Feature selection using feature dissimilarity measure and density-based clustering: Application to biological data

DEBARKA SENGUPTA[1], INDRANIL AICH[2] and SANGHAMITRA BANDYOPADHYAY[3,]*

[1]*Genome Institute of Singapore, Singapore 138 672, Singapore*

[2]*HTL Co. India Pvt. Ltd., New Delhi 110 092, India*

[3]*Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700 108, India*

**Corresponding author (Email, sanghami@isical.ac.in)*

Reduction of dimensionality has emerged as a routine process in modelling complex biological systems. A large number of feature selection techniques have been reported in the literature to improve model performance in terms of accuracy and speed. In the present article an unsupervised feature selection technique is proposed, using maximum information compression index as the dissimilarity measure and the well-known density-based cluster identification technique DBSCAN for identifying the largest natural group of dissimilar features. The algorithm is fast and less sensitive to the user-supplied parameters. Moreover, the method automatically determines the required number of features and identifies them. We used the proposed method for reducing dimensionality of a number of benchmark data sets of varying sizes. Its performance was also extensively compared with some other well-known feature selection methods.

[Sengupta D, Aich I and Bandyopadhyay S 2015 Feature selection using feature dissimilarity measure and density-based clustering: Application to biological data. *J. Biosci.* **40** 721–730] **DOI** 10.1007/s12038-015-9556-y

1. Introduction

Bioinformaticians frequently face the challenge of reducing the number of attributes of high-dimensional biological data for improving the models involved in sequence analysis, microarray analysis, spectral analysis, literature mining, etc. 21. Feature selection is useful for multiple reasons. The main objectives of feature selection are as follows: (a) accelerating the model creation task; (b) avoiding model over-fitting or under-fitting; (c) identifying the salient features, which are decisive of the target categories. Feature selection is widely used in classification, clustering, regression, etc. A typical feature selection process consists of four basic steps for finding the optimal set of features: subset generation, subset evaluation, stopping criterion and result validation 4. Feature selection methods can be categorized as either filter or wrapper 11. A third category called *hybrid* can be introduced to encompass the rest of the methods. Filter methods pick up relevant features by observing their intrinsic properties 21. These methods generally assign some score to each of the features while evaluating them in isolation. These methods scale up well due to their simplicity. The major disadvantage of such methods is that they ignore the relationship of features with the existing classes. GINI index, F-score, Relief-F 13 and Markov Blanket filter 14 are some popular filter methods. Unlike filter methods, wrapper methods learn from the natural grouping of data. These methods produce a subset of features that can efficiently differentiate between classes or clusters. Wrapper methods can be supervised as well as unsupervised in nature. Supervised wrapper approaches utilize class label information to evaluate feature sub sets. Such methods are often computationally expensive as they tend to do a rigorous search in the respective feature space. Genetic algorithm-based supervised feature selection approaches (Pal and Wang 1996; Tan et al. 2006; Mukhopadhyay

Keywords. Clustering; dissimilarity; eigenvalue; feature selection

Published online: 28 September 2015

et al. 2009) are popular as wrapper methods. Unsupervised wrapper methods are becoming increasingly popular because of their lower time requirement compared to the supervised ones. Principal Component Analysis (PCA) and MICI (maximum information compression index) (Mitra et al. 2002) are popular among these. An interesting hybrid filter-wrapper approach is introduced in Ruiz et al. (2006), crossing a univariately pre-ordered feature ranking with an incrementally augmenting wrapper method.

Biological data sets usually contain hundreds and thousands of features. For microarray data sets contain many thousands genes (about 5,000–30,000). These data sets are frequently use molecular classification of various life-threatening diseases like cancers (Golub et al. 1999). Reducing the dimensionality of biological data sets is essential to avoid model over-fitting. Filter methods are preferably use to reduce dimensionality of such data sets. Some frequently used filter methods are F-score, χ^2, Euclidean distance, *i*-test, Information Gain, etc. 21. Filter methods work well for simple cases, where distinction between different classes are quite obvious and apparent. However, in many complex cases these methods fail to give much insight into the molecular differentiation. PCA is a reasonably fast, unsupervised wrapper method, which is commonly used in such cases. However, PCA fails discriminate classes when classes overlap along the direction of maximum variance of the instances. In practice, feature extraction based on PCA often suffer from the problem of under-fitting. Unsupervised methods are indispensable when labeled instances are not available. For example, single cell RNA-sequence data analysis, which reveals tissue heterogeneity require single cells to be mapped to known cell types (Jaitin et al. 2014). Only filter or unsupervised wrapper techniques can be used to reduce dimensionality in such studies.

A disadvantage of the MICI (maximum information compression index) based approach lies with the use of *k*-NN-based clustering algorithm for finding clusters of features based on their similarity (Mitra et al. 2002). It is a common knowledge that any method based on *k*-NN principle is somewhat sensitive to the choice of *k*. Moreover, their approach of selecting a representative features from each of the clusters of similar features tends to discard important (dissimilar) features when the clusters are large in size. In contrast, if clusters of dissimilar features are identified, loss of important features can be minimized by selecting the largest among the clusters. Additionally, it is always preferred that the number of features is determined automatically by the feature selection technique itself. To address these issues we propose a feature selection method that works by discovering natural groups of dissimilar features. We show that the larger eigenvalue of the covariance matrix derived from a pair of features is inversely proportional to their dissimilarity, thereby making it an appropriate distance for obtaining group of dissimilar features. It is to be noted here that the present method is inspired by the work in (Mitra et al. 2002), with notable differences. Moreover a comparison with (Mitra et al. 2002) is also provided in the Results section.

An extensive comparison of the proposed method with several state-of-the-art techniques, viz. MICI (maximum information compression index) (Mitra et al. 2002), mRMR (Max-Dependency, Max-Relevance and Min-Redundancy) (Peng et al. 2005), SFFS, SFBS (Pudil et al. 1994), SBS, SFS, and Branch and Bound (Devijver and Kittler 1982), demonstrate its significance and effectiveness. For ease of reference the proposed feature selection technique is named as Feature Selection using Information Compression Index, or FSICI.

The rest of this paper is organized as follows: In section 2 we explain how the larger eigenvalue corresponding to the covariance matrix derived from a pair of features can be used to find cluster of dissimilar features. In this section we also describe various components as well as the computational complexity of the algorithm. In section 3 we compare the performance of FSICI with multiple established feature selection techniques. In this section we also provide statistical evidence for superior performance of FSICI. A brief analysis is also done to evaluate sensitivity of the approach to the required parameters. We conclude the paper in section 4.

2. Method

The proposed feature selection technique involves three steps: First, all dissimilarities between all possible pairs of features are measured using the principle of linear projections (described later in this section). Natural groups of dissimilar features are then identified using DBSCAN, the density based clustering method (Ester et al. 1996). Finally, the cluster containing the maximum number of features is selected. The reasons for using DBSCAN for clustering are following: (1) DBSCAN is capable of determining outlier; (2) it does not require the possible number of clusters as a input; and (3) it can discover arbitrary shaped clusters. The steps involved in the proposed feature selection method are illustrated below:

2.1 *Steps of FSICI*

2.1.1 *Measuring feature dissimilarity:* Although correlation coefficient is one of the obvious choices for tracking linear dependency between two variables x and y, it has the following shortcomings: the measure is invariant to scaling and translation of the variables and is sensitive to rotation of the scatter diagram in the (x,y) plane. Least square regression, on the other hand, is not symmetric and also sensitive to rotation. In (Mitra et al. 2002), the authors used the smaller eigenvalue of the covariance matrix of random variables x and y in order to quantify information loss between a pair of features. In the following derivation we show how the larger eigenvalue can be used as a measure of feature dissimilarity.

Let x and y be two random variables and the covariance matrix of x and y be Σ:

$$\left[\sum = \begin{pmatrix} var(x) & cov(x,y) \\ cov(x,y) & var(y) \end{pmatrix}.\right]$$

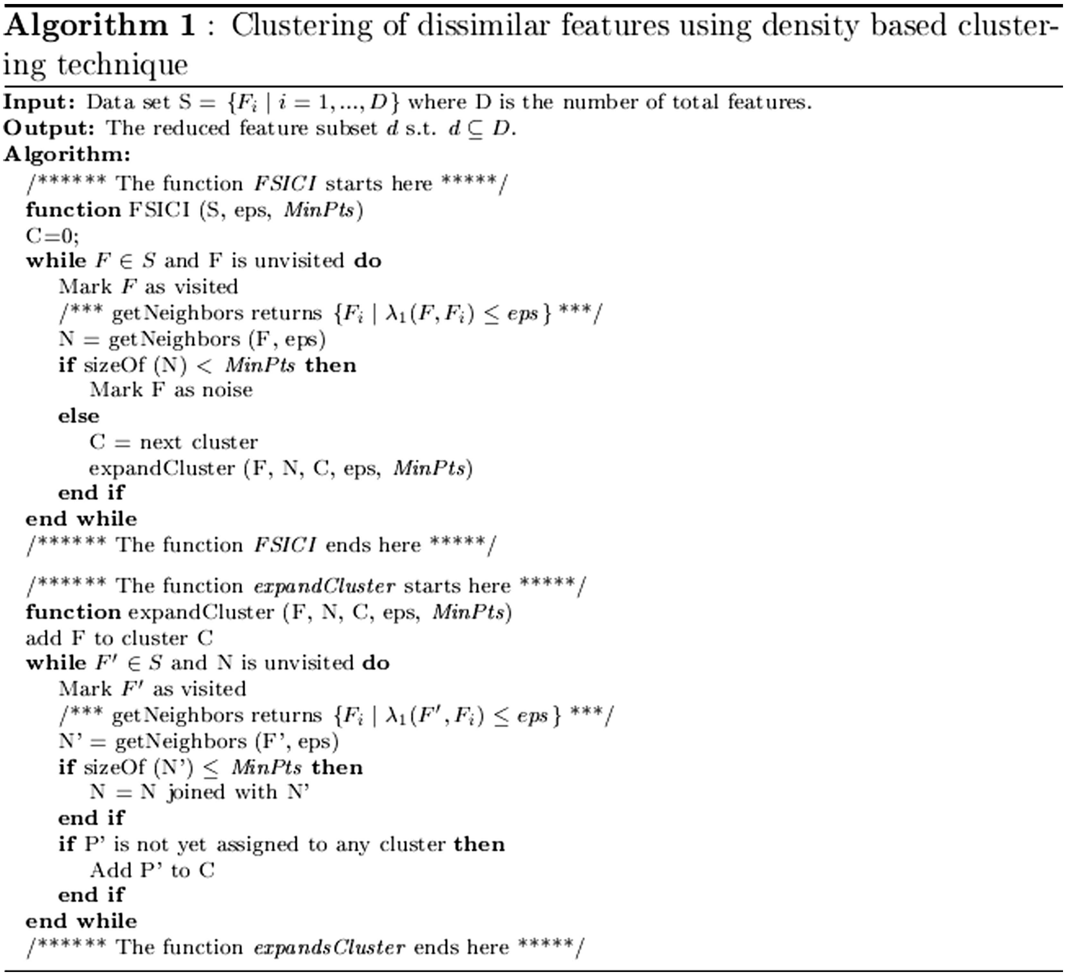
We have to find the eigenvalues λ_1 and λ_2 of the matrix.

$$\begin{aligned}
\left|\Sigma - \lambda \mathrm{I}\right| \quad &= \quad \begin{vmatrix} var(x) - \lambda & cov(x,y) \\ cov(x,y) & var(y) - \lambda \end{vmatrix} \quad = \quad 0 \\
&\Rightarrow \lambda^2 - \lambda(var(x) + var(y)) + var(x)var(y) - cov(x,y)^2 = 0 \\
&\Rightarrow \quad \lambda \quad = \quad \frac{1}{2}(\mathrm{var}(x) + \mathrm{var}(y)) \quad \pm \\
&\frac{1}{2}\sqrt{(var(x) + var(y))^2 - 4\left(var(x)var(y) - cov(x,y)^2\right)} \\
&\Rightarrow \quad 2 \quad \lambda \quad (x,y) \quad = \quad (var(x) + var(y)) \quad \pm \\
&\sqrt{(var(x) + var(y))^2 - 4var(x)var(y)\left(1 - \frac{cov(x,y)^2}{var(x)var(y)}\right)} \\
&\Rightarrow \quad 2 \quad \lambda \quad (x,y) \quad = \quad (var(x) + var(y)) \quad \pm \\
&\sqrt{(var(x) + var(y))^2 - 4var(x)var(y)\left(1 - \rho(x,y)^2\right)},
\end{aligned} \tag{1}$$

where I denotes an identity matrix and $\varrho(x,y)$ denotes the Pearson's correlation coefficient between x and y. For simplicity the smaller and larger eigenvalues are referred to as λ_2 and λ_1 respectively. It is apparent from Equation 1 that the value of λ_2 tends to 0 as the absolute value of $\varrho(x,y)$ increases. Note that λ_1 is directly proportional to the dependency between x and y, i.e. λ_1 increases as the amount of dependency increases. Hence, if we use λ_1 as the similarity measure, we obtain dissimilar features grouped while employing any clustering technique. Maximum information compression is achieved if the bivariate (or multivariate) data is projected along its principal component direction. The corresponding loss of information in reconstruction of the pattern is equal to the eigenvalue along the direction, normal to the principal component. Hence, higher value of λ_1 indicates smaller amount of information loss or higher amount of information compression or increased similarity. Therefore, clustering features using λ_1 as the distance measure would produce clusters of dissimilar features.

2.1.2 *Clustering of features:* Density-based clustering techniques identify regions of high density that are separated from one another by regions of low density. Density threshold is usually defined by the presence of a minimum number of data points (*MinPts* in Algorithm 1) within a specific radius (*Eps* in Algorithm 1). DBSCAN, the most popular density based technique discriminates the core, border and noise data points in terms of their surrounding density configuration (Ester et al. 1996). Core point is a point that satisfies the density requirement, border point is one that does not satisfy the density requirement but falls in the neighbourhood of a core point, and noise point is one which is neither a core point nor a border point. The points are connected based on density reachability. A point p is directly density-reachable from another point q if p belongs to the neighbourhood of q and q is a core point. Given *Eps* and *MinPts*, a point p is said to be density-reachable from q if there is a chain of points $p_1,....p_n$, $p_1{=}q$, $p_n{=}p$ s.t. p_{i+1} is directly density reachable from p_i. Note that *Eps*, in this case is nothing but a fixed upper limit of λ_1 (see the result tables). The pseudo-code of DBSCAN

(Ester et al. 1996) using λ_1 as similarity measure is shown in Algorithm 1.



Algorithm 1 : Clustering of dissimilar features using density based clustering technique

Input: Data set S = $\{F_i \mid i = 1, ..., D\}$ where D is the number of total features.
Output: The reduced feature subset d s.t. $d \subseteq D$.
Algorithm:

```
/****** The function FSICI starts here *****/
function FSICI (S, eps, MinPts)
C=0;
while $F \in S$ and F is unvisited do
    Mark $F$ as visited
    /*** getNeighbors returns $\{F_i \mid \lambda_1(F, F_i) \leq eps\}$ ***/
    N = getNeighbors (F, eps)
    if sizeOf (N) < MinPts then
        Mark F as noise
    else
        C = next cluster
        expandCluster (F, N, C, eps, MinPts)
    end if
end while
/****** The function FSICI ends here *****/

/****** The function expandCluster starts here *****/
function expandCluster (F, N, C, eps, MinPts)
add F to cluster C
while $F' \in S$ and N is unvisited do
    Mark $F'$ as visited
    /*** getNeighbors returns $\{F_i \mid \lambda_1(F', F_i) \leq eps\}$ ***/
    N' = getNeighbors (F', eps)
    if sizeOf (N') ≤ MinPts then
        N = N joined with N'
    end if
    if P' is not yet assigned to any cluster then
        Add P' to C
    end if
end while
/****** The function expandsCluster ends here *****/
```

2.1.3 *Selection of a set of important features:* The largest cluster obtained from the density-based clustering of the features is selected and the features included in the cluster are considered to be the reduced set of features. There could be alternative approaches for finding the most comprehensive cluster of features.

2.2 *Computational complexity of FSICI*

Most of the existing supervised or unsupervised feature selection indices like entropy, class seperability, k-NN classification accuracy have at least quadratic time complexity, whereas the proposed linear dependency measure has a complexity $O(l)$ (l is number of samples). On the other hand, the clustering algorithm DBSCAN has an overall runtime complexity $O(n\log n)$ (n is number of features). Note that the time complexity is mostly governed by the number of *getNeighbors()* calls, resulting in a complexity $O(l*n)$. DBSCAN executes exactly one such call for each feature, and an indexing structure is used that executes such a neighbourhood query in $O(\log n)$. Therefore, the overall run time complexity of FSICI is computed as $O(l.n.\log n)$. The computational complexity of MICI is $O(n^2 l)$. If the desired dimension is denoted by d, the complexity of SFFS and SFBS are in $O(d)$. However, this is massively increased at the time of supervision. Time complexity of Relief-F is $O(m.l.n)$, where m is a constant that determines number of iterations required to estimate weights of features.

Table 1. Experimental data sets

Data set	Dim.	Samples	Class	Ref.
Breast-cancer	9	683	2	UCI-MLR
Statlog (Heart)	13	270	2	UCI-MLR
Parkinson's	22	195	2	UCI-MLR
WDBC	30	569	2	UCI-MLR
Dermatology	34	358	6	UCI-MLR
Arrhythmia	259	452	13	UCI-MLR
Colon Cancer	2000	62	2	Alon *et al*. 1999
Lymphoma	4026	96	8	Alizadeh *et al*. 2000
Leukaemia	7129	72	2	Golub *et al*. 1999

UCI-MLR: The data sets are available in UCI Machine Learning Repository (Blake and Merz 1998).

3. Results

We used various benchmark data sets to evaluate the performance of the proposed algorithm and compare that with the performance of other existing techniques. Data sets with varying dimensions are selected for comprehensive analysis. In total 9 data sets and 8 existing algorithms were used for comparison purposes. Credibility of a set of selected features was determined by the classification accuracies obtained using that. Wilcoxon signed-ranks test was used to evaluate the statistical significance of the observed performance by our algorithm. Finally, we also examined dependency of the proposed method on the user definable parameters.

3.1 *Description of various benchmark data sets*

The real-life public domain data sets, which are used in our experiments can be categorized into three types: low-dimensional (*dim*<30), medium-dimensional (30≤*dim*<100) and high-dimensional (*dim*>100). The details related to the experimental data sets furnished in table 1. From table 1 it is clear that low-dimensional data sets include breast cancer, Statlog (heart) and Parkinson's data sets, medium-dimensional data sets include WDBC and dermatology data sets and high-dimensional data sets include arrhythmia, colon cancer data sets, LYMPHOMA and leukaemia data sets.

3.2 *Experimental set-up and brief analysis of results*

We demonstrates the effectiveness of our proposed feature selection technique on seven biological data sets of varying dimensionality (stated at the beginning of this section). The classification accuracies obtained using different classifiers trained with the FSICI suggested features are compared to those obtained using the features suggested by some well-known feature selection methods. Note that all accuracies reported in this study were obtained from 10-fold cross-validation. Seven popular feature selection techniques, namely, MICI (maximum information compression index) (Mitra et al. 2002), mRMR (Max-Dependency, Max-Relevance and Min-Redundancy) (Peng et al. 2005), SFFS, SFBS (Pudil et al. 1994), SBS, SFS, and Branch and Bound (Devijver and Kittler 1982) were considered in our experiments. The following brief descriptions about some used feature selection methods:

> **MICI:** Minimum Information compression index is an unsupervised wrapper method for feature selection that uses nearest-neighbour-based novel unsupervised clustering to find groups of similar features. Finally it uses some representative features obtained from each of the clusters.

Table 2. Classification results of the data sets without feature selection

	Classifiers							
	NB		*k*-NN		AdaBoost		SVM	
Full data set	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Breast-cancer	95.77	±1.27	95.64	±1.03	95.07	±0.88	94.27	±0.77
Statlog (Heart)	79.13	±3.35	74.19	±4.26	72.22	±6.41	54.86	±0.99
Parkinson's	70.51	±3.61	78.40	±4.56	78.67	±4.63	74.68	±0.38
WDBC	93.12	±0.75	92.83	±1.43	92.01	±2.15	62.72	±.70
Dermatology	86.52	±4.00	88.32	±3.45	45.93	±3.24	57.14	±5.17
Arrhythmia	54.98	±2.97	48.97	±3.71	49.29	±6.66	54.37	±0.73
Colon Cancer	56.07	±9.42	58.57	±8.41	48.75	±9.41	57.14	±10.3
Lymphoma	87.07	±2.51	96.57	±4.32	96.33	±4.22	97.61	±1.8
Leukemia	96.13	±1.18	82.7	±2.41	96.14	±4.23	97.82	±1.03

NB: Naive Bayes classifier, *k*-NN: k-Nearest Neighbour classifier.

mRMR: mRMR (Max-Dependency, Max-Relevance and Min-Redundancy) uses mutual information based method for selecting non redundant features.

Sequential methods: Sequential forward selection search (SFFS) starts with a null feature set and, for each step, the best feature that satisfies some criterion function is included with the current feature set. This is basically nothing but one step of the sequential forward selection (SFS). The algorithm also verifies the possibility of improvement of the criterion if some feature is excluded. In this case, the worst feature is eliminated from the set, that is, it is performed one step of sequential backward selection (SBS). Therefore, the SFFS proceeds dynamically increasing and decreasing the number of features until the desired *d* is reached.

Branch and Bound: Branch and bound is an exact method, which employs backtracking. This is an optimal method for monotonic feature set.

Relief-F: It is a supervised filter method that computes weight for each feature based on some probabilistic supervision.

Naive Bayes, *k*-NN (*k*=1), AdaBoost (number of iterations =10, weight threshold =100) and SVM (RBF kernel, cost =1 and gamma =0) are used for the classification tasks (Weka libraries are used with default parameters [Hall et al. 2009]). Classification accuracies are first measured without performing any feature selection. In the case of Parkinson's, WDBC, dermatology and colon cancer data, FSICI helps classifiers obtain the best accuracies, whereas for rest of

Table 3. Comparison results of different feature selection algorithms for small data sets

Data set	Evaluation criteria		Feature selection method								
			MICI	SFS	SFFS	SBS	SFBS	BB	Relief-F	mRMR	FSICI
Breast cancer	NB	Mean	95.11	95.51	95.20	95.46	94.15	95.87	95.06	96.00	95.90
		SD	±0.53	±1.32	±0.94	±1.17	±0.93	±0.48	±1.86	±0.526	±1
	k-NN	Mean	93.45	95.56	94.80	94.94	93.07	94.33	95.76	95.54	95.66
		SD	±1.37	±1.05	±0.97	±0.55	±2.21	±1.34	±0.666	±1.26	±0.884
D=9	AdaBoost	Mean	94.46	95.40	93.72	95.32	92.85	93.98	94.73	94.99	95.27
d=4		SD	±0.511	±1.75	±1.52	±0.557	±1.87	±2.03	±1.35	±1.44	±0.783
Minpts=2	SVM	Mean	94.67	**96.07**	95.45	95.77	94.26	95.76	94.62	95.30	**96.05**
Eps=1.97e-02		SD	±0.366	±0.873	±0.707	±0.398	±0.715	±0.643	±0.824	±0.82	±0.767
	CPU TIME	-	0.054	1.250	1.750	1.219	1.906	4.734	1.180	1.27	0.041
Statlog heart	NB	Mean	72.47	76.30	74.44	78.52	75.68	**80.58**	77.00	68.69	**79.38**
		SD	±2.7	±4.85	±2.56	±6.80	±2.36	±2.01	±2.02	±2.70	±2.27
	k-NN	Mean	69.59	73.62	73.00	77.08	70.62	73.70	74.20	62.23	76.05
		SD	±3.80	±2.28	±3.38	±3.83	±5.45	±4.07	±3.77	±5.90	±1.69
D=13	AdaBoost	Mean	69.38	74.44	74.77	76.30	74.28	75.93	74.57	66.23	76.71
d=5		SD	±5.13	±2.11	±2.21	±2.32	±6.01	±4.7	±1.9	±4.98	±2.78
Minpts=2	SVM	Mean	53.70	73.05	72.92	77.28	74.77	78.77	75.97	51.88	74.17
Eps=1.87e-02		SD	±3.92	±5.59	±3.56	±2.05	±4.13	±2.88	±1.9	±4.66	±2.29
	CPU TIME	-	0.031	1.625	2.500	5.707	6.202	7.782	5.920	1.450	0.020
Parkinson's	NB	Mean	73.20	73.37	71.31	72.23	69.26	65.66	73.43	71.77	76.60
		SD	±3.54	±2.12	±2.28	±3.84	±2.82	±4.79	±3.33	±3.87	±3.04
	k-NN	Mean	80.11	74.11	77.26	74.40	73.14	72.28	77.20	78.22	**82.69**
		SD	±4.57	±2.57	±4.36	±3.06	±2.07	±3.26	±2.62	±4.96	±4.19
D=22	AdaBoost	Mean	79.31	78.11	78.40	76.80	72.06	73.85	75.14	79.54	**80.23**
d=11		SD	±3.16	±4.59	±4.94	±2.04	±3.56	±7.19	±3.83	±5.17	±3.43
Minpts=2	SVM	Mean	74.91	76.51	76.57	75.09	74.34	74.57	73.94	74.17	74.40
Eps=5.01e-05		SD	±0.684	±4.33	±4.41	±0.552	±0.50	±0.774	±1.38	±1.66	±0.59
	CPU TIME	-	0.063	2.360	1.120	3.875	4.235	295.219	1.080	2.640	0.051

MICI: Feature selection using MICI, *SFS*: Sequential Forward Search, *SFFS*: Sequential Forward Floating Selection, *SBS*: Sequential Backward Search, *SFBS*: Sequential Forward Backward Selection, *BB*: Branch and Bound, NB: Naive Bayes classifier, mRMR: Max-Dependency, Max-Relevance and Min-Redundancy, *k*-NN: k-Nearest Neighbour classifier, D: Number of original features, d: Number of selected features using FSICI. The boldfaced values indicate the top two among the achieved average accuracies.

Table 4. Comparison results of different feature selection algorithms for medium data sets

Data set	Evaluation criteria		Feature selection method								
			MICI	SFS	SFFS	SBS	SFBS	BB	Relief-F	mRMR	FSICI
WDBC	NB	Mean	91.60	86.95	90.21	89.90	72.17	89.75	87.93	92.07	90.66
		SD	±0.719	±2.19	±1.36	±0.803	±2.64	±0.414	±1.08	±1.73	±1.76
	k-NN	Mean	90.45	86.82	87.68	87.52	77.32	87.81	89.28	90.64	**92.50**
		SD	±2.12	±2.35	±2.43	±2.45	±2.6	±2.44	±1.81	±1.81	±1.55
D=30	AdaBoost	Mean	90.57	89.77	90.33	88.73	72.91	89.28	89.65	**92.32**	92.17
d=17		SD	±1.80	±1.27	±2.42	±2.91	±4.80	±2.01	±1.66	±1.29	±1.90
Minpts=2	SVM	Mean	62.77	62.44	62.91	62.56	62.36	62.66	71.50	63.49	90.23
Eps=0.375e+01		SD	±0.926	±0.634	±0.654	±0.452	±0.808	±0.503	±1.74	±1.42	±0.939
	CPU TIME	-	0.219	3.125	2.047	6.172	10.375	18.922	7.660	1.120	0.129
Dermatology	NB	Mean	76.02	80.99	87.33	80.84	74.72	69.63	**88.60**	86.73	**88.79**
		SD	±5.00	±3.48	±3.76	±5.16	±5.21	±4.15	±3.53	±2.96	±2.2
	k-NN	Mean	75.25	82.30	86.09	81.93	72.39	65.45	87.64	86.89	79.66
		SD	±3.64	±2.66	±3.34	±2.35	±3.84	±3.09	±2.71	±3.12	±2.77
D=34	AdaBoost	Mean	46.43	45.12	46.46	45.68	41.02	37.42	45.96	44.53	48.51
d=17		SD	±3.10	±2.22	±3.76	±6.15	±10.3	±8.02	±4.69	±5.02	±2.44
Minpts=3	SVM	Mean	72.45	53.88	49.29	54.57	39.25	65.93	84.44	79.25	80.56
Eps=5.72e-03		SD	±7.48	±5.3	±7.58	±6.49	±9.63	±5.25	±2.79	±5.30	±6.33
	CPU TIME	-	0.156	9.710	16.656	37.500	94.641	281.235	12.050	2.99	0.114

the data sets, FSICI leads to at least the second best results. FSICI performs well consistently in conjunction with different classifiers, which is really promising. The 10-fold cross-validation accuracies are furnished in tables 2–5. For cancer and Statlog heart data, SFS and Branch and Bound (respectively) perform nominally better than FSICI. Across all the data sets it is evident that FSICI selected features produce either the best or the second best of all the reported accuracies. The other important observation is that for data sets of varying sizes, FSICI takes the least time for selecting the features. It is also observed that FSICI extract highest accuracies for the multi-class classification problems with respect to the arrhythmia and dermatology data.

We first tested the performance of the various classifiers on complete data sets, without employing any feature selection. The obtained classification accuracies are furnished in table 2. Among the small-dimensional data sets only for Parkinson’s, a negligible improvement was experienced while working with the unpruned data set. For rest of the small-dimensional data sets, all four classifiers performed strictly better over the feature sets reduced by FSICI. For the WBDC data set, SVM produced poor accuracies using the features suggested by all feature selection methods, whereas FSICI made SVM achieve 90% accuracy with unaltered parameters. For medium-dimensional dermatology data the classification accuracy reported by *k*-NN over the original data was 88%, whereas using FSICI an accuracy of only 80% was achieved. In contrast, Naive Bayes (89%) and SVM (80%) perform well with the FSICI-pruned data (accuracy received from the original data 87% and 57% respectively). Among the large-dimensional data sets, for arrhythmia, Naive Bayes and SVM brought slight improvement when executed on the original data set. FSICI-pruned feature set perform well in conjunction with *k*-NN and AdaBoost for the Arythmia data set. For large-dimensional cancer data FSICI-pruned data performed strictly well as supposed to the unpruned data sets, in conjunction with all the classifiers. Finally, it can be concluded that FSICI promises improved classification accuracies with solid consistency.

3.2.1 *Analysis of statistical significance of the results:* The McNemar test (Gillick and Cox 1989) on the cross-validation results summarized in tables 3–5 identify no significant superiority of any feature selection method on any of the used data sets. The accuracies are too close to each other. From tables 3–5 we see that FSICI sometime achieves the highest accuracy and sometime the second highest accuracy. The cross-validation accuracy of the classification models were not sufficient to conclude about the superiority FSICI. Therefore, we performed one-sided Wilcoxon signed-rank test (Kanji 1999) to test if the proposed methodology is superior. The test accounts for the ranked lists of the feature selection methods based on the cross-validation accuracies and examines the amount of deviation of the ranking from the null hypothesis H_0. For H_0 we assumed that all feature selection methods perform equally well. Table 6 reports the *p*-values corresponding to each pair of feature selection methods. The entry on *i*-th row and *j*-th column specifies the *p*-value related to the testing of the superiority of the *i*-th method over the *j*-th method. From tabular representation of *p*-values it is clear that the performance of (FSICI) is significantly better than the other methods. The Wilcoxon signed-ranks test is a non-parametric alternative to the paired *t*-test. It ranks the differences in performances of two concerned classifiers for each data set while ignoring the signs. After that it compares the ranks for the positive and the negative differences. Let d_i be the difference between the accuracy scores of the two classifiers on i^{th} out of N data sets. The differences are ranked according to their absolute values; average ranks are assigned in case of ties. Let R^+ be the

Table 5. Comparison results of different feature selection algorithms for large data sets

			Feature selection method					
Data set	Evaluation criteria		MICI	SFS	SFFS	Relief-F	mRMR	FSICI
Arrhythmia	NB	Mean	52.604	49.877	52.998	50.885	53.58	53.661
		SD	±4.73	±4.19	±5.23	±5.26	±3.66	±2.35
	k-NN	Mean	49.435	46.683	52.776	46.020	52.482	51.499
		SD	±4.23	±4.98	±4.13	±5.52	±3.72	±2.97
D=259	AdaBoost	Mean	50.000	53.342	**54.496**	**55.160**	54.422	**54.496**
d=98		SD	±5.64	±4.44	±4.48	±2.48	±3.41	±2.14
Minpts=5	SVM	Mean	54.054	53.096	53.784	54.275	54.103	53.243
Eps=30.97e-02		SD	±0.714	±0.915	±0.972	±0.425	±0.903	±0.705
	CPU TIME	-	6.703	101.406	120.453	68.521	308.301	3.537
Colon cancer	NB	Mean	59.107	60.357	60.893	61.429	62.857	**65.535**
		SD	±11.2	±6.78	±5.01	±10.7	±10.9	±2.24
	k-NN	Mean	60.357	61.250	61.429	60.893	**65.178**	64.821
		SD	±9.02	±6.94	±2.55	±7.78	±8.00	±4.30
D=2000	AdaBoost	Mean	55.893	55.000	54.821	58.214	59.107	61.250
d=471		SD	±8.03	±11.4	±9.11	±9.53	±10.90	±3.67
Minpts=3	SVM	Mean	56.607	61.786	62.500	56.250	60.714	63.928
Eps=17.027e-02		SD	±12.9	±6.08	±1.46	±10.9	±10.90	±2.03
	CPU TIME	-	238.562	580.344	579.906	366.170	561.390	227.906
Lymphoma	NB	Mean	86.36	85.51	84.71	84.53	85.2	86.12
		SD	±1.73	±2.14	±1.79	±2.07	±2.33	±2.55
	k-NN	Mean	93.15	95.1	92.71	**97.95**	95.06	**98.31**
		SD	±3.5	±1.48	±2.91	±4.02	±3.44	±4.19
D=4026	AdaBoost	Mean	95.4	96.27	89.15	92.2	93.17	96.82
d=189		SD	±5.12	±5.16	±4.03	±3.49	±4.12	±1.81
Minpts=2	SVM	Mean	95.2	94.7	95.16	94.92	92.99	95.11
Eps=72.8e-02		SD	±2.14	±1.79	±2.11	±7.21	±4.06	±3.72
	CPU TIME	-	1071.81	1502.31	1611.48	1281.08	1579.11	891.43
Leukaemia	NB	Mean	89.41	–	–	–	–	89.23
		SD	±1.46	–	–	–	–	±1.41
	k-NN	Mean	84.35	–	–	–	–	86.92
		SD	3.14	–	–	–	–	1.28
D=7129	AdaBoost	Mean	95.42	–	–	–	–	**98.34**
d=233		SD	4.01	–	–	–	–	3.81
Minpts=4	SVM	Mean	98.12	–	–	–	–	**98.4**
Eps=11.83e-01		SD	±1.75	–	–	–	–	±1.89
	CPU TIME	-	1862.31	–	–	–	–	1271.39

Table 6. Wilcoxon signed-rank test *p*-values

	SFS	SFFS	Relief-F	mRMR	FSICI
MICI	↓0.30772	↓0.28014	↓0.13362	↓0.18352	↓0
SFS	-	↓0.41794	↓0.06576	↓0.34212	↓0
SFFS	-	-	↓0.242	↓0.9162	↓0.00022
Relief-F	-	-	-	↑0.74896	↓0.00072
mRMR	-	-	-	-	↓0.00068

Up arrow indicates that the comparison of rank sums is favorable for the method along the corresponding row. Down arrow indicates the opposite.

sum of ranks for the data sets on which the second algorithm outperformed the first one, and R^- the sum of ranks for the converse. Ranks of d_i=0 are split evenly among the sums; if there is an odd number of them, one is ignored:

$$\begin{cases} R^+ \doteq \sum_{d_i>0} rank(d_i) + 0.5 \times rank(d_i) \\ R^- \doteq \sum_{d_i<0} rank(d_i) + 0.5 \times rank(d_i) \end{cases} \quad (2)$$

Now let *T* be the smaller of the sums, i.e. T=$min(R^+,R^-)$. For a large number of data sets, the following statistic *z* is approximately normally distributed:

$$z \doteq \frac{T - 0.25 \times N \times (N+1)}{\sqrt{\left(\frac{1}{24} \times N \times (N+1) \times (2N+1)\right)}} \quad (3)$$

3.3 *Selection of DBSCAN parameters*

From the earlier discussion it is apparent that the working of the DBSCAN is dependent on two parameters, namely *Eps* and *MinPts*. Selection of *Eps* is important because selection of larger *Eps* makes clusters grow larger when the *MinPts* is not grown proportionately. A high value for *MinPts* makes the density requirement stricter.

If density of the data points varies significantly from region to region, it can miss natural clusters in sparse areas while merging clusters in regions of higher density. Instead of setting a high *Eps* one should set a little higher *MinPts* so that the algorithm identifies truly dense cluster regions. In the present feature selection method, choice of *Eps* must commensurate with the range of the eigenvalues. Figure 1 shows the varying number of features in the largest cluster and the number of clusters over different *MinPts* selections for the WDBC data, which has a total of 30 features. User can simply vary the *Eps* value while sticking to a standard and strict *MinPts*, until a convincing number of features are found in the largest cluster. Notice in figure 1 that, for WBDC data, the *MinPts* selection merely had any impact on number of features selected, with varying *Eps*.

As per our formulation of dissimilarity, choice of *Eps* imposes a threshold for the larger eigenvalue, i.e. λ_1. It is important to observe if performance of the classifier is highly sensitive to the change of *Eps*. To test this, a data set was simulated containing 500 features, 300 instances and 3 classes. Each instance is sampled from a multivariate Gaussian distribution N(μ, Σ), where μ=[$E(f_1)$,$E(f_2)$,…,$E(f_{500})$] denotes the vector of means of 500 features and Σ denotes

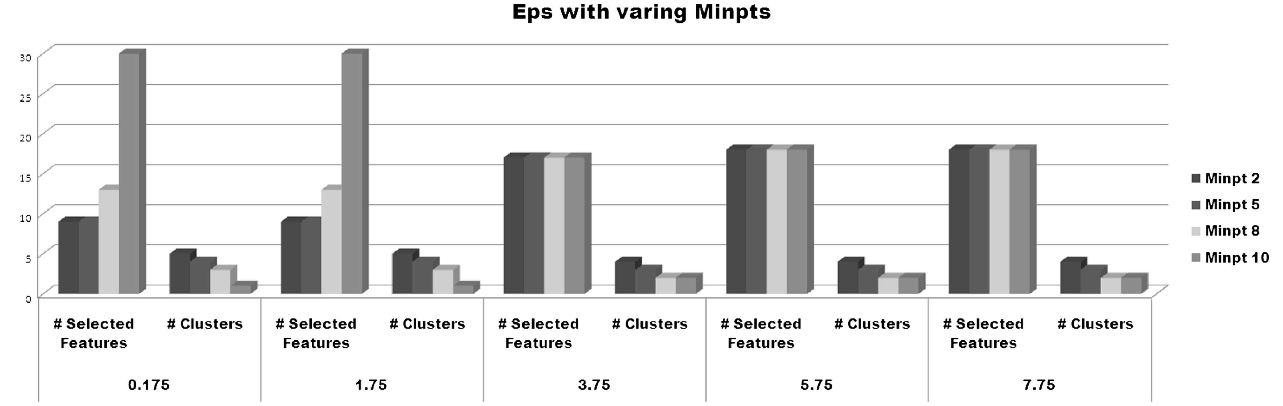


Figure 1. Number features varying with *Eps* and *MinPts* parameter selections, for the WDBC data.

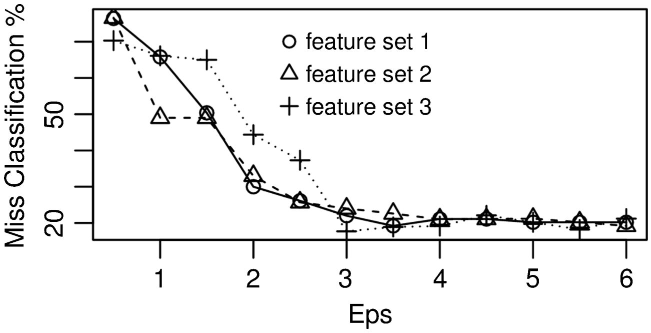


Figure 2. Classifier performance with varying *Eps.*

the specified covariance matrix. R packages *rockchalk* and *MASS* were used to accomplish this. Note that for different classes some controlled variations are introduced in the vector of means. The proposed method of feature selection was used before classification while varying the *Eps* 0.5 to 10.5. Naive Bayes is classifier is used for this purpose. The other parameter *MinPts* was kept fixed at 2. Also, every time the classification performance is tracked for top 3 clusters of features (figure 2). It was observed that performance of the classifier changes fairly smoothly with respect to *Eps* and settles down after a while. Also, it was found that use of different clusters perform equally well.

4. Conclusion

In this article we proposed a fast, unsupervised feature selection technique called FSICI based on the principles of information loss. The method is found performing significantly better than many existing feature selection techniques when applied on a wide range of biological data sets with diverse dimensions. One of the important features of the method is its low time complexity and manageable sensitivity to the parameters. Theoretically our work has some analogy with the method proposed by Mitra *et al.* (2002), but in practice we found that the two eigenvectors derived from covariance matrix of a pair of features have no consistent linear dependency.

References

Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, Boldrick JC, Sabet H, *et al.* 2000 Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature* **403** 503–511

Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D and Levine AJ 1999 Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci.* USA **96** 6745–6750

Blake C and Merz CJ 1998 {UCI} repository of machine learning databases

Devijver PA and Kittler J 1982 *Pattern recognition: a statistical approach* (Englewood Cliffs: Prentice/Hall International)

Ester M, Kriegel H-P, Sander J, and Xu X 1996 A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD* **96** 226–231

Gillick L and Cox SJ 1989 Some statistical issues in the comparison of speech recognition algorithms; in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, pp 532–535. IEEE

Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, *et al.* 1999 Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286** 531–537

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P and Witten IH 2009 The weka data mining software: an update. *ACM SIGKDD Explor. Newsl.* **11** 10–18

Jaitin DA, Kenigsberg E, Keren-Shaul H, Elefant N, Paul F, Zaretsky I, Mildner A, Cohen N, *et al.* 2014 Massively parallel single-cell rna-seq for marker-free decomposition of tissues into cell types. *Science* **343** 776–779

Kanji GK 1999 *100 statistical tests* (Sage)

Mitra P, Murthy C and Pal SK 2002 Unsupervised feature selection using feature similarity. *IEEE Trans. Pattern Anal. Mach. Intell.* **24** 301–312

Mukhopadhyay A, Maulik U and Bandyopadhyay S 2009 Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes. *IEEE Trans. Evol. Comput.* **13** 991–1005

Pal SK and Wang PP 1996 *Genetic algorithms for pattern recognition* (CRC press)

Peng H, Long F and 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** 1226–1238

Pudil P, Novovičová J and Kittler J 1994 Floating search methods in feature selection. *Pattern Recogn. Lett.* **15** 1119–1125

Ruiz R, Riquelme JC and Aguilar-Ruiz JS 2006 Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recogn.* **39** 2383–2392

Tan F, Fu X, Zhang Y, and Bourgeois AG 2006 Improving feature subset selection using a genetic algorithm for microarray gene expression data; In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on IEEE*. pp 2529–2534