# Feature Selection

Gheith A. Abandah

## 1 Introduction

*Feature selection* is typically a search problem for finding an optimal or suboptimal subset of *m* features out of original *M* features. Feature selection is important in many pattern recognition problems for excluding irrelevant and redundant features. It allows reducing system complexity and processing time and often improves the recognition accuracy [1]. For large number of features, exhaustive search for best subset out of $2^M$ possible subsets is infeasible. Therefore, many feature subset selection algorithms have been proposed.

These algorithms can generally be classified as *wrapper* or *filter* algorithms according to the criterion function used in searching for good features. In a wrapper algorithm, the performance of the classifier is used to evaluate the feature subsets. In a filter algorithm, some feature evaluation function is used rather than optimizing the classifier's performance. Many feature evaluation functions have been used particularly functions that measure distance, information, dependency, and consistency [2]. Wrapper methods are usually slower than filter methods but offer better performance.

The simplest feature selection methods select *best individual features*. A feature evaluation function is used to rank individual features, then the highest ranked *m* features are selected. Although these methods can exclude irrelevant features, they often include redundant features. "The *m* best features are not the best *m* features" [3].

Many *sequential* and *random* search algorithms have been used in feature subset selection [4]. The sequential search methods are variations of sequential forward selection, sequential backward elimination, and bidirectional selection. These algorithms are simple to implement and fast; they have time complexity of $O(M^2)$ or less. However, as they don't perform complete search, they may miss the optimal feature subset.

One sequential forward selection algorithm is the *fast correlation-based filter* (FCBF) [5]. This algorithm performs relevance and redundancy analyses using symmetric uncertainty. FCBF creates the feature subset by sequentially adding features in decreasing relevance order while excluding redundant features. The redundancy analysis excludes redundant features whenever a new feature is added to the subset based on one-to-one comparison between the added feature and the remaining features.

The *minimal-redundancy-maximal-relevance* (mRMR) algorithm is another sequential forward selection algorithm [3]. It uses mutual information to analyze relevance and redundancy. However, mRMR grows the selected subset by adding the feature that has the maximum difference between its relevance measure and its aggregate redundancy measure with the already selected features.

*Genetic algorithms* are random search algorithms and often offer efficient solutions to general NP-complete problems. They can explore large, nonlinear search space by performing simultaneous search in many regions. A population of solutions is evaluated using some fitness function. In feature selection, this fitness function usually calls the classifier to evaluate the population's individuals (feature subsets); constituting a wrapper algorithm. The individuals' fitness is then used to select individuals for breeding and producing the next generation. *Multi-objective genetic algorithms* (MOGA) have been successfully used in feature selection [6]. MOGA have the advantage of generating a set of alternative solutions. In feature selection, they are typically used to optimize the classifier error rate and the number of features. Thus, a set of solutions of feature subsets of varying sizes is found.

In this paper, we concentrate on improving the feature extraction stage by selecting efficient subset of features. Figure 1 summarizes the methodology used in this paper. We extract 96 features from a database of handwritten Arabic letter forms. These features are often used in Arabic character recognition [8]. We use five feature selection techniques to select and recommend good features for recognizing handwritten Arabic letters. We analyze the recognition accuracy as a function of the feature subset size using three popular classifiers.

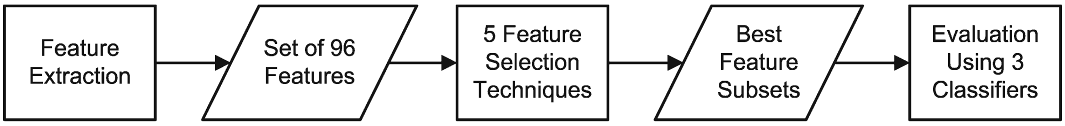
Feature Extraction
Set of 96 Features
5 Feature Selection Techniques
Best Feature Subsets
Evaluation Using 3 Classifiers

**Fig. 1** Methodology of Feature Extraction, Selection, and Evaluation

This paper is organized in six sections. Section 2 overviews the related work. Section 3 describes the five feature selection techniques. Section 4 describes three classifiers used to evaluate feature subsets. Section 5 analyzes the classification accuracy as a function of the feature subset size.

## 2 Related works

There are many good papers on feature selection [1, 2, 3, 4, 5, 12, 13]. Recent problems in feature selection include feature selection for ensembles of classifiers and building efficient classifiers using weak features [14, 15, 16]. Additionally, there are some papers specialized in feature selection for handwritten script recognition [6, 14, 17].

Many researchers have used genetic algorithms for feature selection [18, 19]. After Emmanouilidis *et al*. have suggested using multi-objective genetic algorithms for feature selection [20], several researchers started to use MOGA in feature selection. Oliveira *et al*. used MOGA feature selection for recognition of handwritten digit strings [6]. Morita *et al*. used MOGA in unsupervised feature selection for handwritten words recognition [17]. And Oliveira *et al*. used MOGA for selecting features for ensembles of classifiers [15]. We are unaware of any work that uses MOGA, FCBF, or mRMR for feature selection in handwritten Arabic letter recognition.

Feature selection has been addressed by several researchers working on building solutions for recognizing printed and handwritten Arabic text as early as Nough *et al*.'s work in the 1980s [21]. More recently, Khedher *et al*. optimized feature selection for recognizing handwritten Arabic characters and gave higher weights for better features [22]. Pechwitz *et al*.  made a comparison between two feature sets of handwritten Arabic words: pixel features extracted using a sliding window with three columns and skeleton direction features extracted in five zones using overlapping frames [23]. El Abed and Margner made a comparison among three feature extraction methods: sliding window with pixel feature, skeleton direction-based features, and sliding window with local features [24]. Abandah *et al*. used mRMR to select four sets of features for four classifiers each specialized in recognizing letters of the four letter forms [25].

There are many used feature extraction methods for offline recognition of characters. These methods are extracted from the character's binary image, boundary, or skeleton [26, 27]. Amin *et al*. extracted from the skeleton of thinned printed Arabic characters feature points, loops, lines, and curves [28]. Kavianifar and Amin used features extracted from the boundary to recognize multi-font printed Arabic scripts [29]. El-Hajj *et al*. used baseline dependent features such as distributions and concavities for recognizing handwritten Arabic words [30]. The feature extraction methods used in this research are used in other Arabic character recognition systems such as [24, 30, 31, 32].

Good progress has been made in recognizing handwritten Arabic script. Sari *et al*. use morphological features of the Arabic letters such as turning points, holes, ascenders, descenders, and dots for segmentation and recognition [33]. Menasri *et al*. identified letter body alphabet for handwritten Arabic letters; they classified Arabic letters into root shapes and optional tails. Multiple Arabic letters that only differ in the existence and number of dots are mapped to the same root shape. This alphabet also includes common vertical ligatures of joined letters [34]. AlKhateeb *et al*. use DCT features and neural network classifier. They discard 80% of the DCT coefficients without sacrificing the recognition accuracy [35].

# 3 Feature selection

This section describes the feature subset selection techniques used in this paper. These techniques include two best individual features methods: scatter criterion and the symmetric uncertainty, two heuristic search methods: FCBF and mRMR, and one random search method using MOGA.

Feature subset selection is applied on a set of feature values $x_{ijk}$; $i = 1, 2, \ldots, N$; $j = 1, 2, \ldots, C$; and $k = 1, 2, \ldots, M$, where $x_{ijk}$ is the *ith* sample of the *jth* letter form (class) of the *kth* feature. Therefore, the average of the *kth* feature for letter form $\omega_j$ is

$$\bar{x}_{jk} = \frac{1}{N} \sum_{i=1}^{N} x_{ijk} . \qquad (1)$$

And the overall average of the *kth* feature is

$$\bar{x}_{k} = \frac{1}{C} \sum_{j=1}^{C} \bar{x}_{jk} . \qquad (2)$$

### 3.1 Scatter criterion (*J*)

One approach to select features is to select the features that have highest values of the *scatter criterion* $J_k$, which is a ratio of the mixture scatter to the within-class scatter [36, 37, 38, 39]. The *within-class scatter* of the *kth* feature is

$$S_{w,k} = \sum_{j=1}^{C} P_j \, S_{jk} \,, \qquad (3)$$

where $S_{jk}$ is the variance of class $\omega_j$, and $P_j$ is the priori probability of this class and found by:

$$S_{jk} = \frac{1}{N} \sum_{i=1}^{N} (x_{ijk} - \bar{x}_{jk})^2 \text{ and } P_j = \frac{1}{C} . \qquad (4)$$

The *between-class scatter* is the variance of the class centers with respect to the global center and is found by

$$S_{b,k} = \sum_{j=1}^{C} P_j \, (\bar{x}_{jk} - \bar{x}_k)^2 \, . \qquad (5)$$

And the *mixture scatter* is the sum of the within and between-class scatters, and equals the variance of all values with respect to the global center.

$$S_{m,k} = S_{w,k} + S_{b,k} = \frac{1}{CN} \sum_{j=1}^{C} \sum_{i=1}^{N} (x_{ijk} - \bar{x}_k)^2 \qquad (6)$$

The scatter criterion $J_k$ of the *kth* feature is

$$J_k = \frac{S_{m,k}}{S_{w,k}} \, . \qquad (7)$$

Higher value of this ratio indicates that the feature has high ability in separating the various classes into distinct clusters.

### 3.2 Symmetric uncertainty (SU)

Another approach to select features is to select the features that have highest *symmetric uncertainty* (SU) values between the feature and the target classes [1, 3, 39]. To find this indicator, we first normalize the feature values for zero mean and unit variance by

$$\hat{x}_{ijk} = \frac{x_{ijk} - \bar{x}_k}{\sigma_k} \;,\; \sigma_k^2 = \frac{1}{CN}\sum_{j=1}^{C}\sum_{i=1}^{N}(x_{ijk} - \bar{x}_k)^2 \,. \qquad (8)$$

Then the normalized values of continuous features are discretized into $L$ finite levels to facilitate finding probabilities. The corresponding discrete values are $\tilde{x}_{ijk}$. The *mutual information* of the *kth* feature is

$$I(\mathbf{x}_k, \boldsymbol{\omega}) = \sum_{l=1}^{L}\sum_{j=1}^{C} P(\tilde{x}_{ljk}, \omega_j)\log_2 \frac{P(\tilde{x}_{ljk}, \omega_j)}{P(\tilde{x}_{ljk})P(\omega_j)} \,, \qquad (9)$$

where $P(\omega_j)$ is the prior probability of class $\omega_j$, $P(\tilde{x}_{lk})$ is the distribution of the *kth* feature, and $P(\tilde{x}_{lk}, \omega_j)$ is the joint probability. This indicator measures how much the distribution of the feature values and target classes differ from statistical independence. This is a nonlinear estimation of correlation between the feature values and target classes. The *symmetric uncertainty* (SU) is derived from the mutual information by normalizing it to the entropies of the feature values and target classes.

$$SU(\mathbf{x}_k, \boldsymbol{\omega}) = 2\left(\frac{I(\mathbf{x}_k, \boldsymbol{\omega})}{H(\mathbf{x}_k) + H(\boldsymbol{\omega})}\right), \qquad (10)$$

where the entropy of variable $X$ is found by $H(X) = -\sum_i P(x_i)\log_2 P(x_i)$.

### 3.3 Fast correlation-based filter (FCBF)

The *fast correlation-based filter* (FCBF) algorithm aims to select a subset of relevant features and exclude redundant features. FCBF uses the symmetric uncertainty $SU(\mathbf{x}_k, \boldsymbol{\omega})$ to estimate the relevance of feature $k$ to the target classes. It also uses the symmetric uncertainty between two features $k$ and $o$ $SU(\mathbf{x}_k, \mathbf{x}_o)$ to approximate the redundancy between the two features. This algorithm grows a subset of *predominant* features by adding the relevant features to the empty set in descending $SU(\mathbf{x}_k, \boldsymbol{\omega})$ order. Whenever feature $k$ is added, FCBF excludes from consideration for addition to the subset all remaining redundant features $o$ that have $SU(\mathbf{x}_k, \mathbf{x}_o) \geq SU(\mathbf{x}_o, \boldsymbol{\omega})$. In other words, it excludes all features that their respective correlation with already selected features is larger than or equals their correlation with the target classes.

### 3.4 Minimal-redundancy-maximal-relevance (mRMR)

Similar to FCBF, the *minimal-redundancy-maximal-relevance* (mRMR) algorithm is another forward selection search algorithm for feature selection [3]. mRMR uses the mutual information to select best *m* features that have minimal redundancy and maximal relevance criterion.

For the complete set of features *X*, the subset *S* of *m* features that has the *maximal relevance* criterion is the subset that satisfies the maximal mean value of all mutual information values between individual features $\mathbf{x}_i$ and class $\boldsymbol{\omega}$.

$$\max D(S,\boldsymbol{\omega}), \quad D = \frac{1}{m}\sum_{\mathbf{x}_i \in S} I(\mathbf{x}_i, \boldsymbol{\omega}) \qquad (11)$$

The subset *S* of *m* features that has the *minimal redundancy* criterion is the subset that satisfies the minimal mean value of all mutual information values between all pairs of features $\mathbf{x}_i$ and $\mathbf{x}_j$.

$$\min R(S), \quad R = \frac{1}{m^2}\sum_{\mathbf{x}_i, \mathbf{x}_j \in S} I(\mathbf{x}_i, \mathbf{x}_j) \qquad (12)$$

In the mRMR algorithm, the subset *S* of *m* best features is grown iteratively using forward search algorithm. The following criterion is used to add the $\mathbf{x}_j$ feature to the previous subset of $m-1$ features:

$$\max_{\mathbf{x}_j \in X - S_{m-1}} \left[ I(\mathbf{x}_j, \boldsymbol{\omega}) - \frac{1}{m-1}\sum_{\mathbf{x}_i \in S_{m-1}} I(\mathbf{x}_i, \mathbf{x}_j) \right] \qquad (13)$$

### 3.5 Non-dominated sorting genetic algorithm (NSGA)

The *non-dominated sorting genetic algorithm* (NSGA) is an efficient algorithm for multi-objective evolutionary optimization [40, 41]. We use NSGA to search for optimal set of solutions with two objectives:

i. Minimize the number of features used in classification.

ii. Minimize the classification error.

This algorithm searches for a set of optimal solutions on a front called the *Pareto-optimal front*. Figure 2 shows an example Pareto-optimal front and a population of solutions found in optimizing the number of features and the classification error. This front is the set of *non-dominated* solutions among this population. A non-dominated solution is one that does not have any other solution that dominates it. Solution $S^{(1)}$ dominates Solution $S^{(2)}$ when no objective value of $S^{(2)}$ is less than $S^{(1)}$ and at least one value of $S^{(2)}$ is strictly greater than $S^{(1)}$. In this two-objective case, a non-dominated solution of *m* features is the solution that has the smallest classification error among all solutions that have *m* features.

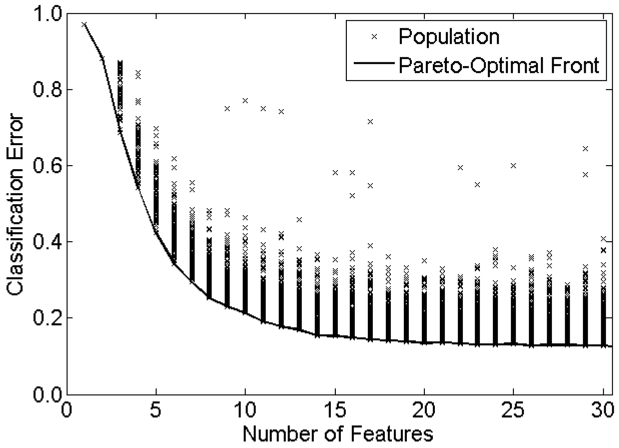

**Fig. 2** Example Pareto-Optimal Front and Population Examined by NSGA

Similar to other genetic algorithms, NSGA evolves a random population of solutions from one generation to the next. In every generation, the fitness of every individual solution is evaluated and the best individuals are selected to breed the next generation. In search of better individuals, crossover and mutation are used when generating a new generation. The NSGA differs from other genetic algorithms in how best individuals are selected. The selection method ranks individuals after evaluating their objective values based on non-domination criterion. A front of non-dominated individuals is identified and assigned a dummy large fitness value. This fitness value is degraded for clustered individuals to maintain diversity in the population. The non-dominated front is removed and successive fronts are identified and given dummy fitness values smaller than the smallest value in the previous front until the entire population is identified and ranked. Thus the multiple objective values are reduced to this dummy fitness which is used to select best individuals for breeding to find the Pareto-optimal front.

# 4 Classifiers

To ensure that our results are not restricted to a specific classifier, we use three widely-used classifiers: *k*-nearest neighbor (*k*-NN), linear discriminant analysis (LDA), and support vector machine (SVM) [39]. These classifiers are often used in evaluating various feature sets [3, 13]. Therefore, we expect that the selected features give good accuracy on various types of classifiers. These classifiers are usually trained using *n* training samples. Each training sample $\mathbf{x}_i; i = 1, 2, \ldots, n$, is a vector of *m* feature values of a known class. Given a testing sample $\mathbf{x}_j$ of an unknown class, the classifier finds the class of this sample. These three classifiers are described below.

***k*-Nearest Neighbor (*k*-NN):** This classical classifier classifies $\mathbf{x}_j$ by assigning it the class most frequently represented among the *k* nearest training samples [53]. Neighborhood is found based on a distance metric, e.g., Euclidian distance and city blocks distance.

**Linear Discriminant Analysis (LDA):** The LDA classifier is one of the earliest classifiers [54]. It learns a linear classification boundary for the training samples space. It can be used for both 2-class and multiclass problems. LDA fits a multivariate normal density to each class, with a pooled estimate of covariance.

**Support Vector Machine (SVM):** SVM is a newer classifier that uses kernels to construct linear classification boundaries in higher dimensional spaces [55]. SVM selects a small number of critical boundary samples from each class and builds a linear discriminant function.

# 5 Classification accuracy

To find how many features are needed to achieve good character recognition accuracy, we find the classification error as a function of the number of features used. In each experiment, we used best *m* features as selected by the five feature selection methods; for *m* = 4, 5, …, 96. The results are shown in Fig. 7. For every feature selection method, we evaluated the best *m* features using the 10-fold cross validation method on the *k*-NN, LDA, and SVM classifiers. The feature subsets used in the three NSGA curves come from respective optimizing experiments with NSGA/*k*-NN, NSGA/LDA, and NSGA/SVM. Note that the curves of the FCBF method stop at $m = 79$ because this method excludes features as discussed earlier.

For the three classifiers, the classification error decreases fast as the number of features increases from 4 to about 20 features. The LDA's classification error keeps decreasing slowly with more features. However, the *k*-NN's classification error increases when the number of features increases after reaching a minimum value in the region $m \in [13, 26]$ depending on the feature selection method used. SVM's classification error also increases with large *m* values, but stays with low values in a larger *m* region. For small *m* values, best classification accuracy is achieved by the SVM classifier. However, as SVM's classification error increases with large *m* values, the best accuracy is achieved by the LDA classifier for large *m* values.

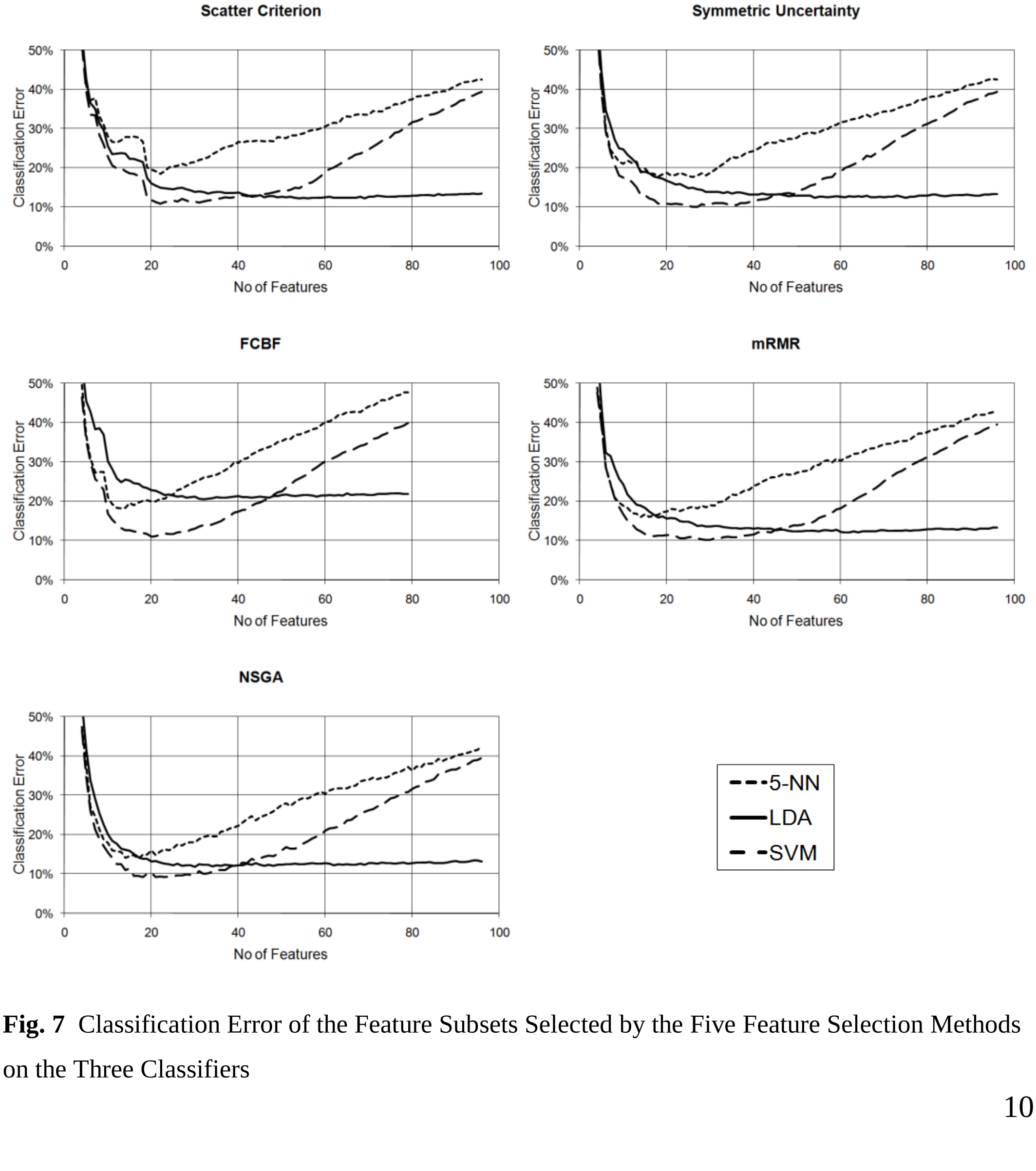

**Fig. 7** Classification Error of the Feature Subsets Selected by the Five Feature Selection Methods on the Three Classifiers

The SVM classifier achieves best classification accuracy for 20 features. And best SVM classifier's results are achieved using features selected by the NSGA/SVM method. Figure 8 gives clearer comparisons among the five feature selection methods on the three classifiers. This figure shows best results achieved for every feature selection method/classifier combination for $m \leq 20$ features. The NSGA/SVM and SVM combination achieves the lowest classification error of 9% at $m = 18$ features.

In general, best results are achieved with the features selected by the NSGA method followed by mRMR method. The FCBF and scatter criterion methods give unreliable results compared with the other three methods. The FCBF method selects features that give the worst classification error (23% with the LDA classifier). The scatter criterion method selects features that give the worst classification error when using the *k*-NN and SVM classifiers. Also note that the SVM classifier has best classification accuracy and the *k*-NN classifier has the worst.

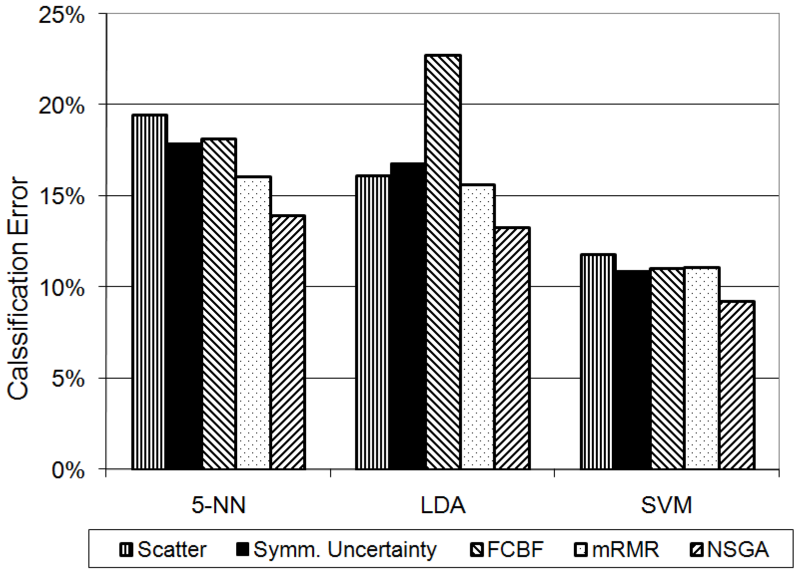

**Fig. 8** Classification Error Using 20 Best Features Selected by the Five Selection Methods on the Three Classifiers

# References

1. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(1):1157–1182
2. Dash M, Liu H (1997) Feature selection for classification. Intell Data Anal 1(3):131–156
3. Peng H, Long F, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27(8):1226–1238
4. Liu H, Yu L (2005) Toward integrating feature selection algorithms for classification and clustering. IEEE Trans Knowl Data Eng 17(4):491–502
5. Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. J Mach Lear Res 5(1):1205–1224
6. Oliveira L, Sabourin R, Bortolozzi F, Suen C (2003) A methodology for feature selection using multiobjective genetic algorithms for handwritten digit string recognition. Int J Pattern Recognit Artif Intell 17(6):903–929
7. Arica N, Yarman-Vural F (2002) Optical character recognition for cursive handwriting. IEEE Trans Pattern Anal Mach Intell 24(6):801–813
8. Lorigo L, Govindaraju V (2006) Offline Arabic handwriting recognition: a survey. IEEE Trans Pattern Anal Mach Intell 28(5):712–724
9. Pechwitz M, Snoussi Maddouri S, Märgner V, Ellouze N, Amiri H (2002) IFN/ENIT–database of handwritten Arabic words. Proc 7th Collque Int Francophone sur l’Ecrit et le Document, CIFED 2002, pp 129–136
10. Märgner V, Pechwitz M, ElAbed H (2005) ICDAR 2005 Arabic handwriting recognition competition. Proc Int Conf Doc Anal and Recognit, pp 70–74
11. Märgner V, El-Abed H (2007) ICDAR 2007Arabic handwriting recognition competition. Proc Int Conf Doc Anal and Recognit, pp 1274–1278
12. Jain A, Zongker D (1997) Feature selection: evaluation, application, and small sample performance. IEEE Trans Pattern Anal Mach Intell 19(2):153–158
13. Wei H-L, Billings S (2007) Feature subset selection and ranking for data dimensionality reduction. IEEE Trans Pattern Anal Mach Intell 29(1):162–166
14. Günter S, Bunke H (2004) Feature selection algorithms for the generation of multiple classifier systems and their application to handwritten word recognition. Pattern Recognit Lett 25(11):1323–1336
15. Oliveira L, Morita M, Sabourin R (2006) Feature selection for ensembles applied to handwriting recognition. Int J Doc Anal 8(4):262–279
16. Drauschke M, Förstner W (2008) Comparison of Adaboost and ADTboost for feature subset selection. Proc 8th Int Workshop on Pattern Recognit in Inf Syst, pp 113–122
17. Morita M, Sabourin R, Bortolozzi F, Suen C (2003) Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition. Proc 7th Int Conf on Doc Analy and Recognit, pp 666–670
18. Yang J, Honavar V (1998) Feature subset selection using a genetic algorithm. IEEE Intell Syst 13(1):44–49
19. Raymer M, Punch W, Goodman E, Kahn L, Jain L (2000) Dimensionality reduction using genetic algorithms. IEEE Trans Evol Comput 4(2):164–171
20. Emmanouilidis C, Hunter A, MacIntyre J (2000) A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. Proc Congress on Evol Comput, vol. 1, pp 309–316
21. Nouh A, Sultan A, Tolba R (1984) On feature extraction and selection for Arabic character recognition. Arab Gulf J Sci Res 2:329–347
22. Khedher M, Abandah G, Al-Khawaldeh A (2005) Optimizing feature selection for recognizing handwritten Arabic characters. Proc 2nd World Enformatika Congress, WEC’05, vol 1, pp 81–84
23. Pechwitz M, Maergner V, El Abed H (2006) Comparison of two different feature sets for offline recognition of handwritten Arabic words. Proc 10th Int Workshop on Front in Handwrit Recognit, pp 109–114
24. El Abed H, Margner V (2007) Comparison of different preprocessing and feature extraction methods for offline recognition of handwritten Arabic words. Proc 9th Int Conf on Doc Anal and Recognit, ICDAR 2007, pp 974–978

25. Abandah G, Younis K, Khedher M (2008) Handwritten Arabic character recognition using multiple classifiers based on letter form. Proc 5th IASTED Int Conf on Signal Process, Pattern Recognit, & Appl, SPPRA 2008, pp 128–133
26. Trier O, Jain A, Taxt T (1996) Feature extraction methods for character recognition: a survey. Pattern Recognit 29(4):641–662
27. Dalal S, Malik L (2008) A survey of methods and strategies for feature extraction in handwritten script identification. Proc 1st Int Conf on Emerging Trends in Eng and Technol, pp 1164–1169
28. Amin A, Al-Sadoun H, Fischer S (1996) Hand-printed Arabic character recognition system using an neural network. Pattern Recognit 29(4):663–675
29. Kavianifar M, AminA (1999) Preprocessing and structural feature extraction for a multi-fonts Arabic/Persian OCR. Proc 5th Int Conf on Doc Anal and Recognit, ICDAR'99, pp 213–216
30. El-Hajj R, Likforman-Sulem L, Mokbel C (2005) Arabic handwriting recognition using baseline dependant features and hidden Markov modeling. Proc Int Conf Doc Anal and Recognit, pp 893–897
31. Amin A (1997) Arabic character recognition. In: Bunke H, Wang P (ed) Handbook of character recognition and document image analysis. World Scientific, pp 397–420
32. Safabakhsh R, Adibi P (2005) Nastaaligh handwritten word recognition using a continuous-density variable-duration HMM. Arab J Sci Eng 30(1B):95–118
33. Sari T, Souici L, Sellami M (2002) Off-Line handwritten Arabic character segmentation algorithm: ACSA. Proc 8th Int Workshop Front in Handwrit Recognit, pp 452–457
34. Menasri F, Vincent N, Cheriet M, Augustin E (2007) Shape-based alphabet for off-line Arabic handwriting recognition. Proc 9th Int Conf on Doc Anal and Recognit, ICDAR 2007, vol 2, pp 969–973
35. AlKhateeb J, Ren J, Jiang J, Ipson S, El Abed H (2008) Word-based handwritten Arabic scripts recognition using DCT features and neural network classifier. Proc 5th Int Multi-Conf on Syst, Signals and Devices, pp 1–5
36. Theodoridis S, Koutroumbas K (2006) Pattern recognition, 3rd edn. Academic Press
37. Fisher F (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7:179–188
38. McLachlan G (1992) Discriminant analysis and statistical pattern recognition. Wiley Interscience, New York
39. Duda R, Hart P, Stork D (2001) Pattern classification, 2nd edn. Wiley-Interscience, New York
40. Srinivas N, Deb K (1995) Multi-objective function optimization using non-dominated sorting genetic algorithms. Evol Comput 2(3):221–248
41. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8(2):173–195
42. Gordon R, editor (2005) Ethnologue: languages of the world, 15th edn. SIL Int, Dallas
43. Abandah G, Khedher M (2008), Analysis of handwritten Arabic letters using selected feature extraction techniques. Int J Comput Process Lang 21(4), in press
44. Rosenfeld A, Kak A (1976) Digital picture processing. Academic Press, New York
45. Jain R, Kasturi R, Schunck B (1995) Machine vision. MacGraw-Hill, New York
46. Reiss T (1991) The revised fundamental theorem of moment invariants. IEEE Trans Pattern Anal Mach Intell 13(8):830-834
47. Deutsch E (1972) Thinning algorithms on rectangular, hexagonal, and triangular arrays. Comm of the ACM 15(9):827–837
48. Ha T, Bunke H (1997) Image processing methods for document image analysis. In: Bunke H, Wang P (ed) Handbook of character recognition and document image analysis. World Scientific, pp 1–47
49. Freeman H (1961) On the encoding of arbitrary geometric configurations. IRE Trans Electron Comput 10(2):260–268
50. Kuhl F, Giardina C (1982) Elliptic Fourier features of a closed contour. Comput Graphs Image Process 18(3):236–258
51. Mezghani N, Mitiche A, Cheriet M (2002) On-line recognition of hand-written Arabic characters using a Kohonen neural network. Proc 8th Int Workshop on Front in Handwrit Recognit, pp 490–495
52. Snoussi-Maddouri S, Amiri H, Belaid A, Choisy C (2002) Combination of local and global vision modeling for Arabic handwritten words recognition. Proc 8th Int Workshop on Front in Handwrit Recognit, pp 128–135
53. Mitchell T (1997) Machine learning. McGraw-Hill, New York
54. Webb A (2002) Statistical pattern recognition, 2nd edn. Wiley, New York

55. Burges C (1998) A tutorial on support vector machines for pattern recognition. Knowl Discov Data Min 2(2):1–43
56. Khedher M, Abandah G (2002) Arabic character recognition using approximate stroke sequence. Proc Workshop Arabic Lang Resources and Evaluation: Status and Prospects at 3rd Int Conf on Lang Resources and Evaluation, LREC 2002
57. Lorigo L, Govindaraju V (2005) Segmentation and pre-recognition of Arabic handwriting. Proc Int Conf Doc Anal and Recognit, ICDAR 2005, pp 605–609
58. Bentrcia R, Elnagar A (2008) Handwriting segmentation of Arabic text. Proc 5th IASTED Int Conf on Signal Process, Pattern Recognit & Appl, SPPRA 2008, pp 122–127
59. Deb K, Pratap A, Agrawal S, Meyarivan T (2002) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. IEEE Trans Evolut Comput 6(2):182–197
60. Kohavi R (1974) A study of cross-validation and bootstrap for accuracy estimation and model selection. Proc 14th Int Joint Conf on Artif Intell, pp 1137–1143
61. Stone M (1974) Cross-validatory choice and assessment of statistical predictions. J R Stat Soc 36(2):111–147
62. Fukunaga K (1990) Introduction to statistical pattern recognition. Academic Press Professional Inc., San Diego
63. Hsu C-W, Lin C-J (2002) A comparison of methods for multi-class support vector machines. IEEE Trans Neural Netw 13(2):415–425