Chapter 3

Unsupervised Feature Selection

3.1 Introduction

An important problem related to mining large data sets, both in dimension and size, is of selecting a subset of the original features [66]. Preprocessing the data to obtain a smaller set of representative features and retaining the optimal salient characteristics of the data not only decrease the processing time but also leads to more compactness of the models learned and better generalization. Dimensionality reduction can be done in two ways, namely, *feature selection* and *feature extraction*. As mentioned in Section 1.2.2, feature selection refers to reducing the dimensionality of the measurement space by discarding redundant or least information carrying features. One uses supervised feature selection when class labels of the data are available; otherwise unsupervised feature selection is appropriate. In many data mining applications class labels are unknown, thereby indicating the significance of unsupervised feature selection there. On the other hand, feature extraction methods utilize all the information contained in the measurement space to obtain a new transformed space, thereby mapping a higher dimensional pattern to a lower dimensional one.

In this chapter we describe an unsupervised feature selection algorithm based on measuring similarity between features and then removing the redundancy therein [166], for data mining applications. This does not need any search and, therefore, is fast. The method involves partitioning of the original feature set into some distinct subsets or clusters so that the features within a cluster are highly similar while those in different clusters are dissimilar. A single feature from each such cluster is then selected to constitute the resulting reduced subset. Before we describe the methodology and experimental results (Sections 3.4.2 and 3.6), we provide in Sections 3.2 and 3.3, in brief, different methods of feature extraction and feature selection for pattern recognition.

3.2 Feature Extraction

Feature extraction is a process of selecting a map of the form $X = f(Y)$, by which a sample $\mathbf{y}$ $(=[y_1, y_2, \ldots, y_p])$ in a p-dimensional measurement space Ω_Y is transformed into a point $\mathbf{x}$ $(=[x_1, x_2, \ldots, x_{p'}])$ in a p'-dimensional feature space Ω_X, where $p' < p$. Strategies involved in feature extraction include basic linear transformation of the input variables, e.g., principal component analysis (PCA), singular value decomposition (SVD), linear discriminant analysis (LDA), independent component analysis (ICA); more sophisticated linear transforms like spectral transforms (Fourier, Hadamard), wavelet transforms or convolution of kernels; and applying non-linear functions to subsets of variables, e.g., non-linear principal component analysis, Sammon's mapping and neural networks. Two distinct goals may be pursued for feature extraction: achieving the best reconstruction of the data or extracted features being the most efficient for making predictions. The first one is usually an unsupervised learning problem, while the second one is supervised.

The pioneering research on feature selection mostly deals with statistical tools. Later, the thrust of the research shifted to the development of various other approaches to feature selection, including fuzzy and neural approaches [200, 203, 243]. Principal component analysis [55] is the most well-known statistical method for feature extraction. It involves a linear orthogonal transform from a p-dimensional feature space to a p'-dimensional space, $p' \leq p$, such that the features in the new p'-dimensional space are uncorrelated and maximal amount of variance of the original data is preserved by only a small number of features.

Some of the recent attempts made for feature extraction are based on connectionist approaches using neural models like multilayer feedforward networks [20, 49, 50, 147, 154, 194, 229, 243, 247, 251] and self-organizing networks [125, 128, 154]. The methods based on multilayer feedforward networks include, among others, determination of saliency (usefulness) of input features [229, 243], development of Sammon's nonlinear discriminant analysis (NDA) network, and linear discriminant analysis (LDA) network [154]. On the other hand, those based on self-organizing networks include development of nonlinear projection (NP-SOM) based Kohonen's self-organizing feature map [154], distortion tolerant Gabor transformations followed by minimum distortion clustering by multilayer self-organizing maps [128], and a nonlinear projection method based on Kohonen's topology preserving maps [125].

Pal et al. [194] have proposed a neuro-fuzzy system for feature evaluation, both in supervised [49] and unsupervised [50] frameworks, along with its theoretical analysis. A fuzzy set theoretic feature evaluation index is defined in terms of individual class membership. Then a connectionist model, which incorporates weighted distance for computing class membership values, is used to perform the task of minimizing the fuzzy evaluation index. This

optimization process results in a set of weighting coefficients representing the importance of the individual features. These weighting coefficients lead to a transformation of the feature space for better modeling the class structures. The upper and lower bounds of the evaluation index, and its relation with interclass distance (e.g., Mahalanobis distance) and weighting coefficient were theoretically established.

The aforesaid neuro-fuzzy system has been extended to perform feature extraction in an unsupervised framework [50]. For this purpose, a set of different linear transformation functions is applied on the original feature space and the computation of the aforesaid evaluation index has been made on the transformed spaces. The similarity between two patterns in the transformed space is computed using a set of weighting coefficients. A layered network is designed where the transformation functions are embedded. An optimum transformed space along with the degrees of individual importance of the transformed (extracted) features are obtained through connectionist minimization. All these operations are performed in a single network where the number of nodes in its second hidden layer determines the desired number of extracted features.

Demartines et al. [52] have described a new strategy called "curvilinear component analysis (CCA)" for dimensionality reduction and representation of multidimensional data sets. The principle of CCA is implemented in a self-organized neural network performing two tasks: *vector quantization* of the submanifold in the data set (input space) and *nonlinear projection* of these quantized vectors toward an output space, providing a revealing unfolding of the submanifold. After learning, the network has the ability to continuously map any new point from one space into another.

The decision boundary feature extraction method, proposed by Lee et al. [131, 132], is based on the fact that all the necessary features for classification can be extracted from the decision boundary between a pair of pattern classes. The algorithm can take advantage of characteristics of neural networks which can solve complex problems with arbitrary decision boundaries without assuming the underlying probability distribution functions of the data.

Chatterjee et al. [38] have described various self-organized learning algorithms and associated neural networks to extract features that are effective for preserving *class separability*. An adaptive algorithm for the computation of $Q^{-1/2}$ (where Q is the correlation or covariance matrix of a random vector sequence) is described. Convergence of this algorithm with probability one is established by using stochastic approximation theory. A single layer linear network, called $Q^{-1/2}$ network, for this algorithm is described. Networks with different architectures are designed for extracting features for different cases.

Principal component analysis network of Rubner and Tavan [242] performs the task of feature extraction through the well-known principal component analysis. The network consists of two layers, viz., input and output. The weights of the network are adjusted through local learning rules.

Hornik et al. [97] have demonstrated the asymptotic behavior of a general class of on-line principal component analysis (PCA) learning networks which

are based strictly on local learning rules [242]. It is established that the behavior of the algorithms is intimately related to an ordinary differential equation which is obtained by suitable averaging over the training patterns. They have studied the equilibria of these equations and their local stability properties. It has been shown that local PCA algorithms should always incorporate hierarchical rather than more competitive, symmetric decorrelation, for providing their superior performance.

Recently, support vector machine (SVM) is also becoming popular for feature extraction in high dimensional spaces. In pattern recognition, SVM constructs nonlinear decision functions by training a classifier to perform a linear separation in some high dimensional space which is nonlinearly related to the input space. A Mercer kernel is used for mapping the input space to the high dimensional space [253]. The same type of kernel has been used to develop a nonlinear principal component analysis technique, namely, the Kernel PCA [160], which can efficiently extract polynomial features of arbitrary order by computing the projections onto principal components in the high dimensional space obtained by the kernels.

Many of the aforesaid feature extraction algorithms are, however, not suitable for data mining applications. Statistical methods like PCA fail for high dimensional data as they need to determine the eigenvalues of a large dimensional sparse matrix. Some of the connectionist approaches involve time-consuming learning iterations and require very high computational time for large data sets. Still so far, the literature on feature extraction algorithms, specifically suitable for data mining, is quite scarce.

3.3 Feature Selection

Conventional methods of feature selection involve evaluating different feature subsets using some index and selecting the best among them. The index usually measures the capability of the respective subsets in classification or clustering depending on whether the selection process is supervised or unsupervised. A problem of these methods, when applied to large data sets, is the high computational complexity involved in searching. The complexity is exponential in terms of the data dimension for an exhaustive search. Several heuristic techniques have been developed to circumvent this problem. Among them the branch and bound algorithm, suggested by Fukunaga and Narendra [55], obtains the optimal subset in expectedly less than exponential computations when the feature evaluation criterion used is monotonic in nature. Greedy algorithms like sequential forward and backward search [55] are also popular. These algorithms have quadratic complexity, but they perform poorly for non-monotonic indices. In such cases, sequential floating searches

[231] provide better results, though at the cost of a higher computational complexity. Beam search variants of the sequential algorithms [6] are also used to reduce computational complexity. Recently robust methods for finding the optimal subset for arbitrary evaluation indices are being developed using genetic algorithms (GAs) [211]. GA-based feature selection methods [126] are usually found to perform better than other heuristic search methods for large and medium sized data sets; however they also require considerable computation time for large data sets. Other attempts to decrease the computational time of feature selection include probabilistic search methods like random hill climbing [258] and Las Vegas Filter (LVF) approach [141]. Comparison and discussion of some of the above methods for many real life data sets may be found in [126].

Feature selection algorithms are sometimes denoted as either *filter* or *wrapper* based depending on the way of computing the feature evaluation indices. The algorithms which do not perform classification/clustering of the data in the process of feature evaluation constitute what is called the *filter* approach. In contrast to this, *wrapper* approach [117] directly uses the classification accuracy of some classifier as the evaluation criterion. The latter one often performs better than the filter approach, though much more time consuming. In the next two sections we briefly discuss some algorithms of filter and wrapper approaches.

3.3.1 Filter approach

We discuss here some of the filter methods for unsupervised feature selection. They can be broadly classified into two categories. Methods in one such category involve maximization of clustering performance, as quantified by some index. These include the sequential unsupervised feature selection algorithm [141], maximum entropy based method and the recently developed neuro-fuzzy approach [194]. The other category considers selection of features based on feature dependency and relevance. The principle is that any feature carrying little or no additional information beyond that subsumed by the remaining features is redundant and should be eliminated. Various dependence measures like correlation coefficients [87], measures of statistical redundancy [96], or linear dependence [47] have been used. Recently the Relief algorithm [113] and its extensions [124] which identify statistically relevant features have been reported. A fast feature selection algorithm based on an information fuzzy network is described in [129]. Another algorithm based on conditional independence uses the concept of Markov blanket [121]. All these methods involve search and require significantly high computation time for large data sets. In [112] an algorithm which does not involve search and selects features by hierarchically merging similar feature pairs is described. However, the algorithm is crude in nature and performs poorly on real life data sets. It may be noted that principal component analysis (PCA) [55] also performs unsupervised dimensionality reduction based on information content

of features. However, PCA involves feature transformation and obtains a set of transformed features rather than a subset of the original features.

3.3.2 Wrapper approach

In its most general formulation, the wrapper methodology consists of using the prediction performance of a given learning machine to assess the relative usefulness of different subsets of variables. In practice, one needs to define: (i) how to search the space of all possible variable subsets; (ii) how to assess the prediction performance of a learning machine to guide the search and halt it; and (iii) which predictor to use. A wide range of search strategies can be used, including breadth-first, branch-and-bound, simulated annealing and genetic algorithms [117]. Performance assessments are usually done using a validation set or by cross-validation methods such as leave-one-out and hold out. Popular predictors include decision trees, naive Bayes, least square linear predictors and support vector machines.

Wrappers are often criticized because they seem to be a "brute force" method requiring massive amounts of computation. Efficient search strategies may be devised to circumvent this. Using such strategies does not necessarily mean sacrificing prediction performance. In fact, it appears to be the converse in some cases; e.g., coarse search strategies may alleviate the problem of overfitting and increase the accuracy. Since wrappers use the learning machine as a black box, they are remarkably universal and simple. An efficient but less universal version of the wrapper methods is the *embedded* technique, which performs variable selection in the process of training, but it is dependent on the learning machines used.

3.4 Feature Selection Using Feature Similarity (FSFS)

Here we describe an unsupervised algorithm, FSFS [166], belonging to the filter approach. The method uses feature dependency/similarity for redundancy reduction but requires no search. It involves partitioning of the original feature set into some distinct subsets or clusters so that the features within a cluster are highly similar while those in different clusters are dissimilar. A single feature from each such cluster is then selected to constitute the resulting reduced subset. A novel similarity measure, called maximal information compression index, is used in clustering. Its comparison with two other measures namely, correlation coefficient and least square regression error, is made. It is also explained how 'representation entropy' can be used for quantifying redundancy in a set.

The nature of both the feature clustering algorithm and the feature simi-

larity measure is geared towards two goals – minimizing the information loss (in terms of second order statistics) incurred in the process of feature reduction and minimizing the redundancy present in the reduced feature subset. The feature selection algorithm owes its low computational complexity to two factors – (a) unlike most conventional algorithms, search for the best subset (requiring multiple evaluation of indices) is not involved, (b) the feature similarity measure can be computed in much less time compared to many indices used in other supervised and unsupervised feature selection methods. Since the method achieves dimensionality reduction through removal of redundant features, it is more related to feature selection for compression rather than for classification.

Superiority of the algorithm, over four related methods, viz., branch and bound algorithm, sequential floating forward search, sequential forward search and stepwise clustering, is demonstrated extensively on nine real life data of both large and small sample sizes and dimension ranging from 4 to 649. Comparison is made on the basis of both clustering/classification performance and redundancy reduction. Effectiveness of the maximal information compression index and the effect of scale parameter are also studied.

In Section 3.4.1 we describe measures of similarity between a pair of features. Section 3.4.2 describes the feature selection algorithm using the similarity measure. Some feature evaluation indices are presented in Section 3.5. In Section 3.6 we provide experimental results along with comparisons.

3.4.1 Feature similarity measures

In this section we discuss some criteria for measuring similarity between two random variables, based on linear dependency between them. In this context we present a novel measure called *maximal information compression index* to be used for feature selection.

There are broadly two possible approaches to measure similarity between two random variables. One is to non-parametrically test the closeness of probability distributions of the variables. Walds-Wolfowitz test and the other run tests [236] may be used for this purpose. However, these tests are sensitive to both location and dispersion of the distributions, hence not suited for the purpose of feature selection. Another approach is to measure the amount of functional (linear or higher) dependency between the variables. There are several benefits of choosing linear dependency as a feature similarity measure. It is known that if some of the features are linearly dependent on the others, and if the data is linearly separable in the original representation, the data is still linearly separable if all but one of the linearly dependent features are removed [47]. As far as the information content of the variables is concerned, second order statistics of the data is often the most important criterion after mean values [236]. All the linear dependency measures that we will discuss are related to the amount of error in terms of second order statistics, in predicting one of the variables using the other. We discuss below two existing [236] linear

dependency measures before explaining the *maximal information compression index*.

3.4.1.1 Correlation coefficient (ρ)

The most well-known measure of similarity between two random variables is the correlation coefficient. Correlation coefficient ρ between two random variables x_1 and x_2 is defined as $\rho(x_1, x_2) = \frac{\text{cov}(x_1,x_2)}{\sqrt{\text{var}(x_1)\text{var}(x_2)}}$, where var( ) denotes the variance of a variable and cov( ) the covariance between two variables. If x_1 and x_2 are completely correlated, i.e., exact linear dependency exists, $\rho(x_1, x_2)$ is 1 or -1. If x_1 and x_2 are totally uncorrelated, $\rho(x_1, x_2)$ is 0. Hence, $1 - |\rho(x_1, x_2)|$ can be used as a measure of similarity between two variables x_1 and x_2. The following can be stated about the measure:

1. $0 \leq 1 - |\rho(x_1, x_2)| \leq 1$.
2. $1 - |\rho(x_1, x_2)| = 0$ if and only if x_1 and x_2 are linearly related.
3. $1 - |\rho(x_1, x_2)| = 1 - |\rho(x_2, x_1)|$ (symmetric).
4. If $u = \frac{x_1 - a}{c}$ and $v = \frac{x_2 - b}{d}$ for some constants a, b, c, d, then $1 - |\rho(x_1, x_2)| = 1 - |\rho(u, v)|$ i.e., the measure is *invariant to scaling and translation* of the variables.
5. The measure is *sensitive to rotation* of the scatter diagram in (x_1, x_2) plane.

Although the correlation coefficient contains many desirable properties as a feature similarity measure, properties 4 and 5, mentioned above, make it somewhat unsuitable for feature selection. Since the measure is invariant to scaling, two pairs of variables having different variances may have the same value of the similarity measure, which is not desirable as variance has high information content. Sensitivity to rotation is also not desirable in many applications.

3.4.1.2 Least square regression error (e)

Another measure of the degree of linear dependency between two variables x_1 and x_2 is the error in predicting x_2 from the linear model $x_2 = a + bx_1$. a and b are the regression coefficients obtained by minimizing the mean square error $e(x_1, x_2)^2 = \frac{1}{n}\sum(e_i(x_1, x_2))^2$, $e_i(x_1, x_2) = x_{2i} - a - bx_{1i}$. The coefficients are given by $a = \bar{x_2}$ and $b = \frac{\text{COV}(x_1,x_2)}{\text{var}(x_1)}$ and the mean square error $e(x_1, x_2)$ is given by $e(x_1, x_2) = \text{var}(x_2)(1 - \rho(x_1, x_2)^2)$. If x_2 and x_1 are linearly related $e(x_1, x_2) = 0$, and if x_1 and x_2 are completely uncorrelated $e(x_1, x_2) = \text{var}(x_2)$. The measure e^2 is also known as the *residual variance*. It is the amount of variance of x_2 unexplained by the linear model. Some properties of e are:

1. $0 \leq e(x_1, x_2) \leq \text{var}(x_2)$.
2. $e(x_1, x_2) = 0$ if and only if x_1 and x_2 are linearly related.
3. $e(x_1, x_2) \neq e(x_2, x_1)$ (unsymmetric).
4. If $u = x_1/c$ and $v = x_2/d$ for some constant a, b, c, d, then $e(x_1, x_2) = d^2 e(u, v)$, i.e., the measure e is *sensitive to scaling* of the variables. It is also clear that e is *invariant to translation* of the variables.
5. The measure e is *sensitive to rotation* of the scatter diagram in $x_1 - x_2$ plane.

Note that the measure e is not symmetric (property 3). Moreover, it is sensitive to rotation (property 5).

Now we present a measure of linear dependency which has many desirable properties for feature selection not present in the above two measures.

3.4.1.3 Maximal information compression index (λ_2)

Let Σ be the covariance matrix of random variables x_1 and x_2. Define, *maximal information compression index* as $\lambda_2(x_1, x_2) =$ smallest eigenvalue of Σ, i.e.,

$$2\lambda_2(x_1, x_2) = (\text{var}(x_1) + \text{var}(x_2) - \sqrt{(\text{var}(x_1) + \text{var}(x_2))^2 - 4\text{var}(x_1)\text{var}(x_2)(1 - \rho(x_1, x_2)^2)}.$$

The value of λ_2 is zero when the features are linearly dependent and increases as the amount of dependency decreases. It may be noted that the measure λ_2 is nothing but the eigenvalue for the direction normal to the principle component direction of feature pair (x_1, x_2). It is shown in [55] that maximum information compression is achieved if multivariate (in this case bivariate) data are projected along its principal component direction. The corresponding loss of information in reconstruction of the pattern (in terms of second order statistics) is equal to the eigenvalue along the direction normal to the principal component. Hence, λ_2 is the amount of reconstruction error committed if the data is projected to a reduced (in this case reduced from two to one) dimension in the best possible way. Therefore, it is a measure of the *minimum amount of information loss* or the *maximum amount of information compression* possible.

The significance of λ_2 can also be explained geometrically in terms of linear regression. It can be easily shown [236] that the value of λ_2 is equal to the sum of the squares of the perpendicular distances of the points (x_1, x_2) to the best fit line $x_2 = \hat{a} + \hat{b}x_1$, obtained by minimizing the sum of the squared perpendicular distances. The coefficients of such a best fit line are given by $\hat{a} = \bar{x_1} cot\theta + \bar{x_2}$ and $\hat{b} = -cot\theta$, where $\theta = 2 \tan^{-1}\left(\frac{2\text{cov}(x_1,x_2)}{\text{var}(x_1)^2 - \text{var}(x_2)^2}\right)$. The nature of errors and the best fit lines for least square regression and

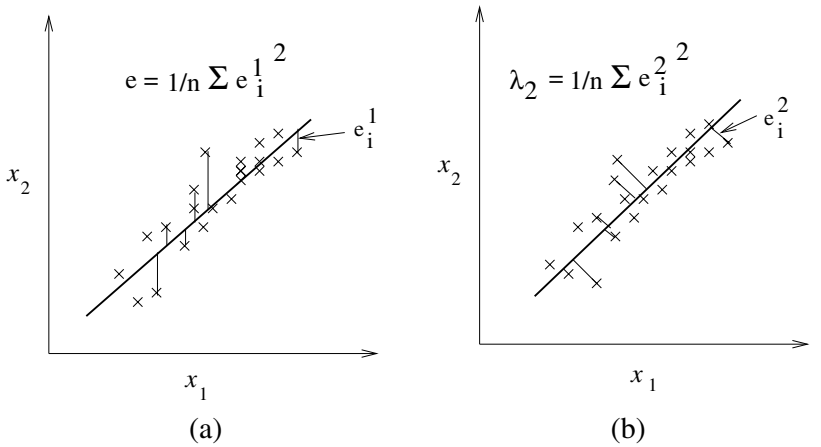


FIGURE 3.1: Nature of errors in linear regression, (a) Least square fit (e), (b) Least square projection fit (λ_2).

principal component analysis are illustrated in Figure 3.1. λ_2 has the following properties:

1. $0 \leq \lambda_2(x_1, x_2) \leq 0.5(\text{var}(x_1) + \text{var}(x_2))$.
2. $\lambda_2(x_1, x_2) = 0$ if and only if x_1 and x_2 are linearly related.
3. $\lambda_2(x_1, x_2) = \lambda_2(x_2, x_1)$ (symmetric).
4. If $u = \frac{x_1}{c}$ and $v = \frac{x_2}{d}$ for some constant a, b, c, d, then $\lambda_2(x_1, x_2) \neq \lambda_2(u, v)$; i.e., the measure is *sensitive to scaling* of the variables. Since the expression of λ_2 does not contain mean, but only the variance and covariance terms, it is *invariant to translation* of the data set.
5. λ_2 is *invariant to rotation* of the variables about the origin (this can be easily verified from the geometric interpretation of λ_2 considering the property that the perpendicular distance of a point to a line does not change with rotation of the axes).

The measure λ_2 possesses several desirable properties such as symmetry (property 3), sensitivity to scaling (property 4), and invariance to rotation (property 5). It is a property of the variable pair (x_1, x_2) reflecting the amount of error committed if maximal information compression is performed by reducing the variable pair to a single variable. Hence, it may be suitably used in redundancy reduction.

3.4.2 Feature selection through clustering

The task of feature selection involves two steps, namely, partitioning the original feature set into a number of homogeneous subsets (clusters) and se-

lecting a representative feature from each such cluster. Partitioning of the features is done based on the k-NN principle using one of the feature similarity measures described in Section 3.4.1. In doing so, the k nearest features of each feature are computed first. Among them the feature having the most compact subset (as determined by its distance to the farthest neighbor) is selected, and its k neighboring features are discarded. The process is repeated for the remaining features until all of them are either selected or discarded.

While determining the k nearest neighbors of features a constant error threshold (ϵ) is assigned; ϵ is set equal to the distance of the kth nearest neighbor of the feature selected in the first iteration. In subsequent iterations, it is checked whether the λ_2 value, corresponding to the subset of a feature, is greater than ϵ or not. If yes, the value of k is decreased. Therefore k may be varying over iterations. The concept of clustering features into homogeneous groups of varying sizes is illustrated in Figure 3.2. The algorithm may be stated as follows:

Algorithm:

Let the original number of features be P, and the original feature set be $A = \{F_i, i = 1, \ldots, P\}$. Represent the dissimilarity between features F_i and F_j by $S(F_i, F_j)$. The higher the value of S is, the more dissimilar are the features. The measures of linear dependency (e.g., ρ, e, λ_2) described in Section 3.4.1 may be used in computing S. Let r_i^k represent the dissimilarity between feature F_i and its kth nearest neighbor feature in R. Then

Step 1: Choose an initial value of $k \leq P - 1$. Initialize the reduced feature subset R to the original feature set A; i.e., $R \leftarrow A$.

Step 2: For each feature $F_i \in R$, compute r_i^k.

Step 3: Find feature $F_{i'}$ for which $r_{i'}^k$ is minimum. *Retain* this feature in R and *discard* k nearest features of $F_{i'}$. (Note: $F_{i'}$ denotes the feature for which removing k nearest neighbors will cause minimum error among all the features in R.) **Let** $\epsilon = r_{i'}^k$.

Step 4: **If** $k >$ cardinality$(R) - 1$: $k =$ cardinality$(R) - 1$.

Step 5: **If** $k = 1$: **Go to Step 8**.

Step 6: **While** $r_{i'}^k > \epsilon$ **do**:

(a) $k = k - 1$.

$r_{i'}^k = \inf_{F_i \in R} r_i^k$.

('k' is decremented by 1, until the 'kth nearest neighbor' of at least one of the features in R is less than ϵ-dissimilar with the feature)

(b) **If** $k = 1$: **Go to Step 8**.

(if no feature in R has less than ϵ-dissimilar 'nearest neighbor' select all the remaining features in R)

End While

Step 7: **Go to Step 2**.

Step 8: Return feature set R as the reduced feature set. □

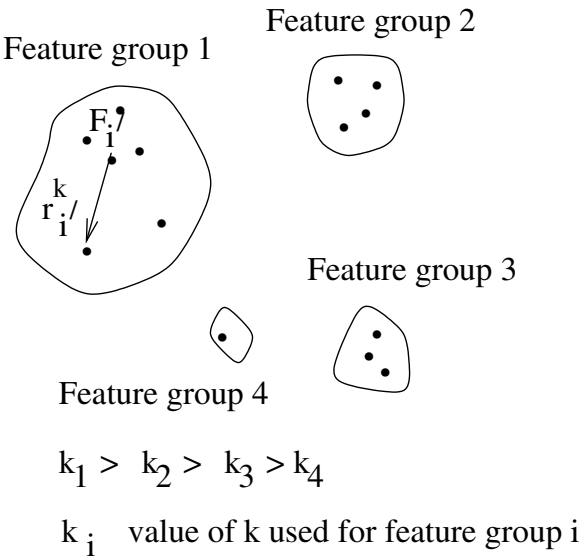


FIGURE 3.2: Feature clusters.

Remarks:

Computational complexity: The algorithm has low computational complexity with respect to both number of features and number of samples of the original data. With respect to the dimension (P) the method has complexity $\mathcal{O}(P^2)$. Among the existing search-based schemes only sequential forward and backward search have complexity $\mathcal{O}(P^2)$, though each evaluation is more time consuming. Other algorithms such as plus-l-take-r, sequential floating search and branch and bound algorithm [55] have complexity higher than quadratic. Most probabilistic search algorithms also require more than quadratic number of evaluations.

The second factor that contributes to the speed-up achieved by the similarity based algorithm is the low computational complexity of evaluating the linear dependency measures of feature similarity. If the data set contains n samples, evaluation of the similarity measure for a feature pair is of complexity $\mathcal{O}(n)$. Thus the feature selection scheme has overall complexity $\mathcal{O}(P^2n)$. Almost all other supervised and unsupervised feature evaluation indices (e.g., entropy, class separability, K-NN classification accuracy) have at least $\mathcal{O}(n^2)$ complexity of computation. Moreover, evaluation of the linear dependency measures involves computation using one-dimensional variables only, while the other measures often involve distance computations at higher dimensions. All these factors contribute to the large speed-up achieved by the similarity-based algorithm compared to other feature selection schemes.

Notion of scale in feature selection and choice of k: In similarity-based feature selection algorithm k controls the size of the reduced set. Since k determines

the error threshold (ϵ), the representation of the data at different degrees of details is controlled by its choice. This characteristic is useful in data mining where *multiscale* representation of the data is often necessary. Note that the said property may not always be possessed by other algorithms where the input is usually the desired size of the reduced feature set. The reason is that changing the size of the reduced set may not necessarily result in any change in the levels of details. In contrast, for the similarity-based algorithm, k acts as a scale parameter that controls the degree of details in a more direct manner.

Non-metric nature of similarity measure: The similarity measures used in the feature selection algorithm need not be a metric. Unlike conventional agglomerative clustering algorithms it does not utilize the metric property of the similarity measures. Also unlike the stepwise clustering method [112] used previously for feature selection, the clustering algorithm described in this section is partitional and non-hierarchical in nature.

3.5 Feature Evaluation Indices

Let us now describe some indices that may be considered for evaluating the effectiveness of the selected feature subsets. The first three indices, namely, class separability, K-NN classification accuracy and naive Bayes classification accuracy, do need class information of the samples while the remaining three, namely, entropy, fuzzy feature evaluation index and representation entropy, do not. Before we discuss them, we mention, for convenience, the following notations: Let n be the number of sample points in the data set, M be the number of classes present in the data set, P be the number of features in the original feature set A, p be the number of features in the reduced feature set R, Ω_A be the original feature space with dimension P, and Ω_R be the transformed feature space with dimension p.

3.5.1 Supervised indices

1. *Class separability* [55]: Class separability S of a data set is defined as $S = trace(S_b^{-1}S_w)$. S_w is the within-class scatter matrix and S_b is the between-class scatter matrix, defined as:

$$S_w = \sum_{j=1}^{M} \pi_j E\{(\mathbf{x} - \mu_j)(\mathbf{x} - \mu_j)^T | \omega_j\} = \sum_{j=1}^{M} \pi_j \Sigma_j$$

$$S_b = \sum_{j=1}^{M} (\mu_j - \bar{\mathbf{x}})(\mu_j - \bar{\mathbf{x}})^T$$

$$\bar{\mathbf{x}} = E\{\mathbf{x}\} = \sum_{j=1}^{M} \pi_j \mu_j \tag{3.1}$$

where π_j is the a priori probability that a pattern belongs to class ω_j, $\mathbf{x}$ is the feature vector, μ_j is the sample mean vector of class ω_j, $\bar{\mathbf{x}}$ is the sample mean vector for the entire data points, Σ_j is the sample covariance matrix of class ω_j, and $E\{.\}$ is the expectation operator. A lower value of the separability criteria S ensures that the classes are well separated by their scatter means.

2. *K-NN classification accuracy*: Here the K-NN rule is used for evaluating the effectiveness of the reduced set for classification. Cross-validation is performed in the following manner – randomly select 90% of the data as training set and classify the remaining 10% points. Ten such independent runs are performed, and the average classification accuracy on test set is used. The value of K, chosen for the K-NN rule, is the square root of the number of data points in the training set.

3. *Naive Bayes classification accuracy*: A Bayes maximum likelihood classifier [55], assuming normal distribution of classes, is also used for evaluating the classification performance. Mean and covariance of the classes are estimated from a randomly selected 10% training sample, and the remaining 90% of the points are used as a test set. Ten such independent runs are performed and the average classification accuracy on the test set is provided.

3.5.2 Unsupervised indices

1. *Entropy*: Let the distance between two data points i, j be

$$\mathcal{D}_{ij} = \left[\sum_{l=1}^{p} \left(\frac{x_{i,l} - x_{j,l}}{max_l - min_l} \right)^2 \right]^{1/2},$$

where $x_{i,l}$ denotes feature value for i along lth direction, and max_l, min_l are the maximum and minimum values computed over all the samples along lth axis, and p is the number of features. Similarity between i, j is given by $\mathrm{sim}(i,j) = e^{-\alpha \mathcal{D}_{ij}}$, where α is a positive constant. A possible value of α is $\frac{-\ln 0.5}{\bar{\mathcal{D}}}$. $\bar{\mathcal{D}}$ is the average distance between data points computed over the entire data set. Entropy is defined as [141]:

$$E = -\sum_{i=1}^{n} \sum_{j=1}^{n} (\mathrm{sim}(i,j) \times \log \mathrm{sim}(i,j) + (1 - \mathrm{sim}(i,j)) \times \log (1 - \mathrm{sim}(i,j))) \tag{3.2}$$

where n is the number of sample points. If the data are uniformly distributed in the feature space entropy is maximum. When the data has well-formed clusters uncertainty is low and so is entropy.

2. *Fuzzy feature evaluation index*: Fuzzy feature evaluation index (FFEI) is defined as [194]:

$$FFEI = \frac{2}{n(n-1)} \sum_{i} \sum_{i \neq j} \frac{1}{2} \left[\mu_{ij}^{R}(1 - \mu_{ij}^{A}) + \mu_{ij}^{A}(1 - \mu_{ij}^{R}) \right] \tag{3.3}$$

where μ_{ij}^{A} and μ_{ij}^{R} are the degrees that both patterns i and j belong to the same cluster in the feature spaces Ω_A and Ω_R, respectively. Membership function μ_{ij} may be defined as

$$\begin{aligned} \mu_{ij} &= 1 - \tfrac{d_{ij}}{\mathcal{D}_{max}} && \text{if } d_{ij} \leq \mathcal{D}_{max} \\ &= 0, && \text{otherwise.} \end{aligned}$$

d_{ij} is the distance between patterns i and j, and $\mathcal{D}_{max}$ is the maximum separation between patterns in the respective feature spaces.

The value of FFEI decreases as the intercluster/intracluster distances increase/ decrease. Hence, the lower the value of FFEI, the more crisp is the cluster structure.

Note that the first two indices, class separability and K-NN accuracy, measure the effectiveness of the feature subsets for classification, while the indices entropy and fuzzy feature evaluation index evaluate the clustering performance of the feature subsets. Let us now describe a quantitative index which measures the amount of redundancy present in the reduced subset.

3.5.3 Representation entropy

Let the eigenvalues of the $p \times p$ covariance matrix of a feature set of size p be $\lambda_l, l = 1, \ldots, p$. Let $\tilde{\lambda}_l = \frac{\lambda_l}{\sum_{l=1}^{p} \lambda_l}$. $\tilde{\lambda}_l$ has similar properties like probability, namely, $0 \leq \tilde{\lambda}_l \leq 1$ and $\sum_{l=1}^{p} \lambda_l = 1$. Hence, an entropy function can be defined as

$$H_R = -\sum_{l=1}^{p} \tilde{\lambda}_l \log \tilde{\lambda}_l. \tag{3.4}$$

The function H_R attains a minimum value (zero) when all the eigenvalues except one are zero, or in other words when all the information is present along a single co-ordinate direction. If all the eigenvalues are equal, i.e., information is equally distributed among all the features, H_R is maximum and so is the uncertainty involved in feature reduction.

The above measure is known as *representation entropy*. It is a property of the *data set as represented by a particular set of features* and is a measure of the amount of information compression possible by dimensionality reduction. This is equivalent to the amount of redundancy present in that particular representation of the data set. Since the feature similarity based algorithm involves partitioning of the original feature set into a number of homogeneous (highly compressible) clusters, it is expected that representation entropy of the individual clusters are as low as possible, while that of the final reduced set of features has low redundancy, i.e., a high value of representation entropy.

It may be noted that among all the p dimensional subspaces of an original P dimensional data set, the one corresponding to the Karhunen-Loeve coordinates [55] (for the first p eigenvalues) has the highest representation entropy, i.e., is least redundant. However, for large dimensional data sets K-L transform directions are difficult to compute. Also, K-L transform results in general transformed variables and not exact subsets of the original features.

3.6 Experimental Results and Comparisons

Organization of the experimental results is as follows [166]: First the performance of the similarity-based feature selection algorithm (FSFS) in terms of the feature evaluation indices, presented in Section 3.5, is compared with five other feature selection schemes. Then the redundancy reduction aspect of the algorithm is quantitatively discussed along with comparisons. Effect of varying the parameter k, used in feature clustering, is also shown.

Three categories of real life public domain data sets are considered: low dimensional ($P \leq 10$) (e.g., Iris, Wisconsin cancer, and Forest cover type (considering numerical features only) data), medium dimensional ($10 < P \leq 100$) (e.g., Ionosphere, Waveform and Spambase data), and high dimensional ($P > 100$) (e.g., Arrhythmia, Multiple features and Isolet data), containing both large and relatively smaller number of points. Their characteristics are described in Appendix B.

3.6.1 Comparison: Classification and clustering performance

Four indices, viz., entropy (Equation 3.2), fuzzy feature evaluation index (Equation 3.3), class separability (Equation 3.1), K-NN and naive Bayes classification accuracy are considered to demonstrate the efficacy of the FSFS algorithm and for comparing it with other methods. Four unsupervised feature selection schemes considered for comparison are:

1. Branch and Bound Algorithm (BB) [55]: A search method in which all possible subsets are implicitly inspected without exhaustive search.

If the feature selection criterion is monotonic BB returns the optimal subset.

2. Sequential Forward Search (SFS) [55]: A suboptimal search procedure where one feature at a time is added to the current feature set. At each stage, the feature to be included in the feature set is selected from among the remaining available features so that the new enlarged feature set yields a maximum value of the criterion function used.

3. Sequential Floating Forward Search (SFFS) [231]: A near-optimal search procedure with lower computational cost than BB. It performs sequential forward search with provision for backtracking.

4. Stepwise Clustering (using correlation coefficient) (SWC) [112]: A non-search-based scheme which obtains a reduced subset by discarding correlated features.

In the experiments, entropy (Equation 3.2) is mainly used as the feature selection criterion with the first three search algorithms.

Comparisons in terms of five indices are reported for different sizes of the reduced feature subsets. Tables 3.1, 3.2 and 3.3 provide such a comparative result corresponding to high, medium and low dimensional data sets when the size of the reduced feature subset is taken to be about half of the original size as an example. Comparison for other sizes of the reduced feature set is provided in Figure 3.3 considering one data set from each of the three categories, namely, multiple features (high), ionosphere (medium) and cancer (low). The CPU time required by each of the algorithms on a Sun UltraSparc 350 MHz workstation are also reported in Tables 3.1–3.3. Since the branch and bound (BB) and the sequential floating forward search (SFFS) algorithms require infeasibly high computation time for the large data sets, the figures for them could not be provided in Table 3.1. For the classification accuracies (using K-NN and Bayes), both mean and standard deviations (SD) computed for ten independent runs are presented.

Compared to the search-based algorithms (BB, SFFS and SFS), the performance of the feature similarity-based (FSFS) scheme is comparable or slightly superior, while the computational time requirement is much less for the FSFS scheme. On the other hand, compared to the similarity-based SWC method the performance of the FSFS algorithm is much superior, keeping the time requirement comparable. It is further to be noted that the superiority in terms of computational time increases as the dimensionality and sample size increase. For example, in the case of low dimensional data sets, the speed-up factor of the FSFS scheme compared to BB and SFFS algorithms is about 30−50, for Forest data which is low dimensional but has large sample size the factor is about 100, for medium dimensional data sets, BB and SFFS are about 100 times slower and SFS about 10 times slower, while for the high dimensional data sets SFS is about 100 times slower, and BB and SFFS could not be compared as they require infeasibly high run time.

TABLE 3.1: Comparison of feature selection algorithms for large dimensional data sets

Data set	Method	Evaluation Criteria					CPU Time (sec)
		E	FFEI	S	KNNA (%) Mean SD	BayesA (%) Mean SD	
Isolet	SFS	0.52	0.41	1.09	95.02 0.89	92.03 0.52	14.01×10^4
p=310	SWC	0.71	0.55	2.70	72.01 0.71	68.01 0.44	431
P=617	Relief-F	0.70	0.52	2.24	95.81 0.81	95.52 0.47	5.03×10^3
$k = 305$	FSFS	0.50	0.40	1.07	96.00 0.78	95.01 0.52	440
Mult. Feat.	SFS	0.67	0.47	0.45	77.01 0.24	75.02 0.14	5.00×10^4
p=325	SWC	0.79	0.55	0.59	52.00 0.19	50.05 0.10	401
P=649	Relief-F	0.71	0.50	0.52	78.37 0.22	75.25 0.11	1.10×10^3
$k = 322$	FSFS	0.68	0.48	0.45	78.34 0.22	75.28 0.10	451
Arrhythmia	SFS	0.74	0.44	0.25	52.02 0.55	50.21 0.43	1511
p=100	SWC	0.82	0.59	0.41	40.01 0.52	38.45 0.38	70
P=195	Relief-F	0.78	0.55	0.27	56.04 0.54	54.55 0.40	404
$k = 95$	FSFS	0.72	0.40	0.17	58.93 0.54	56.00 0.41	74

E: Entropy, FFEI: Fuzzy Feature Evaluation Index, S: Class Separability, KNNA: k-NN classification accuracy, BayesA: naive Bayes classification accuracy, SD: standard deviation. SFS: Sequential Forward Search, SWC: Stepwise Clustering, FSFS: Feature selection using feature similarity. p: number of selected features, P: number of original features, k: parameter used by the similarity-based method.

TABLE 3.2: Comparison of feature selection algorithms for medium dimensional data sets

Data set	Method	Evaluation Criteria					CPU Time (sec)
		E	FFEI	S	KNNA (%) Mean SD	BayesA (%) Mean SD	
Spambase	BB	0.50	0.30	0.28	90.01 0.71	88.17 0.55	1579
	SFFS	0.50	0.30	0.28	90.01 0.72	88.17 0.55	1109
	SFS	0.52	0.34	0.29	87.03 0.68	86.20 0.54	121.36
p=29	SWC	0.59	0.37	0.41	82.04 0.68	79.10 0.55	11.02
P=57	Relief-F	0.59	0.36	0.34	87.04 0.70	86.01 0.52	70.80
$k = 27$	FSFS	0.50	0.30	0.28	90.01 0.71	88.19 0.52	13.36
Waveform	BB	0.67	0.47	0.29	78.02 0.47	62.27 0.41	1019
	SFFS	0.68	0.48	0.31	77.55 0.45	62.22 0.41	627
	SFS	0.69	0.49	0.37	74.37 0.44	59.01 0.42	71.53
p=20	SWC	0.72	0.55	0.41	62.03 0.40	47.50 0.40	8.01
P=40	Relief-F	0.73	0.54	0.38	74.88 0.41	62.88 0.40	50.22
$k = 17$	FSFS	0.68	0.48	0.30	75.20 0.43	63.01 0.40	8.28
Ionosphere	BB	0.65	0.44	0.07	75.96 0.35	65.10 0.28	150.11
	SFFS	0.65	0.44	0.08	74.73 0.37	65.08 0.31	50.36
	SFS	0.65	0.44	0.10	69.94 0.32	62.00 0.27	10.70
p=16	SWC	0.66	0.47	0.22	62.03 0.32	59.02 0.25	1.04
P=32	Relief-F	0.62	0.47	0.15	72.90 0.34	64.55 0.27	8.20
$k = 11$	FSFS	0.64	0.43	0.10	78.77 0.35	65.92 0.28	1.07

BB: Branch and Bound, SFFS: Sequential Floating Forward Search

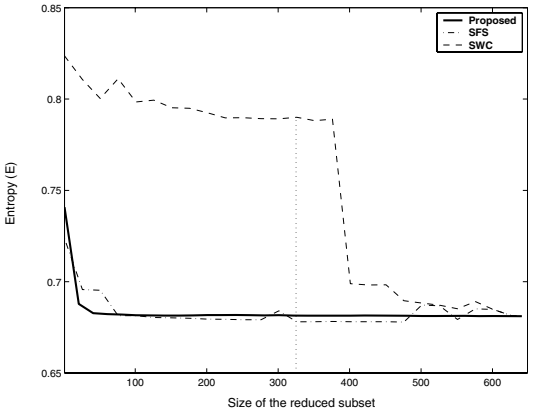
TABLE 3.3: Comparison of feature selection algorithms for low dimensional data sets

Data set	Method	Evaluation Criteria					CPU Time (sec)
		E	FFEI	S	KNNA (%) Mean SD	BayesA (%) Mean SD	
Forest	BB	0.65	0.40	0.90	64.03 0.41	63.55 0.40	4.01×10^4
	SFFS	0.64	0.39	0.81	67.75 0.43	66.22 0.41	3.02×10^4
	SFS	0.64	0.41	0.98	62.03 0.41	61.09 0.40	7.00×10^3
p=5	SWC	0.68	0.45	1.00	54.70 0.37	53.25 0.35	50.03
P=10	Relief-F	0.65	0.40	0.90	64.03 0.41	63.55 0.40	2.80×10^4
$k = 5$	FSFS	0.65	0.40	0.90	64.03 0.41	63.55 0.40	55.50
Cancer	BB	0.59	0.36	1.84	94.90 0.17	94.45 0.14	3.39
	SFFS	0.59	0.36	1.84	94.90 0.17	94.45 0.14	6.82
	SFS	0.61	0.37	2.68	92.20 0.17	91.05 0.15	1.16
p=4	SWC	0.60	0.37	2.69	90.01 0.19	89.11 0.17	0.10
P=9	Relief-F	0.59	0.36	1.84	94.90 0.17	94.25 0.17	0.91
$k = 5$	FSFS	0.56	0.34	1.70	95.56 0.17	94.88 0.17	0.10
Iris	BB	0.55	0.34	22.0	96.80 0.14	97.33 0.10	0.56
	SFFS	0.55	0.34	22.0	96.80 0.14	97.33 0.10	0.71
	SFS	0.57	0.35	27.0	92.55 0.17	93.10 0.14	0.25
p=2	SWC	0.60	0.37	29.2	92.19 0.19	93.02 0.17	0.01
P=4	Relief-F	0.55	0.34	22.0	96.80 0.14	97.33 0.10	0.14
$k = 2$	FSFS	0.55	0.34	22.0	96.80 0.14	97.33 0.10	0.01

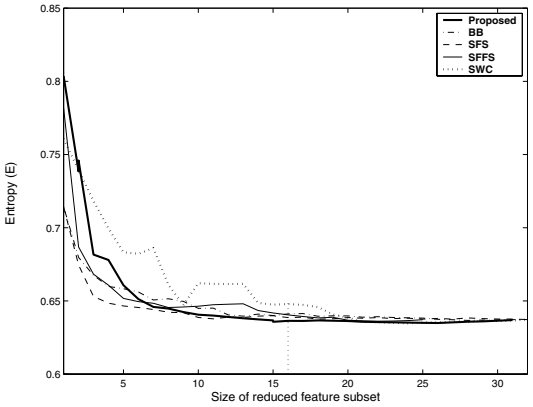
It may be noted that the aforesaid unsupervised feature selection algorithms (viz., BB, SFFS, SFS) usually consider 'entropy' as the selection criterion. Keeping this in mind detailed results are provided in Tables 3.1−3.3. However, some results using another unsupervised measure, namely, fuzzy feature evaluation index (FFEI) (Equation 3.3) are also depicted in Table 3.4. These are shown, as an illustration, only for the four large data sets (Isolet, Multiple features, Arrhythmia and Forest cover type). These results corroborate the findings obtained using entropy.

For comparing the performance with that of a supervised method, Relief-F, which is widely used, 50% of the samples were used as design set. Results are presented in Tables 3.1−3.3. The Relief-F algorithm provides classification performance comparable to the similarity-based scheme in spite of using class label information. Moreover, it has a much higher time requirement, especially for data sets with large number of samples, e.g., the Forest data. Its performance in terms of the unsupervised indices is also poorer.

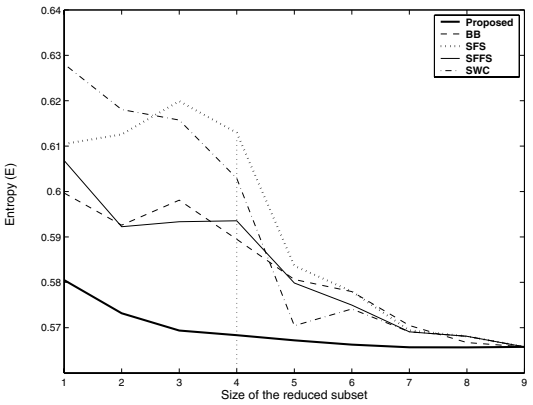
Statistical significance of the classification performance of the similarity-based method compared to those of the other algorithms is tested. Means and SD values of the accuracies, computed over 10 independent runs, are used for this purpose. The test statistics described in Section 2.6.2 is used. It is observed that the FSFS method has significantly better performance compared to the SWC algorithm for all the data sets, and the SFS algorithm for most of the data sets. For the other algorithms, namely, Relief-F, BB and SFFS, the performance is comparable; i.e., the difference of the mean values of the classification scores is statistically insignificant.



(a)



(b)



(c)

FIGURE 3.3: Variation in classification accuracy with size of the reduced subset for (a) Multiple features, (b) Ionosphere, and (c) Cancer data sets. The vertical dotted line marks the point for which results are reported in Tables 3.1−3.3.

TABLE 3.4: Comparison of feature selection algorithms for large data sets when search algorithms use FFEI as the selection criterion

Data set	Method	Evaluation Criteria					CPU Time (sec)
		FFEI	E	S	KNNA (%) Mean SD	BayesA (%) Mean SD	
Isolet	SFS	0.40	0.54	0.98	95.81 0.82	92.19 0.72	28.01×10^4
p=310, P=617	FSFS	0.40	0.50	1.07	96.00 0.78	95.01 0.52	440
Mult. Feat.	SFS	0.44	0.67	0.44	77.71 0.44	75.81 0.17	9.20×10^4
p=325, P=649	FSFS	0.48	0.68	0.45	78.34 0.22	75.28 0.10	451
Arrhythmia	SFS	0.40	0.77	0.21	53.22 0.59	52.25 0.44	2008
p=100, P=195	FSFS	0.40	0.72	0.17	58.93 0.54	56.00 0.41	74
Forest	BB	0.40	0.65	0.90	64.03 0.41	63.55 0.40	9.21×10^4
	SFFS	0.40	0.66	0.83	67.01 0.45	66.00 0.44	7.52×10^4
	SFS	0.43	0.66	1.01	61.41 0.44	60.01 0.41	17.19×10^3
p=5, P=10	FSFS	0.40	0.65	0.90	64.03 0.41	63.55 0.40	55.50

TABLE 3.5: Representation entropy H_R^s of subsets selected using some algorithms

Data set	BB	SFFS	SFS	SWC	Relief-F	FSFS
Isolet	-	-	2.91	2.87	2.89	3.50
Mult. Ftrs.	-	-	2.02	1.90	1.92	3.41
Arrhythmia	-	-	2.11	2.05	2.02	3.77
Spambase	2.02	1.90	1.70	1.44	1.72	2.71
Waveform	1.04	1.02	0.98	0.81	0.92	1.21
Ionosphere	1.71	1.71	1.70	0.91	1.52	1.81
Forest	0.91	0.82	0.82	0.77	0.91	0.91
Cancer	0.71	0.71	0.55	0.55	0.59	0.82
Iris	0.47	0.47	0.41	0.31	0.47	0.47

3.6.2 Redundancy reduction: Quantitative study

As mentioned before, the FSFS algorithm involves partitioning the original feature set into a certain number of homogeneous groups and then replacing each group by a single feature, thereby resulting in the reduced feature set. Representation entropy (H_R), defined in Section 3.5, is used to measure the redundancy in both the homogeneous clusters and the final selected feature subset. H_R when computed over the individual clusters should be as low as possible (indicating high redundancy among the features belonging to a single cluster), while giving as high value as possible for the selected subset (indicating minimum redundancy). Let us denote the average value of H_R computed over the homogeneous groups by H_R^g and the value of H_R for the final selected subset by H_R^s.

Table 3.5 shows the comparative results of the FSFS method with other feature selection algorithms in terms of H_R^s. It is seen that the subset obtained by the FSFS scheme is least redundant having the highest H_R^s values.

To demonstrate the superiority of the *maximal information compression index* λ_2, compared to the other two feature similarity measures (ρ and e) used

TABLE 3.6: Redundancy reduction using different feature similarity measures

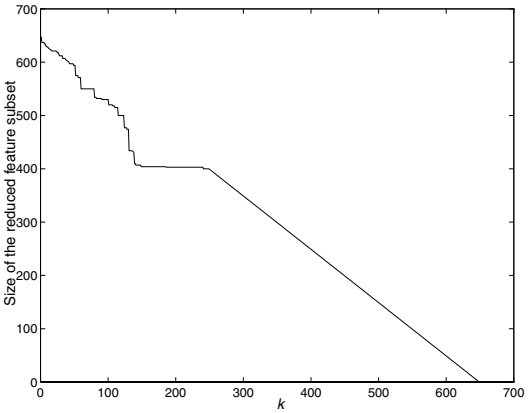
Data set	Similarity Measure: λ_2		Similarity Measure: e		Similarity Measure: ρ	
	H_R^g	H_R^s	H_R^g	H_R^s	H_R^g	H_R^s
Isolet	0.001	3.50	0.007	3.01	0.003	3.41
Mult. Ftrs.	0.002	3.41	0.008	2.95	0.007	3.01
Arrhythmia	0.007	3.77	0.017	2.80	0.010	3.41
Spambase	0.04	2.71	0.07	2.01	0.05	2.53
Waveform	0.10	1.21	0.14	1.04	0.11	1.08
Ionosphere	0.05	1.81	0.07	1.54	0.07	1.54
Forest	0.10	0.91	0.17	0.82	0.11	0.91
Cancer	0.19	0.82	0.22	0.71	0.19	0.82
Iris	0.17	0.47	0.22	0.31	0.17	0.47

H_R^g: average representation entropy of feature groups, H_R^s: representation entropy of selected subset, λ_2: maximal information compression index, e: least square regression error, ρ: correlation coefficients.

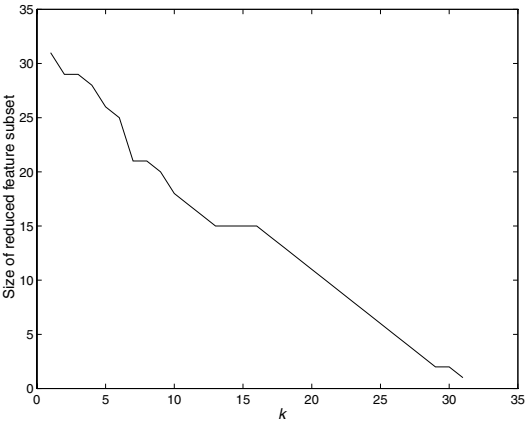
previously, Table 3.6 is provided, where both H_R^s and H_R^g values obtained using each of the similarity measures are compared, in the feature clustering algorithm. It is seen from Table 3.6 that λ_2 has superior information compression capability compared to the other two measures as indicated by the lowest and highest values of H_R^g and H_R^s, respectively.

3.6.3 Effect of cluster size

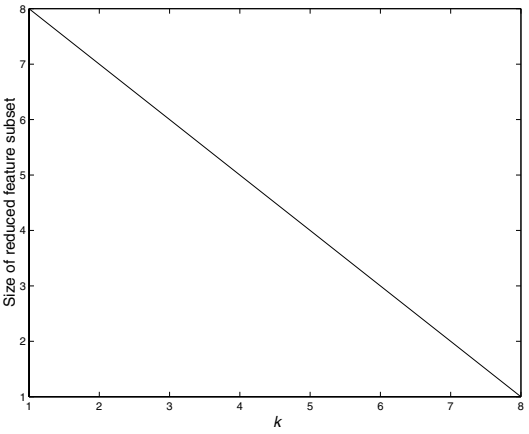
In the FSFS algorithm the size of the reduced feature subset and hence the scale of details of data representation is controlled by the parameter k. Figure 3.4 illustrates such an effect for three data sets – multiple features, ionosphere and cancer, considering one data from each of the high, medium and low categories. As expected, the size of the reduced subset decreases overall with increase in k. However, for medium and particularly large dimensional data (Figure 3.4a) it is observed that for certain ranges of k at the lower side, there is no change in the size of the reduced subset; i.e., no reduction in dimension occurs. Another interesting fact observed in all the data sets considered is that, for all values of k in the case of small dimensional data sets, and for high values of k in the case of medium and large dimensional data sets, the size of the selected subset varies linearly with k. Further, it is seen in those cases, $p + k \approx P$, where p is the size of the reduced subset and P is the size of the original feature set.



(a)



(b)



(c)

FIGURE 3.4: Variation in size of the reduced subset with parameter k for (a) multiple features, (b) ionosphere, and (c) cancer data.

3.7 Summary

After providing a brief review on various feature selection and feature extraction methodologies, an algorithm for unsupervised feature selection using feature similarity measures is described, in detail, for data mining applications. The novelty of the scheme, as compared to other conventional feature selection algorithms, is the absence of search process which contributes to the high computational time requirement of those feature selection algorithms. The algorithm is based on pairwise feature similarity measures, which are fast to compute. It is found to require several orders less CPU time compared to other schemes. Unlike other approaches that are based on optimizing either classification or clustering performance explicitly, here one determines a set of maximally independent features by discarding the redundant ones. In other words, the method is more related to feature selection for information compression rather than for classification/clustering. This enhances the applicability of the resulting features to compression and other tasks such as forecasting, summarization, association mining in addition to classification/clustering. Another characteristic of the aforesaid algorithm is its capability of multiscale representation of data sets. The scale parameter k used for feature clustering efficiently parametrizes the trade-off between representation accuracy and feature subset size. All these make it suitable for a wide variety of data mining tasks involving large (in terms of both dimension and size) data sets.

The feature clustering algorithm uses a novel feature similarity measure called *maximal information compression index*. One may note that the definition of the said parameter is not new; it is its use in feature subset selection framework which is novel. The superiority of this measure for feature selection is established experimentally. It is also demonstrated through extensive experiments that *representation entropy* can be used as an index for quantifying both redundancy reduction and information loss in a feature selection method.

The information loss in this filter approach is measured in terms of second order statistics. The similarity measure used for feature selection is selected/defined accordingly. One may modify these measures suitably in case even higher order statistics are used. In this regard modifications of correlation indices [236] which measure higher order polynomial dependency between variables may be considered. Also the similarity measure is valid only for numeric features; its extension to accommodate other kinds of variables (e.g., symbolic, categorical, hybrid) as input may also be investigated for data mining applications.