features and the number of features selected for each class for the Breast Cancer, Iris, Wine, Glass and Seeds datasets are shown in Table 7. In the Breast Cancer dataset, there are totally 9 features and two classes. In Class 1, 5 features are chosen and in Class 2, 6 features are chosen.

These reduced class-specific features were used to classify a test set in each of the datasets. The results are presented in Table 8 and Fig. 1.

It can be seen that in most of the cases, the class based feature selection is helpful in improving the classification. In data sets where the number of classes

Table 4. No. of features chosen in each class for Opt.Dig.Rec

Tot.features	C11	C12	C13	C14	C15	C16	C17	C18	C19	C110
64	32	42	41	37	35	37	36	43	34	37

Table 5. No. of features chosen in each class for OCR data

Tot.features	C11	C12	C13	C14	C15	C16	C17	C18	C19	C110
192	103	104	107	95	94	98	98	93	106	92

Table 6. No. of features chosen in each class for letter recog. data

Tot.features	Cl 1	Cl 2	Cl 3	Cl 4	Cl 5	Cl 6	Cl 7	Cl 8	Cl 9	Cl 10
16	11	10	10	10	9	14	11	11	11	11
	Cl 11	Cl 12	Cl 13	Cl 14	Cl 15	Cl 16	Cl 17	Cl 18	Cl 19	Cl 20
	10	11	10	10	12	12	14	9	11	11
	Cl 21	Cl 22	Cl 23	Cl 24	Cl 25	Cl 26				
	11	15	9	13	12	10				

Table 7. No. of features chosen in each class for some datasets

Dataset	Tot.features	Cl 1	Cl 2	Cl 3	Cl 4	Cl 5	Cl 6
Breast Cancer	9	5	6	-	-	-	-
Iris	4	2	3	3	-	-	-
Wine	13	9	7	11	-	-	-
Glass	9	5	6	2	7	6	4
Seeds	7	3	5	5	-	-	-

Table 8. Comparison of class based feature selection(CBFS) with using all features

	CBFS Acc %	All features Acc %
Opt.Dig.Rec	97.22	97.50
OCR	92.56	91.33
Letter Recog	95.04	93.32
Breast Cancer	96.55	98.71
Iris	93.33	96.67
Wine	93.10	89.66.
Glass	47.22	44.44
Seeds	80.56	86.11

are high and the features are large, there is more improvement. It is only in the case of Iris and Seeds that the classification accuracy does not improve in the case of CBFS as compared to using all features. This is likely to be due to the small number of classes or attributes.

The class based feature selection has also been compared with feature selection which is not class based. A simulated annealing procedure similar to the one used for class based feature selection was used. Only one feature subset is found. The evaluation of the current solution of the simulated annealing is carried out by finding the classification accuracy of a validation set by carrying out 1NN on the training set using only the feature subset selected. The feature subset giving the best classification accuracy is used as the final feature subset selected.

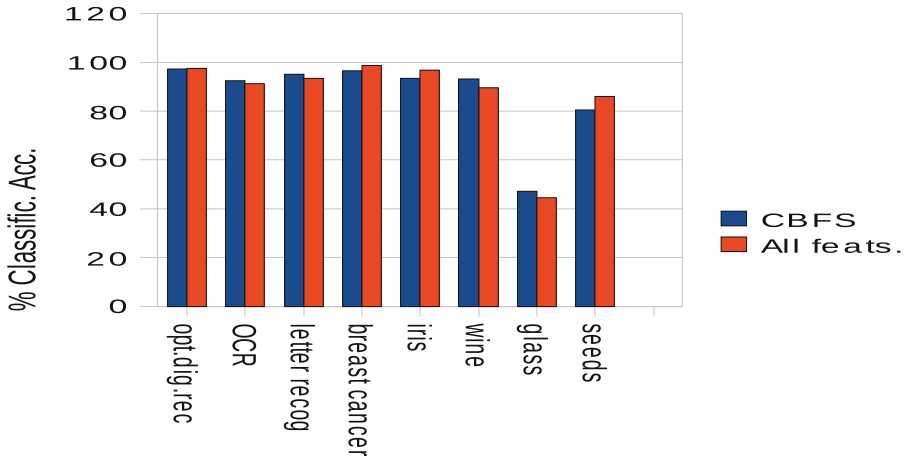


Fig. 1. Comparison of using all features Vs. class-based feature selection

Table 9 and Fig. 2 gives the classification accuracy obtained on the test dataset using feature selection and CBFS. It can be seen that for the datasets like Opt.Dig.Rec, OCR and Letter Recognition where the number of classes are 10 or more and the number of attributes is high, CBFS gives better results. Opt.Dig.Rec which has 10 classes and 64 features, shows an improvement using CBFS. In the case of OCR there are 192 features and 10 classes and here too, the improvement is evident. In Letter Recognition, there are 26 classes and 16 features and CBFS gives better results than FS. It is only in the case of datasets with only a few classes and few attributes like Breast Cancer, Wine, Glass, and Seeds that FS does better than CBFS.

Table 9. Comparison of feature selection(FS) and class based feature selection(CBFS)

	FS Acc %	CBFS Acc %
Opt.Dig.Rec	96.99	97.22
OCR	89.62	92.56
Letter Recog	94.84	95.04
Breast Cancer	98.28	96.55
Iris	93.33	93.33
Wine	94.83	93.10
Glass	48.61	47.22
Seeds	81.94	80.56

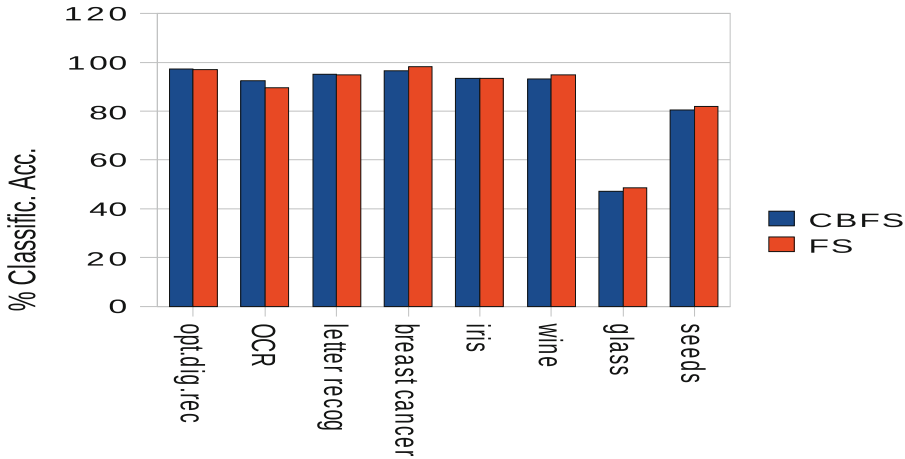


Fig. 2. Comparison of using feature selection Vs. class-based feature selection

5 Conclusion

Class-based feature selection is based on the principle that different classes have different features which are important for that class. A simulated annealing procedure is used to find the feature subset for each class so that patterns of this class are classified correctly. To classify a test pattern, a combination of k classifiers are used one for each class. While more bookkeeping is required as it is necessary to store the feature subset for each class, the feature selection is done only once and stored.

The use of class-based feature selection is found to give a good classification accuracy. This is specially true when the number of classes is large. In such a case only a few features maybe important for a particular class. In addition, the features selected for each class are an indication of which features are important for a class. It gives a description of a class. Class-based feature selection needs to be tried out with more data sets in the future, especially datasets where the number of features and the number of classes are large. It is also possible to use other classifiers besides the nearest neighbour bases approaches to carry out class specific feature selection. This is to be investigated.

References

1. Mackin, P.D., Roy, A., Mukhopadhyay, S.: Methods for pattern selection, class-specific feature selection and classification for automated learning. *Neural Netw.* (2013). doi:[10.1016/j.neunet.2012.12.007](https://doi.org/10.1016/j.neunet.2012.12.007)
2. Gilbert, J.E., Soares, C., Williams, P., Dozier, G.: A class-specific ensemble feature selection approach for classification problems. In: *ACMSE 2010* (2010)
3. Dash, M., Liu, H.: Feature selection for classification. *Intell. Data Anal.* **1**, 131–156 (1997)

4. Debuse, J.C.W., Rayward-Smith, V.J.: Feature subset selection within a simulated annealing data mining algorithm. *J. Intell. Inf. Syst.* **9**, 57–81 (1997)
5. Francois, D., de Lannoy, G., Verleysen, M.: Class-specific feature selection for one-against-all multiclass svms. In: *ESANN 2011 Proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 263–268 (2011)
6. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3**, 1157–1182 (2003)
7. Oh, J.-S.L.I.-S., Moon, B.-R.: Hybrid genetic algorithms for feature selection. *IEEE Trans. PAMI* **26**(11), 1424–1437 (2004)
8. Lanzi, P.L.: Fast feature selection with genetic algorithms: a filter approach. In: *IEEE International Conference on Evolutionary Computation*, pp. 537–540 (1997)
9. Lie, Y., Wang, G., Chen, H., Dong, H., Zhu, X., Wang, S.: An improved particle swarm optimization for feature selection. *J. Bionic Eng.* **8**, 191–200 (2011)
10. Murty, M.N., Devi, V.S.: *Pattern Recognition : An Algorithmic Approach. Undergraduate Topics in Computer Science*. Springer, London (2011)
11. UCI Repository of Machine Learning Databases (1998). <http://www.ics.uci.edu/mllearn/MLRepository.html>
12. Chen, Y.W., Lin, C.J.: Combining svms with various feature selection strategies. *Strat.* **324**(1), 1–10 (2006)
13. Zhang, H., Sun, G.: Feature selection using tabu search method. *Pattern Recog.* **35**, 701–711 (2002)

Mining Intelligence and Knowledge Exploration
Third International Conference, MIKE 2015, Hyderabad,
India, December 9-11, 2015, Proceedings
Prasath, R.; Vuppala, A.K.; Kathirvalavakumar, T. (Eds.)
2015, XVIII, 713 p. 216 illus. in color., Softcover
ISBN: 978-3-319-26831-6