Filter-Based Feature Selection Using Two Criterion Functions and Evolutionary Fuzzification

Ohm Sornil(✉)

Graduate School of Applied Statistics,
National Institute of Development Administration, Bangkok, Thailand
osornil@as.nida.ac.th

Abstract. Real world problems often contain noise features which can decrease effectiveness of classification models. This article proposes a filter-based technique to select a minimal set of features for classification problems. The proposed method employs fuzzification of original features based on irregular-shaped membership functions created by genetic algorithm and particle swarm optimization, and a feature selection process using two criterion functions to evaluate feature subsets. The first function is applied to eliminate features with redundant effects, and the second function is applied to select a feature subset that maximizes inter-class distances and minimize intra-class distances. Standard machine learning data sets in various sizes and complexities are used in experiments. The results show that the proposed technique is effective and performs well in comparisons with other research.

1 Introduction

A feature selection method selects a small subset of highly predictive features from the original set of features. It most of the time yields better results due to reduction of noises and distractions, and takes less training time for a classifier than using the entire set of features. Feature selection approaches can be classified into three categories which are wrapper, filter, and hybrid approaches.

Given a classification problem, a wrapper method incorporates the classification itself in the feature evaluation process. To evaluate a candidate feature subset, a classification model is built and used to evaluate the set. Maroño et al. [11] propose a wrapper based feature selection using ANOVA decomposition and functional networks to calculate global sensitivity indices. Features with high index values are selected. Zhuo et al. [19] use a genetic algorithm (GA) to optimize a support vector machine (SVM) kernel parameters for selecting a feature subset. The fitness function is accuracy which also is used as the criterion function for selecting features. The wrapper approach is expected to return a subset of features that yields high accuracy since every candidate feature set is evaluated by the classifier that is used in the problem. Since classification models are trained and tested many times, and data becomes larger in dimensionality and number of instances, this approach takes a long time for learning such data and in many cases is inapplicable.

C. Sombattheera et al. (Eds.): MIWAI 2016, LNAI 10053, pp. 173–183, 2016.
DOI: 10.1007/978-3-319-49397-8_15

In a filter method, instead of performing classification as part of the feature selection process, a quality measure is used to evaluate each feature set. The filter-based approach composes two important components which are a selection algorithm and a criterion function. The selection algorithm creates candidate features while the criterion function selects features and evaluates feature subsets. The criterion function can be independent from the classification model, but it should be suitable for the problem. The filter-based approach generally takes less time than does the wrapper approach since no classifier is trained and tested as in the wrapper approach. It is more preferable for real-world problems, especially those with large data sets. Many researchers find that it yields subsets with lower accuracy than do the other two approaches. However, it is not true to state that the filter approach always gives lower accuracy. Some criterion functions may return subsets with equivalent or better performance than other approaches.

Yu and Liu [15] use symmetrical uncertainty as the measure to select features relevant to classes which are not redundant with other selected features. Zhou et al. [18] propose a forward algorithm to select features using conditional maximum entropy modeling to approximate the gain for features. Fleuret [4] uses conditional mutual information (CMI) as the criterion function to fasten the forward search process. Haindl et al. [6] propose a backward filter-based feature selection method based on mutual correlation, a similarity measure between two variables, to select features which are uncorrelated.

The hybrid approach takes advantage of both the wrapper and the filter approaches. It applies a filter-based technique to select highly significant features and applies a wrapper-based technique to add candidate features and evaluate candidate sets. Zhang et al. [17] apply the RELIEFF algorithm to estimate the quality of attributes according to how well their values distinguish between instances that are close to each other, and apply GA with classifier accuracy as the fitness function to search for an optimal feature subset. Somol et al. [13] present a hybrid floating search, named hSFFS, by applying a filter criterion function first to filter some features and applying a wrapper criterion to generate a candidate set. After that a wrapper criterion function is applied to select the best feature from the candidate set. This is a wrapper-dominating hybrid method. Gan et al. [5] propose an alternative to hSFFS, which is a filter-dominating hybrid method. A filter criterion is used to select the best feature from an unselected set, and a wrapper criterion is then used to evaluate a feature subset.

Problems usually found in real-world applications are mixtures of ambiguous and noisy data. This results in an inaccurate classification model. Fuzzy Logic, which is a multi-value logic that allows intermediate values to be defined between conventional crisp evaluations, e.g., true/false, yes/no, etc., provides a simple way to define conclusions based upon vague, ambiguous, imprecise, noisy, or missing input information [3]. Membership functions for fuzzy sets can be of any shape or type, such as triangular, trapezoidal, and Gaussian-shaped, as determined by experts in the domain over which the sets are defined.

This paper proposes a feature selection technique for classification using two criterion functions and feature fuzzification using irregular-shaped membership functions, evolved by genetic algorithm and particle swarm optimization. The technique is evaluated using standard machine learning data sets in various sizes and complexities.

2 Proposed Feature Fuzzification

Irregular-shaped membership functions for every continuous attribute are evolved. Values of those attributes are fuzzified to create a suitable set of value ranges. All attributes are then fed into the filter-based feature selection algorithm which employs two criterion functions to generate the best set of predictive features.

The membership function (MF) shape determined in advance by experts may not be suitable for a specific problem at hand, especially those with large and complex search spaces. We convert the wrapper-based hierarchical co-evolutionary by Huang et al. [7] for generating irregular-shaped membership functions (ISMFs) into a filter-based algorithm using two optimization techniques: genetic algorithm and particle swarm optimization, where a criterion function is used in order to improve efficiency. An MF shape is represented as one pivot point, left shoulder points, and right shoulder points, depicted in Fig. 1.

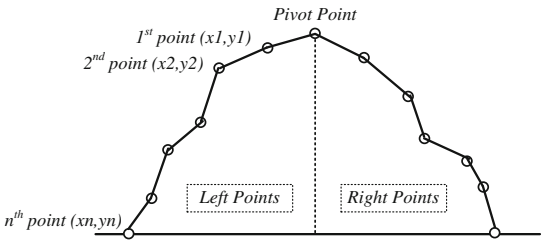

Fig. 1. An irregular-shaped membership function

2.1 Membership Function Evolution by Genetic Algorithm

A genetic algorithm can be employed to create membership functions for continuous variables. An irregular-shaped MF is represented as one pivot point, left shoulder points and right shoulder points, as shown in Fig. 2(a). Fuzzy partitions on each input variable are encoded in genetic segmentations and concatenated into one chromosome in the first level (L1-level) for the corresponding variable. A chromosome in the second level (L2-level) composes of genes pointing to chromosomes for all variables in L1-level. An L2-level gene contains the integer value of an index in the L1-level chromosome. With GA operations (crossover, mutation and selection), coordinates of points will be changed, and it results in changing shapes. Constraints and repairing schemes are applied before decoding the genetic representation.

The algorithm partitions and encodes possible solutions as populations in different levels, allowing for different kinds of chromosomes and genetic operations. A higher level chromosome selects a set of lower-level chromosomes to form a solution. In this case, a highly complicated search task can be partitioned into several subtasks which are simultaneously and effectively handled. The structure of the chromosome is shown in Fig. 2(b).

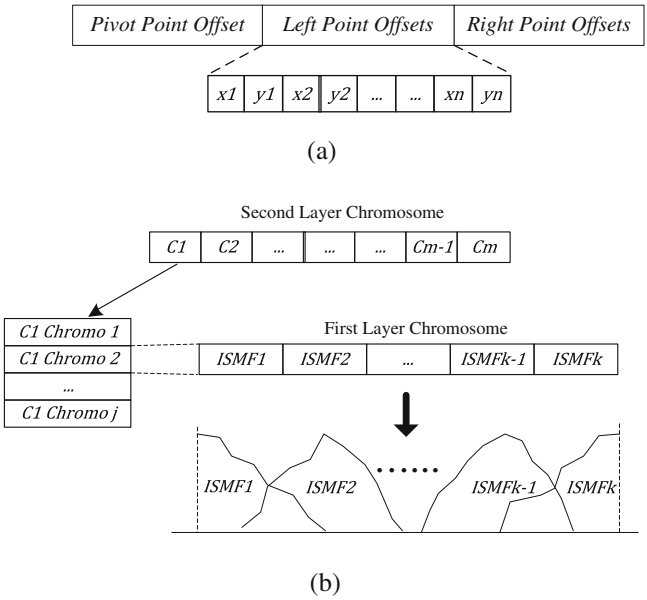

Fig. 2. Chromosome structure

2.2 Membership Function Evolution by Particle Swarm Optimization

Particle swarm optimization (PSO) [2] can be used to evolve optimal locations of points on ISMFs for a feature. PSO is a heuristic global optimization method based on swarm intelligence. Potential solutions, called particles, fly through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. This value is called *pbest*. Another best value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the neighbors of the particle. In this research, a particle takes the entire population as its topological neighbors, the best value is a global best and is called *gbest*. At each time step, we change the velocity of (accelerating) each particle toward its pbest and the gbest locations. Acceleration is weighted randomly toward the pbest and the gbest locations. Content of a PSO particle for generating ISMFs is shown in Fig. 3.

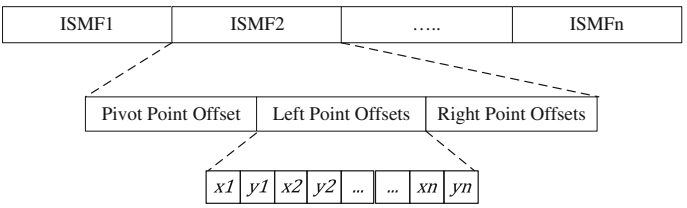

Fig. 3. Particle content

Evolution in PSO is the process to update particles' positions. A particle position is updated as follows:

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$$

and

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)\left[b_{ij}(t) - x_{ij}(t)\right] + c_2 r_{2j}(t)\left[\hat{b}(t) - x_{ij}(t)\right]$$

where x_{ij} is the vector of i-th particle with j dimensions, t denotes a discrete time step or iteration, w is the inertia weight, r is a random number in the range [0..1] sampled from a uniform distribution, c is an acceleration constant, $\hat{b}(t)$ is the best position among all particles, and $b_{ij}\ (t)$ is the best position of the i-th particle.

To determine the values of three important parameters needed by PSO which are w, c_1 and c_2, Zhang et al. [16] constructs a relationship between the dynamic process of particle swarm optimization and the transition process of a control system. It reduces the three parameters to the percentage overshoot, and from their experiments the value should fall between 0.6 and 0.8. The percentage overshoot allows us to determine the values of w and c, and further $c = c_1 = c_2$. Comparing to other parameter setting strategies, this method leads to similar optimization results but faster convergence.

Opposition-based PSO technique [8] is used to initialize particles to preserve the coverage. The process can be described as:

1. Randomly initialize n particles.
2. Calculate opposite particles of first n particles.
3. Evaluate $2n$ particles from steps 1 and 2, and select the best n particles to be in the swarm of optimization process.

3 Proposed Feature Selection Process

We improve the filter-based sequential forward floating search algorithm [12] by employing two criterion functions with different characteristics to complement each other and allowing more thorough search for features by introducing candidate sets. Conditional mutual information (CMI) is employed as the first criterion function. It measures dependency between two variables with respect to a class, conditional to the response of features already picked [4]. CMI selects features which maximize MI to the target class where such information must not have been caught by features already selected to reduce redundant features. It generates a candidate set of features which are suitable to be added to or removed from a selected subset instead of examining one feature at a time. Using the candidate sets makes the search more thorough. The second criterion function selects a feature to be added or removed from this set.

Input to the algorithm consists of the original feature set S, the first criterion function J_1, and the second criterion function J_2. Let D be the total number of original features. d_{sel} is the number of selected features. d_{cand} is the number of features in a candidate set where $d_{cand} \geq 1$. S_{sel} is the selected feature subset. S^{-}_{cand} is the candidate set in the backward step, and S^{+}_{cand} is the candidate set in the forward step.

In the forward step, unselected features are evaluated by the a criterion function J_1 and sorted in descending order. A candidate feature set is created as follows:

$$S^{+}_{cand} = \{x_n | x_n \in S \backslash S_{sel} \, and \, n = [1 \ldots d_{cand}] \; and \, J_1(x_1) \geq J_1(x_2) \cdots \geq J_1(x_n)\}$$

where $J_1 = min\, I(Y; X_n | S_{sel})$ where $X_n \in S \backslash S_{sel}$

As mentioned earlier, CMI is used as J_1, and it can be calculated as follows:

$$I(Y; X_n | X_m) = H(Y, X_m) - H(X_m) - H(Y, X_n, X_m) + H(X_n, X_m)$$

where *I(Y; Xn|Xm)* is the conditional mutual information between Y and X_n given X_m, and H is an entropy function. For more information on how to compute CMI, see [4].

The feature selected is the one when combined with the previously selected subset of size k gives the best subset when evaluated with J_2, forming the selected subset of size $k + 1$. Then the algorithm compares the new subset with the previously selected subset of size $k + 1$ and retains the better one.

In the backward step, a feature to be removed must be the one providing the least information to target classes, and its information has been caught by features already picked. Therefore, J_1 in the backward step is calculated as follows:

$$J_1 = max\, I(Y; X_n | S_{sel} \backslash X_n) \; where \; X_n \in S_{sel}$$

Selected features are evaluated by J_1 and sorted in ascending order. A candidate set is generated as follows:

$$S^{-}_{cand} = \{x_n | x_n \in S \backslash S_{sel} \, and \, n = [1 \ldots d_{cand}] \; and \, J_1(x_1) \leq J_1(x_2) \leq \cdots \leq J_1(x_n)\}$$

The feature to be removed is the one when removed from the selected subset yields the best subset with k features according to J_2. The algorithm compares the new subset and the previously selected subset of size k and retains the better one. The exclusion step continues to smaller subsets if the new subset is better, or else the algorithm goes back to the inclusion step. The algorithm terminates when the selected subset size is $d_{sel} + \Delta$.

3.1 The Second Criterion Function

As part of the feature selection process, the second criterion function (J_2)'s role is to select a feature subset that maximizes inter-class distances and minimizes intra-class distances. Three effective measures are studied as candidates for J_2.

Mutual Information (MI). MI can be calculated as follows:

$$I(Y; X_n) = H(Y) + H(X_n) - H(Y, X_n)$$

where H is an entropy function, Y is a class attribute, and X_n is the feature to be selected.

Jeffreys-Matusita Distance Bound to the Bayes Error (JMBH). JMBH can be calculated as follows:

$$J_{bh} = \sum_{i=1}^{c} \sum_{j=1}^{c} \sqrt{P(\omega_i)P(\omega_j)} J_{ij}^2$$

$$J_{ij} = \left[2(1 - e^{-B_{ij}})\right]^{1/2}$$

$$B_{ij} = \frac{1}{8}(m_i - m_j)^t \left(\frac{\Sigma_i + \Sigma_j}{2}\right)^{-1} (m_i - m_j) + \frac{1}{2}\log\left[\frac{\left(\frac{\Sigma_i + \Sigma_j}{2}\right)}{\sqrt{|\Sigma_i||\Sigma_j|}}\right]$$

Where m_i, m_j and Σ_i, Σ_j are mean vectors and covariance matrices for the classes ω_i and ω_j, respectively.

Mahalanobis Distance (MAHA). MAHA can be calculated as follows:

$$D_M(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)}$$

where μ is the mean vector, and S is the covariance matrix for a group.

3.2 Classification Model

Classification and Regression Trees (CART) was introduced by Breiman et al. [1]. CART is based on a fundamental idea that each split should be selected so that the data in each descendant subset is purer than the data in the parent node. The node impurity is largest when all classes are equally mixed together and smallest when the node contains only one class. CART produces binary splits. Hence, it produces binary trees. CART uses Gini impurity index as an attribute selection measure to build a decision tree. Consider a parent node m, which contains the data that belongs to the jth class. The impurity function for node t is given by $i(t) = 1 - \sum_i p^2(j|m)$. The decrease of split impurity is given by $\Delta i(\delta, t) = i(t) - p_L i(m_L) - p_R i(m_R)$, where t is a parent node using a splitting coefficient δ to split into two nodes m_L and m_R. The split with the largest decrease in impurity is chosen for that particular node.

4 Experimental Evaluation

The data sets used in the experiments using standard data sets from the UCI machine learning repository. For any data set without a separate test set provided, a 10-fold cross validation is employed to measure performance. The stopping criterion for genetic algorithm and PSO is set at 100 iterations.

Table 1. Classification accuracy of applying and not applying feature fuzzification by genetic algorithm and particle swarm optimization, with 3 different J_2 criterion functions

Data Set	Feature fuzzification by Genetic algorithm					
	MI		JMBH		MAHA	
	Fuzzified Features	Non-Fuzzified Features	Fuzzified Features	Non-Fuzzified Features	Fuzzified Features	Non-Fuzzified Features
Wine	94.44	88.89	100	100	100	88.89
Pima	75.33	75.33	79.22	76.63	75.33	75.33
Image segmentation	92.57	90.62	92.57	90.19	92.57	90.62
Breast cancer	96.49	92.98	98.24	96.49	98.25	96.49
Sonar	95.23	80.95	95.23	85.71	95.23	80.95
Hill with noise	59.4	56.6	59.9	56.6	59.08	56.6
Arrhythmia	80	62.22	82.22	60	66.67	62.22
Madelon	78.33	69.83	84.83	74	81.5	76
Data set	Feature fuzzification by Particle swarm optimization					
	MI		JMBH		MAHA	
	Fuzzified features	Non-fuzzified features	Fuzzified features	Non-fuzzified features	Fuzzified features	Non-fuzzified features
Wine	94.44	94.44	100	94.44	100	94.44
Pima	75.33	76.32	76.62	73.68	75.33	76.32
Image segmentation	91.48	91.48	90.81	90.81	90.95	90.95
Breast cancer	94.74	91.23	96.49	94.74	96.49	94.74
Sonar	80.95	100	88.28	85	85.71	100
Hill with noise	58.58	58.58	61.67	57.43	57.43	57.43
Arrhythmia	74.19	65.22	80.64	60.87	67.74	69.56
Madelon	80.17	80.17	82.67	82.67	85.67	85.67

4.1 Effectiveness of Feature Fuzzification

In this experiment, we study the effectiveness of the fuzzification process using both genetic algorithm and particle swarm optimization against not using the fuzzification at all in different classification problems. An initial study shows that the swarm size of 30 particles gives the highest accuracy and will be used in all experiments. The results are shown in Table 1. We can see that in almost all configurations using fuzzification yields higher accuracy then not using it, thus the fuzzification is useful. In addition, the configuration that gives the best results is fuzzification using GA and JMBH as J_2 function (referred to as Fuzzified GA+JMBH).

4.2 Performance of the Proposed Technique

Since fuzzification and JMBH are beneficial to the performance of the proposed feature selection technique, in this section we focus more on the feature reduction abilities of fuzzification by genetic algorithm and particle swarm optimization. The results in

Table 2 show that although GA yields higher accuracies in general, however, PSO tends to give better feature reduction rates.

Table 2. Accuracies and feature reduction abilities of fuzzification by GA and PSO

Data Set		Original features	Fuzzified GA +JMBH	Fuzzified PSO +JMBH
Wine	Accuracy	83.33	100	100
	Features	14	3	2
Pima	Accuracy	70.13	79.22	76.62
	Features	8	4	2
Image segmentation	Accuracy	90.29	92.57	90.81
	Features	20	9	9
Breast cancer	Accuracy	89.47	98.24	96.49
	Features	31	4	3
Sonar	Accuracy	71.43	95.23	88.28
	Features	61	5	5
Hill valley with noise	Accuracy	60.07	59.9	61.67
	Features	101	12	1
Arrhythmia	Accuracy	66.67	82.22	80.64
	Features	280	20	6
Madelon	Accuracy	75.67	84.83	82.67
	Features	501	12	9

4.3 Comparisons with Other Research

Lastly, the proposed method (Fuzzified GA+JMBH) is compared against three recent research on fuzzy-based feature selection which are: Jalali et al. [9], Vieira et al. [14], Li and Wu [10], using the performance numbers reported in each paper. The results (in Table 3) show that the proposed method outperforms [9] and [10] in all common data sets. Comparing with [14], we find that the proposed method gives higher accuracy in

Table 3. Results of (Fuzzified GA + JMBH) compared to other previous fuzzy-based research. Feature reduction percentages relative to the original feature sets are shown in parentheses.

Data Set	Original number of features	Jalali et al. [9]	Vieira et al. [14]	Li and Wu [10]	Proposed Method
Pima	8	–	–	71.51 (89.17)	80.52 (50)
Wine	14	95.4 (65.38)	96 (69.23)	90.99 (83.59)	100 (76.92)
Breast cancer	31	63.6 (91.66)	98 (86.67)	95.97 (96.67)	98.24 (87.09)
Sonar	61	–	86 (86.66)	68.06 (96.78)	95.24 (93.33)
Arrhythmia	280	–	87 (95.69)	–	82.22 (92.83)

3 out of 4 data sets. Thus, the proposed technique is shown to perform very well across different data sets and in comparison with other techniques.

5 Conclusion

As data sets grow in size and complexity, a feature selection technique is needed to select a small subset of highly predictive features from the entire set of features. The technique is expected to reduce noises and distractions, thus improve both effectiveness and efficiency of machine learning. This paper presents a new filter-based technique to select a minimal set of features for classification problems. The proposed technique employs fuzzification of original features using irregular-shaped membership functions evolved by genetic algorithm and particle swarm optimization, and a filter-based feature selection using two criterion functions where the first function is applied to eliminate features with redundant effects, and the second function is used to select a feature subset that maximizes inter-class distances and minimize intra-class distances. The technique is evaluated using standard UCI data sets and compared to recent fuzzy-based feature selection research papers. The results show that feature selection improves classification accuracy; that the use of evolutionary feature fuzzification and two criterion functions enhances the performance of feature selection; and that the best configuration is using Jeffreys-Matusita Distance Bound to the Bayes Error as the second criterion function and genetic algorithm to evolve irregular-shaped fuzzy membership functions. In addition, the proposed technique performs well in comparison to previous research on common data sets.

References

1. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworht, Pacific Grove (1984)
2. Kennedy, J., Eberhart, B.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Network, Pert, Australia, pp. 1942–1948 (1995)
3. Engelbrecht, A.P.: Computational Intelligence: An Introduction, 2nd edn. Wiley, New York (2007)
4. Fleuret, F.: Fast binary feature selection with conditional mutual information. J. Mach. Learn. Res. **5**(11), 1531–1555 (2004)
5. Gan, J.Q., Awwad Shiekh Hasan, B., Tsui, C.S.L.: A hybrid approach to feature subset selection for brain-computer interface design. In: Yin, H., Wang, W., Rayward-Smith, V. (eds.) IDEAL 2011. LNCS, vol. 6936, pp. 279–286. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23878-9_34
6. Haindl, M., Somol, P., Ververidis, D., Kotropoulos, C.: Feature selection based on mutual correlation. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 569–577. Springer, Heidelberg (2006). doi:10.1007/11892755_59
7. Huang, H., Pasquier, M., Quek, C.: HiCEFS – A Hierarchical Coevolutionary Approach for the Dynamic Generation of Fuzzy System, pp. 3426–3443. IEEE Congress on Evolutionary Computation, CEC (2007)

8. Jabeen, H., Jalil, Z., Baig, A.: Opposition based initialization in particle swarm optimization (O-PSO). In: Proceedings of Genetic and Evolutionary Computation Conference, Montreal, Canada, pp. 2047–2052 (2009)
9. Jalali, L., Nasiri, M., Minaei, B.: A hybrid feature selection method based on fuzzy feature selection and consistency measures. In: Intelligent Computing and Intelligent System (ICIS), pp. 718–722 (2009)
10. Li, Y., Wu, Z.F.: Fuzzy feature selection based on min-max learning rule and extension matrix. Pattern Recogn. **41**, 217–226 (2008)
11. Maroño, N.S., Betanzos, A.A., Castillo, E.: A new wrapper method for feature subset selection. In: Proceedings-European Symposium on Artificial Neural Networks, pp. 515–520 (2005)
12. Pudil, P., Novovičová, J., Kittler, J.: Floating search methods in feature selection. Pattern Recogn. Lett. **15**, 1119–1125 (1994)
13. Somol, P., Novovičová, J., Pudil, P.: Flexible-hybrid sequential floating search in statistical feature selection. In: Yeung, D.-Y., Kwok, J.T., Fred, A., Roli, F., Ridder, D. (eds.) SSPR/SPR 2006. LNCS, vol. 4109, pp. 632–639. Springer, Heidelberg (2006). doi:10.1007/11815921_69
14. Vieira, S.M., Sousa, J.M.C., Kaymak, U.: Fuzzy criteria for feature selection. Fuzzy Sets Syst. **189**, 1–18 (2012)
15. Yu, L., Liu, H.: Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003) (2003)
16. Zhang, W., Ma, D., Wei, J., Liang, H.: A parameter selection strategy for particle swarm optimization based on particle positions. Expert Syst. Appl. **41**, 3576–3584 (2014)
17. Zhang, L.X., Wang, J.X., Zhao, Y.N., Yang, Z.H.: A novel hybrid feature selection algorithm: using relief estimation for GA-wrapper search. In: Proceedings of the Second International Conference on Machine Learning and Cybernetics, pp. 380–384 (2003)
18. Zhou, Y., Weng, F., Wu, L., Schmidt, H.: A fast algorithm for feature selection in conditional maximum entropy modeling. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pp. 153–159 (2003)
19. Zhuo, L., Zheng, J., Wang, F., Li, X., Ai, B., Qian, J.: A genetic algorithm based wrapper feature selection method for classification of hyperspectral images using support vector machine. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. **XXXVII Par B7**, 397–402 (2008)