Application of the Bees Algorithm to the Selection Features for Manufacturing Data

D.T. Pham, M. Mahmuddin, S. Otri, and H. Al-Jabbouli

Intelligent Systems Laboratory, Manufacturing Engineering Centre, Cardiff University, Cardiff, CF24 3AA, UK

Abstract

Data with a large number of features tend to be deficient in accuracy and precision. Some of the features may contain irrelevant information caused by data redundancy or by noise. A "wrapper" feature selection method using the Bees Algorithm and Multilayer Perceptron (MLP) networks is described in this paper. The Bees Algorithm is employed to select an optimal set of features for a particular pattern classification task. Each "bee" represents a possible set of features. The MLP classification error is computed for a data set with those features. This information is supplied to the Bees Algorithm to enable it to select the combination of features producing the lowest classification error. The proposed method has been tested on data collected in semiconductor manufacturing. The results presented in the paper clearly demonstrate the effectiveness of the method.

Keywords: Bees Algorithm, Feature selection, MLP network.

1. Introduction

The volume of data generated in manufacturing has grown dramatically in recent years. An example is the large amounts of data produced in connection with various fabrication processes [1]. Often, the data contain patterns that are useful to management for decision making. Unfortunately, it is very expensive to process such data due to the large number of features present in the data, some of which might be irrelevant, redundant or otherwise useless. A possible approach is to employ only a subset of the available features. Deciding which features to include in this subset is known as feature selection. This paper describes the use of the Bees Algorithm to perform feature selection.

The organisation of this paper is as follows. Section 2 discusses feature selection in general and outlines the Bees Algorithm which is an optimisation tool developed at the authors' Centre. Section 3 explains the proposed Bees-Algorithm-based feature selection procedure. Section 4 presents test results for the proposed feature selection procedure. Section 5 concludes the paper.

2. Feature Selection and the Bees Algorithm

2.1 Feature Selection

Feature selection facilitates data processing by reducing the dimensions of the data and by removing features that are not useful for data classification. The feature selection method chosen in this work belongs to the "wrapper" category [2]. In other words, feature selection is performed at the same time as the evaluation of how accurately data are classified with different feature subsets.

Different feature selection techniques have been investigated. For example, Zhou and Jin [3] used principal component analysis to reduce the dimensions of forging tonnage signals. Pfingsten et al. [4] investigated combining feature selection and Support Vector Machines (SVM) to simplify and classify data collected in large-scale production of semiconductor devices. Campoccia et al. [5] used a Genetic Algorithm (GA) based feature selection method to minimise the number of parameters in an electrical distribution fault identification problem.

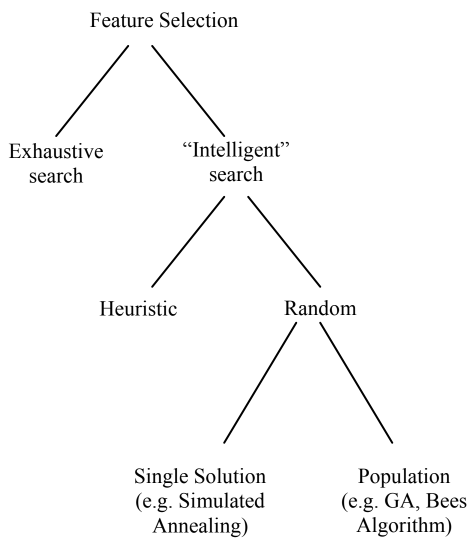

Fig. 1. Feature selection methods

A number of feature selection algorithms have been proposed. They can be grouped according to whether they perform exhaustive search or "intelligent" search (see Figure 1). The exhaustive search strategy involves examining all points in the solution space and discovering all possible solutions. This brute- force strategy is very time consuming especially for large data sets.

"Intelligent" search algorithms use either heuristic or random search techniques. Heuristic search employs rules of thumbs to guide the search process. An example of a heuristic search strategy is given in [6]. A selection algorithm implementing random search randomly explores the feature space and randomly forms feature subsets. This approach tends to be employed in conjunction with some classifier, such as an MLP neural network [7], an SVM [4], or a Radial Basis Function network [8] as a training or evaluation tool to guide the search for the optimal subset of features.

"Intelligent" search algorithms adopting random search techniques include the GA [5], Ant Colony Optimisation [6], Particle Swarm Optimisation [9], Tabu Search [10] and Simulated Annealing [11]. Some of these algorithms (for example, the GA) simultaneously explore different points of the search space using a population of candidate solutions. Others (for example, Simulated Annealing) examine the search space a point at a time.

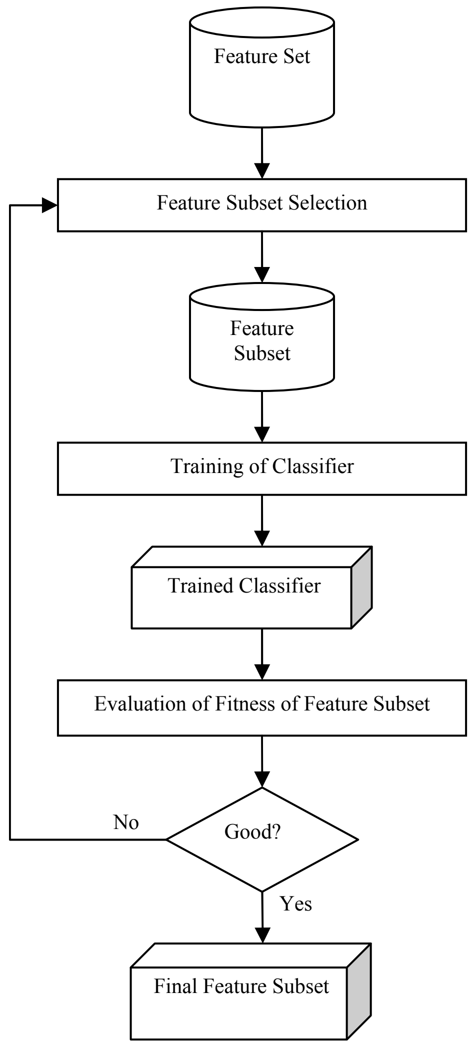

Fig. 2 Wrapper Feature Selection (adapted from [12])

Feature selection for pattern classification can take one of two approaches. If the selection process is independent of the classification process, the approach is called "filter" selection. The filtering operation usually employs a measure such as "information gain" to guide a filter selection procedure. When the selection of features relies on feedback from a classifier, the approach is known as "wrapper selection" [12] (see Figure 2). As mentioned previously, wrapper selection was adopted in this work as it can yield feature sets with more relevant and informative features when the data sets have very large numbers of features [13].

```
Initialise population and evaluate
fitnesses.
Do
  Select elite bees for neighbourhood
      search.
  Select other bees for neighbourhood
      search.
  Recruit bees around selected bees and
      evaluate fitnesses.
  Select fittest bee from each site.
  Assign remaining bees to search
      randomly and evaluate their
      fitnesses.
While stopping criterion not met.
```

Fig. 3. Main steps of Bees Algorithm

2.2 Bees Algorithm

The Bees Algorithm [14] is a new population-based optimisation algorithm inspired by the foraging behaviour of bees. Figure 3 gives a simplified description of the main steps of the algorithm. Since its development, the algorithm has found a variety of applications including function optimisation [14], training of neural networks for pattern recognition [15], the formation of homogeneous data clusters [16] and the generation of multiple feasible solutions to a preliminary design problem [17]. For further details of the Bees Algorithm, see [14].

3. Proposed Feature Selection Method

As mentioned previously, the proposed method employs MLP networks as classifiers to guide the selection of features. This guidance is provided in the form of feedback to the selection process as to how well a given set of features characterises patterns from different classes.

The method also requires a data set for use in the feature selection process. The data set comprises patterns, each with N_t features. The classes of all the patterns in the training set are known. From the original data sets, new data sets can be constructed in which patterns only have a subset of the original features. In other words, a pattern in a new data set will have N_s features $(1 \le N_s \le N_t)$ selected from the original set of N_t features.

A bee represents a subset of N_s features. It can be uniquely identified by a binary string (e.g. 010110111) where the total number of bits is N_t and the total number of non-zero bits is N_s. The position $i(1 \le i \le N_t)$ of a bit along the string indicates a particular feature. If a feature is selected to form a data set, the corresponding bit is 1. Otherwise, it is zero.

Feature selection starts with the random generation of a population of binary strings (or bees). For each string, a data set is constructed using the selected features specified in the string. Part of the data sets (training data) is taken to train an MLP. The remaining data (the test data) is employed to evaluate the classification accuracy of the trained MLP.

The classification accuracy obtained for a particular data set and the number of features in the data set give the fitness of the corresponding bee, as follows:

$$Fitness = \frac{1}{k_1 \times MSE + k_2 \times \frac{N_S}{N_t}} \quad (1)$$

where k_1 and k_2 are weighting factors and MSE is the mean squared error of the MLP in classifying the test data. The term $\frac{N_S}{N_t}$ is included in Eq (1) to reflect the fact that it is more desirable to have small feature subsets. k_1 and k_2 enable the scaling of the contribution of MSE and $\frac{N_S}{N_t}$ to suit particular problems.

As can be seen in Figure 3, the Bees Algorithm involves neighbourhood searching. In this work, this means generating and evaluating neighbours of the fittest bees. Various operators could be employed to create neighbours of a given bee, including monadic operators such as mutation, inversion, swap and insertion (single or multiple). For the test problems considered in the next section, the 2-opt and 3-opt operators [18] were adopted.

4. Selection of Features for Semiconductor Data

4.1 Data Sets

The data sets described in [19] were employed in this work to test the proposed feature selection method. The original data making up the data sets had been obtained in the fabrication of semiconductor devices at Motorola [19]. The original data set comprised 16381 instances, 14074 of them representing good products and 2307 sub-standard or faulty products. Each instance had 131 features, of which 59 (the "C" features) were measurements of different attributes of the product (i.e. these features characterise its appearance and feel), 39 (the "K" features) pertained to the outcomes of tests on the product and 33 (the "X" features) gave the processs parameters adopted for the product (i.e. these features describe how it was made). The X features were a mixture of text and numbers and could not directly be handled by MLPs. Therefore, they were not used in this work.

The original data set was projected in two ways to produce two different sets, the "C" set and the "K" set comprising instances with only the "C" features and the "K" features, respectively. The projected data sets had the same total numbers of instances and numbers of "good" and "bad" instances as the original data set. Table 1 summarises the main properties of these data sets.

4.2 Experiments

The proposed method was used to select the best collections of features for the "C" and "K" data sets. In all experiments, 80% of the data were employed for training MLP networks and 20 % for testing the trained networks. The proportion of "good" and "bad" instances in the projected data sets and in the training and test data sets were kept the same as in the original data set so that the MLP networks were exposed to data of approximately the same nature as in the case when all the features were used. The empirically chosen parameters of the MLP networks are given in Table 2, which also shows the parameters adopted for the Bees Algorithm. The values used for k_1 and k_2 in Eq (1) were both equal to 1.

4.3 Results

The proposed feature selection method was applied three times to derive the most useful features for both the C and K data sets. The results obtained are presented in Table 3. For comparison, the results for the full-feature data sets (i.e. the data sets with all the features) are also shown in the table.

From the results obtained, it can be noted that classification accuracies comparable with those for the full-feature cases were achieved despite large reductions in the number of features. This confirms the ability of the proposed method to choose informative features, thus simplifying the subsequent task of training reliable pattern classifiers.

Table 1
Details of data sets

	Original	**"C"**	**"K"**
No. total instances	16381	16381	16381
No. good instances	14074	14074	14074
No. bad instances	2307	2307	2307
No. features	131	59	39

Table 2
Parameters used for the MLP networks and the Bees Algorithm

	Parameters	**Value**
MLP	Learning rate	0.3
	Momentum	0.1
	Threshold	0.001
	Number of epochs	10000
Bees Algorithm	Number of scout bees	25
	Number of sites selected for neighbourhood search	5
	Number of elite bees	2
	Number of bees recruited for the elite sites	20
	Number of bees recruited for the other selected sites	15
	Number of iterations	200

Table 3
MLP classification accuracies using reduced-feature data sets and full-feature data sets

Data set	Run	Feature selected	Accuracy
"C"	1	38/59	84.99
	2	29/59	84.99
	3	28/59	84.96
	-	59/59	84.99
"K"	1	22/39	96.49
	2	23/39	97.01
	3	24/39	97.62
	-	39/39	98.38

5. Conclusion

Classifying data with large numbers of features can be a difficult process. Training classifiers for such data is a time-consuming task that does not produce consistent results. This paper has presented a method of reducing the number of features by selecting only those most relevant to separating the different groups in a data collection. The proposed method employs the Bees Algorithm, an optimisation technique inspired by the foraging behaviour of honey bees. The new method has been applied to real data sets, yielding substantial reductions in the numbers of features without unduly affecting classification accuracies.

In common with "wrapper" feature selection methods, the proposed method incorporates the training and testing of classifiers as part of the feature selection process. This makes the whole procedure very lengthy. Future work aimed at producing faster versions of the Bees Algorithm should alleviate this problem.

Acknowledgements

The authors would like to thank Kari Torkkola (Motorola Labs), for the use of the Motorola data set. This work was carried out within the ERDF (Objective One) project "Supporting Innovative Product Engineering and Responsive Manufacturing" (SUPERMAN), the EC-funded Network of Excellence on "Innovative Production Machines and Systems" (I*PROMS) and the EPSRC-funded Innovative Manufacturing Research Centre at Cardiff University.

References

[1] Russell EL. *Massive Data Sets in Semiconductor Manufacturing*, in *Workshop on Massive Data Sets*. 1997, National Academy of Sciences: Washington D.C. pp. 69-76.

[2] Kohavi R and John GH. *Wrappers for Feature Subset Selection.* Artificial Intelligence, 1997. **97**(1-2): pp. 273-324.

[3] Zhou S and Jin JJ. *Automatic Feature Selection for Unsupervised Clustering of Cycle-based Signals in Manufacturing Processes.* IIE Transactions, 2005. **37**(6): pp. 569-584.

[4] Pfingsten T, Herrmann D, Schnitzler T, Feustel A, and Schölkopf B. *Feature Selection for Trouble Shooting in Complex Assembly Lines.* IEEE Transactions on Automation Science and Engineering, 2007. **4**(3): pp. 465-469.

[5] Campoccia A, Dusonchet L, Augugliaro A, Sanseverino ER, and Di Silvestri ML. *GA-based Feature Selection for Faults Identification in Electrical Distribution Systems*. *Proc International Conference on Electric Power Engineering, 1999. PowerTech Budapest 99*. 1999. Budapest, Hungary, pp. 186.

[6] Al-Ani A. *Feature Subset Selection Using Ant Colony Optimization.* International Journal of Computational Intelligence, 2005. **2**(1): pp. 53-58.

[7] Verikas A and Bacauskiene M. *Feature Selection with Neural Networks.* Pattern Recognition Letters, 2002. **23**(11): pp. 1323-1335.

[8] Jia P and Sang N. *Feature Selection Using a Radial Basis Function Neural Network based on Fuzzy Set Theoretic Measure*. *Proc Third International Symposium on Multispectral Image Processing and Pattern Recognition (MIPPR'03)*. 2003. Beijing, China: SPIE, pp. 109-114.

[9] Liu Y, Qin Z, Xu Z, and He X. *Feature Selection with Particle Swarms*, in *Computational and Information Science*. 2004, Springer Berlin/Heidelberg. pp. 425-430.

[10] Zhang H and Sun G. *Feature Selection Using Tabu Search Method.* Pattern Recognition, 2002. **35**(3): pp. 701-711.

[11] Debuse JCW and Rayward-Smith VJ. *Feature Subset Selection within a Simulated Annealing Data Mining Algorithm.* Journal of Intelligent Information Systems, 1997. **9**(1): pp. 57-81.

[12] Kalapanidas E and Avouris N. *Feature Selection for Air Quality Forecasting: a Genetic Algorithm*

Approach. AI Communications, 2003. **16**(4): pp. 235-251.

[13] Bins J and Draper BA. *Feature Selection from Huge Feature Sets*. *Proc Eighth IEEE International Conference on Computer Vision, 2001 (ICCV 2001)*. 2001. Vancouver, BC, Canada, pp. 159-165.

[14] Pham DT, Ghanbarzadeh A, Koc E, Otri S, and Zaidi M. *The Bees Algorithm – A Novel Tool for Complex Optimisation Problems*. *Proc 2nd Virtual International Conference on Intelligent Production Machines and Systems (IPROMS 2006)*. 2006. Cardiff, UK: Elsevier, pp. 454-459.

[15] Pham DT, Otri S, Ghanbarzadeh A, and Koc E. *Application of the Bees Algorithm to the Training of Learning Vector Quantisation Networks for Control Chart Pattern Recognition*. *Proc 2nd IEEE Information and Communication Technologies*. 2006. Damascus, Syria, pp. 1624-1629.

[16] Pham DT, Otri S, Afify A, Mahmuddin M, and Al-Jabbouli H. *Data Clustering Using the Bees Algorithm*, in *40th CIRP International Seminar on Manufacturing Systems*. 2007: Liverpool.

[17] Pham DT, Castellani M, and Ghanbarzadeh A. *Preliminary Design Using the Bees Algorithm*. *Proc 8th International Conference on Laser Metrology, CMM and Machine Tool Performance, LAMDAMAP*. 2007. Cardiff, UK: Euspen, pp. 420-429.

[18] Croes GA. *A Method for Solving Traveling-Salesman Problems.* Operations Research, 1958. **6**(6): pp. 791-812.

[19] Torkkola K and Tuv E. *Visualization of Massive Mixed Type Semiconductor Manufacturing Data Using Self Organizing Maps*, in *Engineering Application of Neural Networks Conference*. 2003: Malaga, Spain.