Dimension Reduction

Pádraig Cunningham
University College Dublin

Technical Report UCD-CSI-2007-7
August 8th, 2007

Abstract

When data objects that are the subject of analysis using machine learning techniques are described by a large number of features (i.e. the data is high dimension) it is often beneficial to reduce the dimension of the data. Dimension reduction can be beneficial not only for reasons of computational efficiency but also because it can improve the accuracy of the analysis. The set of techniques that can be employed for dimension reduction can be partitioned in two important ways; they can be separated into techniques that apply to *supervised* or *unsupervised* learning and into techniques that either entail *feature selection* or *feature extraction*. In this paper an overview of dimension reduction techniques based on this organisation is presented and representative techniques in each category is described.

1 Introduction

Data analysis problems where the data objects have a large number of features are becoming more prevalent is areas such as multimedia data analysis and bioinformatics. In these situations it is often beneficial to reduce the dimension of the data (describe it in less features) in order to improve the efficiency and accuracy of data analysis. Statisticians sometimes talk of problems that are "Big p Small n"; these are extreme examples of situations where dimension reduction (DR) is necessary because the number of explanatory variables p exceeds (sometimes greatly exceeds) the number of samples n [46]. From a statistical point of view it is desirable that the number of examples in the training set should significantly exceed the number of features used to describe those examples (see Figure 1(a)). In theory the number of examples needs to increase exponentially with the number of features if inference is to be made about the data. In practice this is not the case as real high-dimension data will only occupy a manifold in the input space so the *implicit* dimension of the data will be less than the number of features p. For this reason data sets as depicted in Figure 1(b) can still be analysed.

Nevertheless, traditional algorithms used in machine learning and pattern recognition applications are often susceptible to the well-known problem of the *curse of dimensionality* [5], which refers to the degradation in the performance of a given learning algorithm as

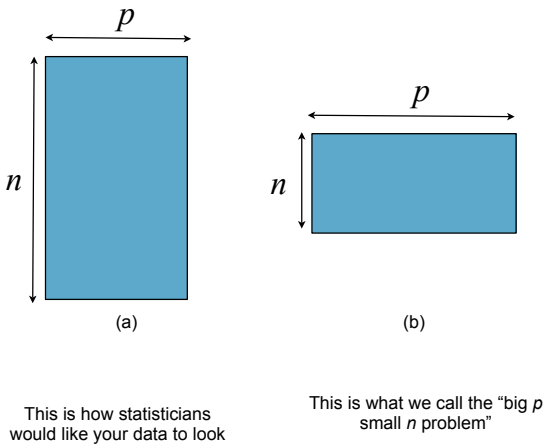

Figure 1: Big p small n problems are problems where the number of features in a dataset is large compared with the number of objects: (a) This is how statisticians would like your data to look, (b) This is what we call the "big p small n problem".

the number of features increases. To deal with this issue, dimension reduction techniques are often applied as a data pre-processing step or as part of the data analysis to simplify the data model. This typically involves the identification of a suitable low-dimensional representation for the original high-dimensional data set. By working with this reduced representation, tasks such as classification or clustering can often yield more accurate and readily interpretable results, while computational costs may also be significantly reduced. The motivation for dimension reduction can be summarised as follows:

- The identification of a reduced set of features that are predictive of outcomes can be very useful from a knowledge discovery perspective.
- For many learning algorithms, the training and/or classification time increases directly with the number of features.
- Noisy or irrelevant features can have the same influence on classification as predictive features so they will impact negatively on accuracy.
- Things look more similar on average the more features used to describe them (see Figure 2). The example in the figure shows that the resolution of a similarity measure can be worse in 20D than in a 5D space.

Research on dimension reduction has itself two dimensions as shown in Figure 3. The first design decision is whether to *select* a subset of the existing features or to *transform* to a new reduced set of features. The other dimension in which DR strategies differ is the question of whether the learning process is *supervised* or *unsupervised*. The dominant strategies used in practice are Principal Components Analysis (PCA) which is an *unsupervised* feature *transformation* technique and *supervised* feature *selection*

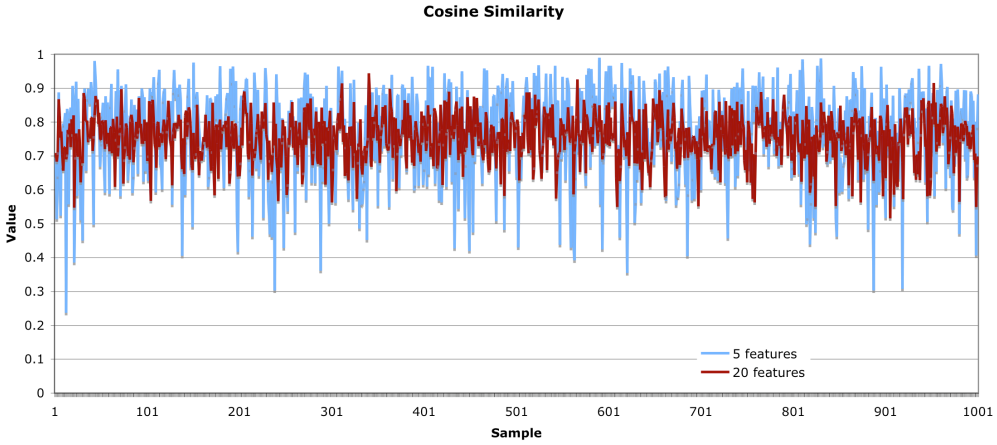

Figure 2: The more dimensions used to describe objects the more similar on average things appear. This figure shows the cosine similarity between randomly generated data objects described by 5 and by 20 features. It is clear that in 20 dimensions similarity has a lower variance than in 5.

strategies such as the use of Information Gain for feature ranking/selection. This paper proceeds with subsections dedicated to each of the four (2×2) possible strategies.

2 Feature Transformation

Feature transformation refers to a family of data pre-processing techniques that transform the original features of a data set to an alternative, more compact set of dimensions, while retaining as much information as possible. These techniques can be sub-divided into two categories:

Feature extraction involves the production of a new set of features from the original features in the data, through the application of some mapping. Well-known unsupervised feature extraction methods include *Principal Component Analysis* (PCA) [18] and spectral clustering (e.g. [36]). The important corresponding supervised approach is *Linear Discriminant Analysis* (LDA) [1].

Feature generation involves the discovery of missing information between features in the original dataset, and the augmentation of that space through the construction of additional features that emphasise the newly discovered information.

Recent work in the literature has primarily focused on the former approach, where the number of extracted dimensions will generally be significantly less than the original number of features. In contrast, feature generation often expands the dimensionality of the data, though feature selection techniques can subsequently be applied to select a smaller subset of useful features.

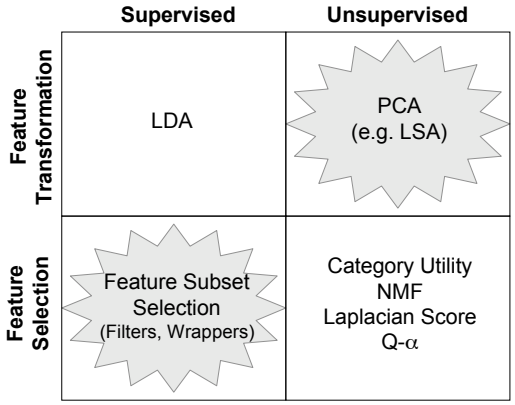

Figure 3: The two key distinctions in dimension reduction research are the distinction between supervised and unsupervised techniques and the distinction between feature transformation and feature extraction techniques. The dominant techniques are feature subset selection and principal component analysis.

For feature transformation let us assume that we have a dataset D made up of $(\mathbf{x}_i)_{i\in[1,n]}$ training samples. The examples are described by a set of features F ($p = |F|$) so there are n objects described by p features. This can be represented by a feature-object matrix $\mathbf{X}_{p\times n}$ where each column represents an object (this is the transpose of what is shown in Figure 1).The objective with Feature Transformation is to transform the data into another set of features F' where $k = |F'|$ and $k < p$, i.e. $\mathbf{X}_{p\times n}$ is transformed to $\mathbf{X'}_{k\times n}$. Typically this is a linear transformation $\mathbf{W}_{k\times p}$ that will transform each object $\mathbf{x}_i$ to $\mathbf{x'}_i$ in k dimensions.

$$\mathbf{x'}_i = \mathbf{W}\mathbf{x}_i \tag{1}$$

The dominant feature transformation technique is Principal Components Analysis (PCA) that transforms the data into a reduced space that captures most of the variance in the data (see section 2.1). PCA is an unsupervised technique in that it does not take class labels into account. By contrast Linear Discriminant Analysis (LDA) seeks a transformation that maximises between-class separation (section 2.2).

2.1 Principal Component Analysis

In PCA the transformation described in equation (1) is achieved so that feature f'_1 is in the dimension in which the variance on the data is maximum, f'_2 is in an orthogonal dimension where the remaining variance is maximum and so on (see Figure (4)).

Central to the whole PCA idea is the covariance matrix of the data $\mathbf{C} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^\mathsf{T}$ [18]. The diagonal terms in $\mathbf{C}$ capture the variance in the individual features and the off-diagonal terms quantify the covariance between the corresponding pairs of features. The objective with PCA is to transform the data so that the the covariance terms are

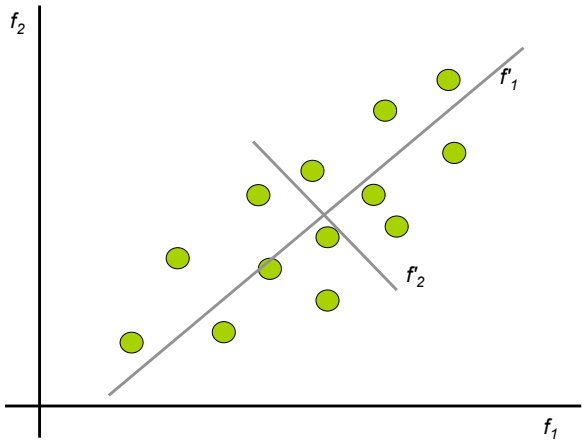

Figure 4: In this example of PCA in 2D the feature space is transformed to f_1' and f_2' so that the variance in the f_1' direction is maximum.

zero, i.e. $\mathbf{C}$ is diagonalised to produce $\mathbf{C}_{PCA}$. The data is transformed by $\mathbf{Y} = \mathbf{PX}$ where the rows of $\mathbf{P}$ are the eigenvectors of $\mathbf{XX}^\mathsf{T}$, then

$$\mathbf{C}_{PCA} = \frac{1}{n-1}\mathbf{YY}^\mathsf{T} \quad (2)$$

$$= \frac{1}{n-1}(\mathbf{PX})(\mathbf{PX})^\mathsf{T} \quad (3)$$

The i^{th} diagonal entry in $\mathbf{C}_{PCA}$ quantifies the variance of the data in the direction of the corresponding principal component. Dimension reduction is achieved by discarding the lesser principal components, i.e. $\mathbf{P}$ has dimension $(k \times p)$ where k is the number of principal components retained.

In multimedia data analysis a variant on the PCA idea called Singular Value Decomposition or Latent Semantic Analysis (LSA) has become popular – this will be described in the next section.

2.1.1 Latent Semantic Analysis

LSA is a variant on the PCA idea presented by Deerwester et al. in [9]. LSA was originally introduced as a text analysis technique so the objects are documents and the features are terms occurring in these text documents – so the feature-object matrix $\mathbf{X}_{p\times n}$ is a term-document matrix. LSA is a method for identifying an informative transformation of documents represented as a bag-of-words in a vector space. It was developed for information retrieval to reveal semantic information from document co-occurrences. Terms that did not appear in a document may still associate with a document. LSA derives uncorrelated index factors that might be considered artificial concepts, i.e. the

latent semantics. LSA is based on a singular-value decomposition of the term-document matrix as follows:

$$\mathbf{X} = \mathbf{TSV}^\mathsf{T} \tag{4}$$

where:

- $\mathbf{T}_{p\times m}$ is the matrix of eigenvectors of $\mathbf{XX}^\mathsf{T}$; m is the rank of $\mathbf{XX}^\mathsf{T}$
- $\mathbf{S}_{m\times m}$ is a diagonal matrix containing the squareroot of the eigenvalues of $\mathbf{XX}^\mathsf{T}$
- $\mathbf{V}_{n\times m}$ is the matrix of eigenvectors of $\mathbf{X}^\mathsf{T}\mathbf{X}$

In this representation the diagonal entries in S are the singular values and they are normally ordered with the largest singular value (largest eigenvalue) first. Dimension reduction is achieved by dropping all but k of these singular values as shown in Figure 5. This gives us a new decomposition:

$$\hat{\mathbf{X}} = \mathbf{T}'\mathbf{S}'\mathbf{V}'^\mathsf{T} \tag{5}$$

where $\mathbf{S}'$ is now $(k \times k)$ and corresponding columns have been dropped in $\mathbf{T}'$ and $\mathbf{D}'$. In this situation $\mathbf{V}'\mathbf{S}'$ is a $(n \times k)$ matrix that gives us the coordinates of the n documents in the new k-dimension space. Reducing the dimension of the data in this way may remove noise and make some of the relationships in the data more apparent. Furthermore the transformation

$$\mathbf{q}' = \mathbf{S}'^{-1}\mathbf{T}'^\mathsf{T}\mathbf{q} \tag{6}$$

will transform any new query $\mathbf{q}$ to this new feature space. This transformation is a linear transformation of the form outlined in (1).

It is easy to understand the potential benefits of LSA in the context of text documents. The LSA process exploits co-occurences of terms in documents to produce a mapping into a *latent semantic* space where documents can be associated even if they have very few terms in common in the original term space. LSA is particularly appropriate for the analysis of text documents because the term document matrix provides an excellent basis on which to perform the singular value decomposition. It has also been employed on other types of media despite the difficulty in identifying a base representation to take the place of the term document matrix. LSA has been employed on; image data [23], video [43] and music and audio [44]. It has also been applied outside of multimedia on gene expression data [38]. More generally PCA is often a key data preprocessing step across a range of disciplines, even if it is not couched in the terms of latent semantic analysis.

The fact that PCA is constrained to be a linear transformation would be considered a shortcoming in many applications. Kernel PCA [33] has emerged as the dominant technique to overcome this. With Kernel PCA the dimension reduction occurs in the kernel induced feature space with the algorithm operating on the kernel matrix representation of the data. The introduction of the kernel function opens up a range of possible non-linear transformations that may be appropriate for the data.

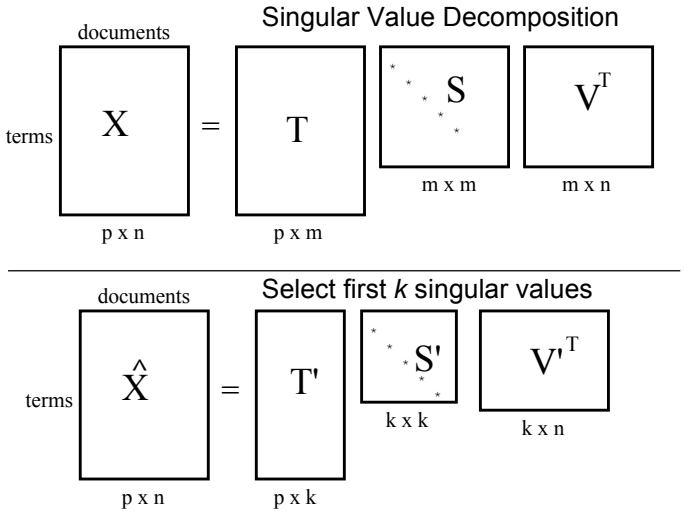

Figure 5: Latent Semantic Analysis is achieved by performing a Singular Value Decomposition on the term document matrix and dropping the least significant singular values, in this scenario k singular values are kept.

2.2 Linear Discriminant Analysis

PCA is *unsupervised* in that it does not take class labels into account. In the supervised context the training examples have class labels attached, i.e. data objects have the form $(\mathbf{x}_i, y_i)$ where $y_i \in C$, a set of class labels or simply $y_i \in \{-1, +1\}$, the binary classification situation. In situations where class labels are available we are often interested in discovering a transformation that emphasises the separation in the data rather than one that discovers dimensions that maximise the variance in the data as happens with PCA. This distinction is illustrated in Figure (6). In this 2D scenario PCA projects the data onto a single dimension that maximises variance; however the two classes are not well separated in this dimension. By contrast Fisher's Linear Discriminant Analysis (LDA) discovers a projection on which the two classes are better separated [15, 16]. This is achieved by uncovering a transformation that maximises between class separation.

While the mathematics underpinning LDA are more complex than those on which PCA is based the principles involved are fairly straightforward. The objective is to uncover a transformation that will maximise between-class separation and minimise within-class separation. To do this we define two scatter matrices, $\mathbf{S}_B$ for between-class separation and $\mathbf{S}_W$ for within-class separation:

$$\mathbf{S}_B = \sum_{c \in C} n_c (\mu_c - \mu)(\mu_c - \mu)^\mathsf{T} \tag{7}$$

$$\mathbf{S}_W = \sum_{c \in C} \sum_{j:y_j = c} (x_j - \mu_c)(x_j - \mu_c)^\mathsf{T} \tag{8}$$

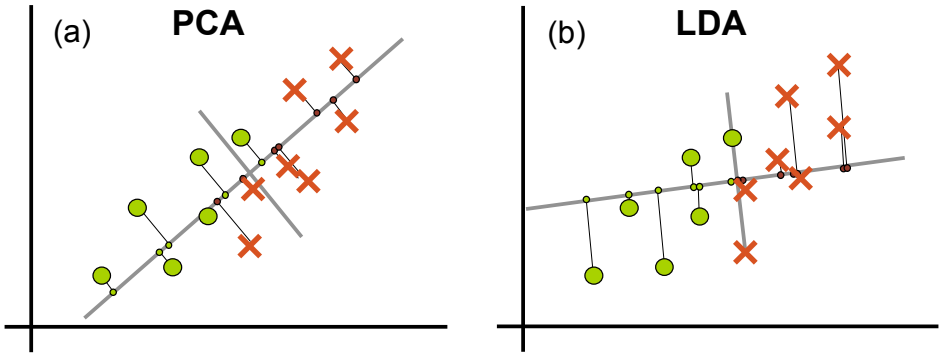

Figure 6: In (a) it is clear that PCA will not necessarily provide a good separation when there are two classes in the data. In (b) LDA seeks a projection that maximises the separation in the data.

where n_c is the number of objects in class c, μ is the mean of all examples and μ_c is the mean of all examples in class c:

$$\mu_c = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \mu_c = \frac{1}{n_c}\sum_{j:y_j=c} x_j \tag{9}$$

The components within these summations (μ, μ_c, x_j) are vectors of dimension p so $\mathbf{S}_B$ and $\mathbf{S}_W$ are matrices of dimension $p \times p$.

The objectives of maximising between-class separation and minimising within-class separation can be combined into a single maximisation called the Fisher criterion [15, 16]:

$$\mathbf{W}_{LDA} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W}^\mathsf{T}\mathbf{S}_B\mathbf{W}|}{|\mathbf{W}^\mathsf{T}\mathbf{S}_W\mathbf{W}|} \tag{10}$$

i.e. find $\mathbf{W} \in \mathbb{R}_{p\times k}$ so that this fraction is maximised ($|\mathbf{A}|$ denotes the determinant of matrix $\mathbf{A}$). This matrix $\mathbf{W}_{LDA}$ provides the transformation described in equation (1). While the choice of k is again open to question it is sometimes selected to be $k = |C| - 1$, i.e. one less than the number of classes in the data.

It transpires that $\mathbf{W}_{LDA}$ is formed by the eigenvectors $(\mathbf{v}_1|\mathbf{v}_2|...\mathbf{v}_k|)$ of ${\mathbf{S}_W}^{-1}\mathbf{S}_B$. The fact that this requires the inversion of $\mathbf{S}_W$ which can be of high dimension can be problematic so the alternative approach is to use simultaneous diagonalisation [29], i.e solve:

$$\mathbf{W}^\mathsf{T}\mathbf{S}_W\mathbf{W} = \mathbf{I} \qquad \mathbf{W}^\mathsf{T}\mathbf{S}_B\mathbf{W} = \mathbf{\Lambda} \tag{11}$$

Here $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues $\{\lambda\}_{i=1}^{k}$ that solve the generalised eigenvalue problem:

$$\mathbf{S}_B\mathbf{v}_i = \lambda_i\mathbf{S}_W\mathbf{v}_i \tag{12}$$

Most algorithms that are available to solve this simultaneous diagonalisation problem require that $\mathbf{S}_W$ be non-singular [29, 24]. This can be a particular issue if the data is of high dimension because more samples than features are required if $\mathbf{S}_W$ is to be non-singular. Addressing this topic is a research issue in its own right [24]. Even if $\mathbf{S}_W$ is non-singular there may still be issues as the "small p large n" problem [46] may manifest itself by overfitting in the dimension reduction process, i.e. dimensions that are discriminating *by chance* in the training data may be selected.

As with PCA the constraint that the transformation is linear is sometimes considered restricting and there has been research on variants of LDA that are non-linear. Two important research directions in this respect are Kernel Discriminant Analysis [4] and Local Fisher Discriminant Analysis [45].

3 Feature Selection

Feature selection (FS) algorithms take an alternative approach to dimension reduction by locating the "best" minimum subset of the original features, rather than transforming the data to an entirely new set of dimensions. For the purpose of knowledge discovery, interpreting the output of algorithms based on feature extraction can often prove to be problematic, as the transformed features may have no physical meaning to the domain expert. In contrast, the dimensions retained by a feature selection procedure can generally be directly interpreted.

Feature selection in the context of supervised learning is a reasonably well posed problem. The objective can be to identify features that are correlated with or predictive of the class label. Or more comprehensively, the objective may be to select features that will construct the most accurate classifier. In unsupervised feature selection the object is less well posed and consequently it is a much less explored area.

3.1 Feature Selection in Supervised Learning

In supervised learning, selection techniques typically incorporate a search strategy for exploring the space of feature subsets, including methods for determining a suitable starting point and generating successive candidate subsets, and an evaluation criterion to rate and compare the candidates, which serves to guide the search process. The evaluation schemes used in both supervised and unsupervised feature selection techniques can generally be divided into three broad categories [25, 6]:

Filter approaches attempt to remove irrelevant features from the feature set prior to the application of the learning algorithm. Initially, the data is analysed to identify those dimensions that are most relevant for describing its structure. The chosen feature subset is subsequently used to train the learning algorithm. Feedback regarding an algorithm's performance is not required during the selection process, though it may be useful when attempting to gauge the effectiveness of the filter.

Table 1: The objective in supervised feature selection is to identify how well the distribution of feature values predicts a class variable. In this example the class variable is binary $\{c_+, c_-\}$ and the feature under consideration has r possible values. n_{i+} is the number of positive examples with feature value i and μ_{i+} is the *expected* value for that figure if the data were uniformly distributed, i.e. $\mu_{i+} = \frac{n_i n_+}{n}$.

Feature Value	c_+	c_-	
v_1	$n_{1+}(\mu_{1+})$	$n_{1-}(\mu_{1-})$	n_1
. . .	. . .	. . .	
v_i	$n_{i+}(\mu_{i+})$	$n_{i-}(\mu_{i-})$	n_i
. . .	. . .	. . .	
v_r	$n_{r+}(\mu_{r+})$	$n_{r-}(\mu_{r-})$	n_r
	n_+	n_-	n

Wrapper methods for feature selection make use of the learning algorithm itself to choose a set of relevant features. The wrapper conducts a search through the feature space, evaluating candidate feature subsets by estimating the predictive accuracy of the classifier built on that subset. The goal of the search is to find the subset that maximises this criterion.

Embedded approaches apply the feature selection process as an integral part of the learning algorithm. The most prominent example of this are the decision tree building algorithms such as Quinlan's C4.5 [40]. There are a number of neural network algorithms that also have this characteristic, e.g. Optimal Brain Damage from Le Cun et al. [27]. Breiman [7] has shown recently that Random Forests, an ensemble technique based on decision trees, can be used for scoring the importance of features. He shows that the increase in error due to perturbing feature values in a data set and then processing the data through the Random Forest is an effective measure of the relevance of a feature.

3.1.1 Filter Techniques

Central to the filter strategy for feature selection is the criterion used to score the predictivness of the features. In this section we will outline three of the most popular techniques for scoring the predictiveness of features - these are the Chi-Square measure, Information Gain and Odds Ratio. The overall scenario is described in Table 1. In this scenario the feature being assessed has r possible values and the table shows the distribution of those values across the classes. Intuitively, the closer these values are to an even distribution the less predictive that feature is of the class. It happens that all three of these techniques as described here require that the features under consideration are discrete valued. These techniques can be applied to numeric features by *discretising* the data. Summary descriptions of the three techniques are as follows:

Chi-Square measure: The Chi-Square measure is based on a statistical test for comparing proportions [48]. It produces a score that follows a χ^2 distribution, however

this aspect is not that relevant from a feature selection perspective as the objective is simply to rank the set of input features. The Chi-Square measure for scoring the *relatedness* of feature f to class c based on data D is as follows:

$$\chi^2(D,c,f) = \sum_{i=1}^{r} \left(\frac{(n_{i+} - \mu_{i+})^2}{\mu_{i+}} + \frac{(n_{i-} - \mu_{i-})^2}{\mu_{i-}} \right) \tag{13}$$

In essence this scores the deviation of counts in each feature-value category against expected values if the feature were not correlated with the class (e.g. n_{i+} is the number of objects that have positive class and feature value v_i, μ_{i+} is the expected value if there were no relationship between f and c).

Information Gain: In recent years information gain (IG) has become perhaps the most popular criterion for feature selection. The IG of a feature is a measure of the amount of information that a feature brings to the training set [40]. It is defined as the expected reduction in entropy caused by partitioning the training set D using the feature f as shown in Equation 14 where D_v is that subset of the training set D where feature f has value v.

$$IG(D,c,f) = Entropy(D,c) - \sum_{v \in values(f)} \frac{|D_v|}{|D|} Entropy(D_v,c) \tag{14}$$

Entropy is a measure of how much randomness or impurity there is in the data set. It is defined in terms of the notation presented in Table 1 for binary classification as follows:

$$Entropy(D,c) = -\sum_{i=1}^{r} \left(\frac{n_{i+}}{n_i} log_2 \frac{n_{i+}}{n_i} + \frac{n_{i-}}{n_i} log_2 \frac{n_{i-}}{n_i} \right) \tag{15}$$

Given that for each feature the entropy of the complete dataset $Entropy(D,c)$ is constant, the set of features can be ranked by IG by simply calculating the remainder term - the second term in equation 14. Predictive features will have small remainders.

Odds Ratio: The odds ratio (OR)[34] is an alternative filtering criterion that is popular in medical informatics. It is really only meaningful to calculate the odds ratio when the input features are binary; we can express this in the notation presented in Table 1 by assigning v_1 to the positive feature value and v_2 to the negative feature value.

$$OR(D,c_+,f) = \frac{n_{1+}/n_{1-}}{n_{2+}/n_{2-}} = \frac{n_{1+}n_{2-}}{n_{2+}n_{1-}} \tag{16}$$

For feature selection, the features can be ranked according to their OR with high values indicating features that are very predictive of the class. The same can be done for the negative class to highlight features that are predictive of the negative

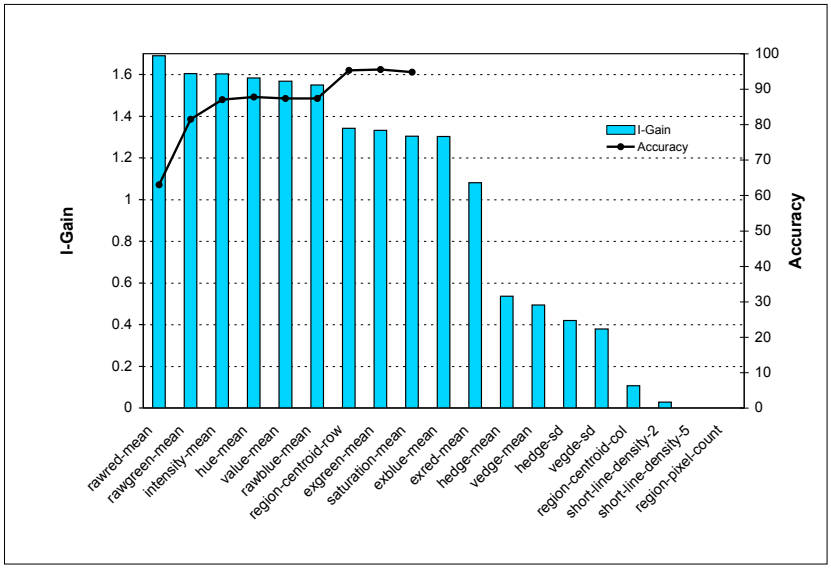

Figure 7: This graph shows features from the UCI segment dataset scored by IG and also the accuracies of classifiers built with the top ranking sets of features.

class. Where a specific feature does not occur in a class, it can be assigned a small fixed value so that the OR can still be calculated.

Filtering Policy: The three filtering measures (Chi-Square, Information Gain and Odds Ratio) provide us with a *principle* on which a feature set might be filtered; we still require a filtering *policy*. There is a variety of policies that can be employed:

1. Select the top m of n features according to their score on the filtering criterion (e.g. select the top 50%).

2. Select all features that score above some threshold T on the scoring criterion (e.g. select all features with a score within 50% of the maximum score).

3. Starting with the highest scoring feature, evaluate using cross-validation the performance of a classifier built with that feature. Then add the next highest ranking feature and evaluate again; repeat until no further improvements are achieved.

This third strategy is simple but quite straightforward. An example of this strategy in operation is presented in Figure 7. The graph shows the IG scores of the features in the UCI segment dataset [35] and the accuracies of classifiers built with the top feature, the top two features and so on. It can be seen that after the ninth feature ($\mathsf{saturation-mean}$) is added the accuracy drops slightly so the process would stop after selecting the first eight features. While this strategy is straightforward and effective it does have some potential shortcomings. The features are scored in isolation so two highly correlated

features can be selected even if one is redundant in the presence of the other. The full space of possible feature subsets is not explored so there may be some very effective feature subsets that act in concert that are not discovered.

While these strategies are effective for feature selection they have the drawback that features are considered in isolation so redundancies or dependancies are ignored as already mentioned. Two strongly correlated features may both have high IG scores but one may be redundant once the other is selected. More sophisticated filter techniques that address these issues using Mutual Information to score *groups* of features have been researched by Novovičová et al. [39] and have been shown to be more effective than these simple filter techniques.

3.1.2 Wrapper Techniques

The obvious criticism of the filter approach to feature selection is that the filter criterion is separate from the induction algorithm used in the classifier. This is overcome in the wrapper approach by using the performance of the classifier to guide search in feature selection – the classifier is *wrapped* in the feature selection process [26]. In this way the merit of a feature subset is the generalisation accuracy it offers as estimated using cross-validation on the training data. If 10-fold cross validation is used then 10 classifiers will be built and tested for each feature subset evaluated – so the wrapper strategy is very computationally expensive. If there are p features under consideration then the search space is of size 2^p so it is an exponential search problem.

A simple example of the search space for feature selection where $p = 4$ is shown in Figure 8. Each node is defined by a feature mask; the node at the top of the figure has no features selected while the node at the bottom has all features selected. For large values of p an exhaustive search is not practical because of the exponential nature of the search. Four popular strategies are:

- **Forward Selection (FS)** which starts with no features selected, evaluates all the options with just one feature, selects the best of these and considers the options with that feature plus one other, etc.
- **Backward Elimination (BE)** starts with all features selected, considers the options with one feature deleted, selects the best of these and continues to eliminate features.
- **Genetic Search** uses a genetic algorithm (GA) to search through the space of possible feature sets. Each state is defined by a feature mask on which crossover and mutation can be performed [30]. Given this convenient representation, the use of a GA for feature selection is quite straightforward although the evaluation of the fitness function (classifier accuracy as measured by cross-validation) is expensive.
- **Simulated Annealing** is an alternative stochastic search strategy to GAs [31]. Unlike GAs, where a population of solutions is maintained, only one solution (i.e. feature mask) is under in Simulated Annealing (SA). SA implements a stochastic

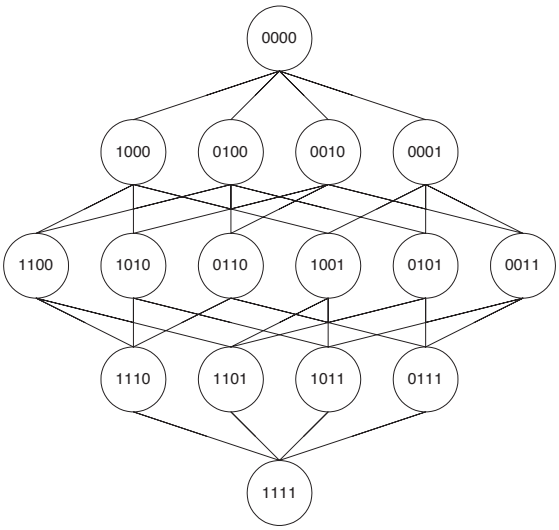

Figure 8: The search space of feature subsets when $n = 4$. Each node is represented by a feature mask; in the topmost node no features are selected and in the bottom node all features are selected.

search since there is a chance that some deteriorations in solution are accepted – this allows a more effective exploration of the search space.

The first two strategies will terminate when adding (or deleting) a feature will not produce an improvement in classification accuracy as assessed by cross validation. Both of these are greedy search strategies and so are not guaranteed to discover the best feature subset. More sophisticated search strategies such as GA or SA can be employed to better explore the search space; however, Reunanen [41] cautions that more intensive search strategies are more likely to overfit the training data.

A simple example of BE is shown in Figure 9. In this example there are just four features (A,B,C and D) to consider. Cross-validation gives the full feature set a score of 71%, the best feature set of size 3 is (A,B,D) and the best feature set of size 2 is (A,B) and the feature sets of size 1 are no improvement on this.

Of the two simple wrapper strategies (BE and FS) BE is considered to be more effective as it more effectively considers features in the context of other features [3].

3.2 Unsupervised Feature Selection

Feature selection in a supervised learning context is a well posed problem in that the objective can be clearly expressed. The objective can be to identify features that are correlated with the outcome or to identify a set of features that will build an accurate classifier – in either case the objective is to discover a reduced set of the original features in which the classes are well separated. By contrast feature selection in an unsupervised context is ill posed in that the overall objective is less clear. The difficulty is further

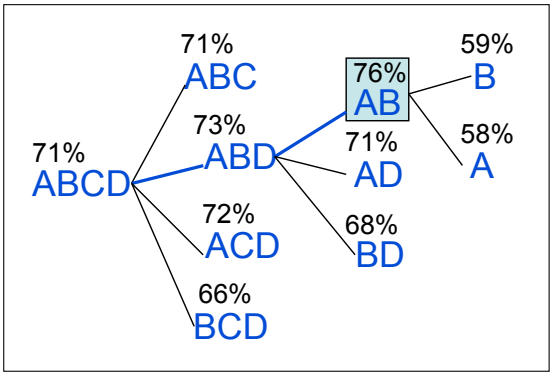

Figure 9: This graph shows a wrapper-based search employing backward elimination. The search starts with all features (A,B,C,D) selected – in this example this is judged to have a score of 71%. The best feature subset uncovered in this example would be (AB) which has a score of 76%.

exacerbated by the fact that the number of clusters in the data is generally not known in advance; this further complicates the problem of finding a reduced set of features that will help *organise* the data.

If we think of unsupervised learning as clustering then the objective with feature selection for clustering might be to select features that produce clusters that are well separated. This objective can be problematic as different feature subsets can produce different well separated clusterings. This can produce a "chicken and egg" problem where the question is "which comes first, the feature selection or the clustering?". A simple example of this is shown in Figure 10; in this 2D example selecting feature f_1 produces the clustering $\{C_a, C_b\}$ while selecting f_2 produces the clustering $\{C_x, C_y\}$. So there are two alternative and very different valid solutions. If this data is initially clustered in 2D with $k = 2$ then it is likely that partition $\{C_x, C_y\}$ will be selected and then feature selection would select f_2.

This raises a further interesting question, does this clustering produced on the original (full) data description have special status? The answer to this is surely problem dependent; in problems such as text clustering, there will be many irrelevant features and the clustering on the full vocabulary might be quite noisy. On the other hand, in carefully designed experiments such as gene expression analysis, it might be expected that the clustering on the full data description has special merit. This co-dependence between feature selection and clustering is a big issue in feature selection for unsupervised learning; indeed Dy & Brodley [13] suggest that research in this area can be categorised by where the feature selection occurs in the clustering process:

Before clustering: To perform feature selection prior to clustering is analogous to the filter approach to supervised feature selection. A simple strategy would be to employ variance as a ranking criterion and select the features in which the data has the highest variance [11]. A more sophisticated strategy in this category is the

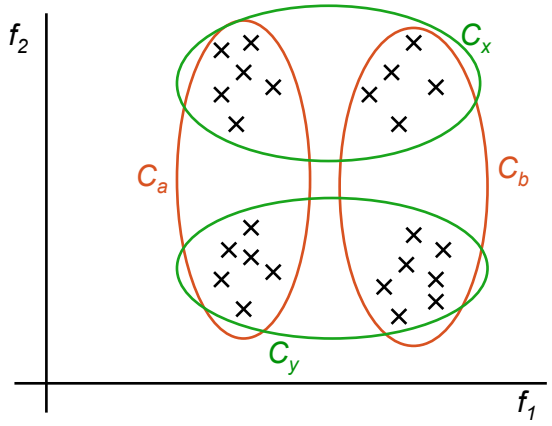

Figure 10: Using cluster separation as a criterion to drive unsupervised feature selection is problematic because different feature selections will produce different clusterings with good separation. In this example if f_1 is selected then the obvious clustering is $\{C_a, C_b\}$, if f_2 is selected then $\{C_x, C_y\}$ is obvious.

Laplacian Score [21] described in section 3.2.1.

During clustering: Given the co-dependence between the clustering and the feature selection process, it makes sense to integrate the two processes if that is possible. Three strategies that do this are; the strategy based on category utility described in section 3.2.2, the $Q-\alpha$ algorithm [47] described in section 3.2.3 and biclustering [8].

After clustering: If feature selection is postponed until after clustering then the range of supervised feature selection strategies can be employed as the clusters can be used as class labels to provide the *supervision*. However, the strategy of using a set of feature for clustering and then deselecting some of those features because they are deemed to be not relevant will not make sense in some circumstances.

One of the reasons why unsupervised feature selection is a challenging problem is because the success criterion is ill posed as stated earlier. This is particularly an issue if the feature selection stage is to be integrated into the clustering process. Two criteria that can be used to quantify a good partition are the criterion based on the scatter matrices presented in section 2.2 and category utility which is explained in section 3.2.2. The objective with the criterion based on scatter is to maximise $trace(\mathbf{S}_W^{-1}\mathbf{S}_B)$ [13] – this is particularly appropriate when the data is numeric. For categorical data the category utility measure described in the next section is applicable.

In the remainder of this section on unsupervised feature selection we will describe a variety of unsupervised feature selection techniques that have emerged in recent research. These techniques will be organised into the categories of; filter, wrapper and embedded in

the same manner as in the section on supervised feature selection (section 3.1). However, the distinction between these categories is less clear-cut in the unsupervised case.

3.2.1 Unsupervised Filters

The defining characteristic of a filter-based feature selection technique is that features are scored or ranked by a criterion that is separate from the classification or clustering process.

A prominent example of such a strategy is the Laplacian score that can be used as a criterion in dimension reduction when the motivation is that *locality* is preserved. Such locality preserving projections [22] are appropriate in image analysis where images that are similar in the input space should also be similar in the reduced space. The Laplacian Score (LS) embodies this idea for unsupervised feature selection [21]. LS selects features so that objects that are close in the input space are still close in the reduce space. This is an interesting criterion to optimise as it contains the implication that none of the input features are irrelevant; they may be just redundant.

The calculation of LS is based on a graph G that captures nearest neighbour relationships between the n data points. G is represented by a square matrix $\mathbf{S}$ where $\mathbf{S}_{ij} = 0$ unless x_i and x_j are neighbours, in which case:

$$\mathbf{S}_{ij} = e^{-} \frac{||x_i - x_j||^2}{t} \tag{17}$$

where t is a bandwith parameter. The neighbourhood idea introduces another parameter k which is the number of neighbours used to construct $\mathbf{S}$. $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is the Laplacian of this graph where $\mathbf{D}$ is a degree diagonal matrix $\mathbf{D}_{ii} = \sum_j \mathbf{S}_{ij}$, $D_{ij,i\neq j} = 0$ [42]. If $\mathbf{m}_i$ is the vector of values in the dataset for the i^{th} feature then the LS is defined using the following calculations [21]:

$$\widetilde{\mathbf{m}}_i = \mathbf{m}_i - \frac{\mathbf{m}_i^\top \mathbf{D1}}{\mathbf{1}^\top \mathbf{D1}} \mathbf{1} \tag{18}$$

where $\mathbf{1}$ is a vector or 1s of length n. Then the Laplacian Score for the i^{th} feature is:

$$LS_i = \frac{\widetilde{\mathbf{m}}_i^\top \mathbf{L} \widetilde{\mathbf{m}}_i}{\widetilde{\mathbf{m}}_i^\top \mathbf{D} \widetilde{\mathbf{m}}_i} \tag{19}$$

This can be used to score all the features in the data set on how effective they are in preserving locality. This has been shown to be an appropriate criterion for dimension reduction in applications such as image analysis where locality preservation is an effective motivation [21]. However, if the data contains irrelevant features, as can occur in text classification or the analysis of gene expression data, then locality preservation is not a sensible motivation. In such circumstances selecting the features in which the data has the highest variance [11] might be a more appropriate filter.

3.2.2 Unsupervised Wrappers

The defining characteristic of a wrapper-based feature selection technique is that the classification or clustering process is used to evaluate feature subsets. This is more problematic in clustering than in classification as there is no single criterion that can be used to score cluster quality and many cluster validity indices have biases – e.g. toward small numbers of clusters or balanced cluster sizes [12, 19]. Nevertheless there has been work on unsupervised wrapper-like feature selection techniques and two such techniques – one based on category utility, and the other based on the EM clustering algorithm – are described here.

Category Utility: Devaney and Ram [10] proposed a wrapper-like unsupervised feature subset selection algorithm based on the notion of category utility (CU) [17]. This was implemented in the area of conceptual clustering, using Fisher's [14] COBWEB system as the underlying concept learner. Devaney and Ram demonstrate that if feature selection is performed as part of the process of building the concept hierarchy (i.e. concepts are defined by a subset of features) then a better concept hierarchy is developed. As with the original COBWEB system, they use CU as their evaluation function to guide the process of creating concepts – the CU of a clustering C based on a feature set F is defined as follows:

$$CU(C,F) = \frac{1}{k} \sum_{c_l \in C} \left[\sum_{f_i \in F} \sum_{j=1}^{r_i} p(f_{ij}|C_l)^2 - \sum_{f_i \in F} \sum_{j=1}^{r_i} p(f_{ij})^2 \right] \tag{20}$$

where $C = \{C_1, ..., C_l, ..., C_k\}$ is the set of clusters and $F = \{F_1, ..., F_i, ..., F_p\}$ is the set of features. CU measures the difference between the conditional probability of a feature i having value j in cluster l and the prior probability of that feature value. The inner most sum is over r feature values, the middle sum is over p features and the outer sum is over k clusters. This function measures the increase in the number of feature values that can be predicted correctly given a set of concepts, over those which can be predicted without using any concepts.

Their approach was to generate a set of feature subsets (using either FS or BE as described in section 3.1.2), run COBWEB on each subset, and then evaluate each resulting concept hierarchy using the category utility metric on the first partition. BSS starts with the full feature set and removes the least useful feature at each stage until utility stops improving. FSS starts with an empty feature set and adds the feature providing the greatest improvement in utility at each stage. At each stage the algorithm checks how many feature values can be predicted correctly by the partition – i.e. if the value of each feature f can be predicted for most of the clusters C_l in the partition, then the features used to produce this partition were informative or relevant. The highest scoring feature subset is retained, and the next larger (or smaller) subset is generated using this subset as a starting point. The process continues until no higher CU score can be achieved.

The key idea here is that CU is used to score the quality of clusterings in a wrapper-like search. It has been shown by Gluck and Corter [17] that CU corresponds to mutual information so this is a quite a principled way to perform unsupervised feature selection.

Devaney and Ram improved upon the time it takes to reconstruct a concept structure by using their own concept learner, Attribute-Incremental Concept Creator (AICC), instead of COBWEB. AICC can add features without having to rebuild the concept hierarchy from scratch, and shows large speedups.

Expectation Maximisation (EM): Dy & Brodley present a comprehensive analysis of unsupervised wrapper-based feature selection in [13]. They present their analysis in the context of the EM clustering algorithm [32]. Specifically, they consider wrapping the EM clustering algorithm where feature subsets are evaluated with criteria based on cluster separability and maximum likelihood. However they emphasise that the approach is general and can be used with any clustering algorithm by selecting an appropriate criterion for scoring the clusterings produced by different feature subsets. They discuss the biases associated with cluster validation techniques (e.g. biases on cluster size, data dimension, a balanced cluster sizes) and propose ways in which some of these issues can be ameliorated.

3.2.3 The Embedded Approach

The final category of feature selection technique mentioned in section 3.1 is the embedded approach, i.e. feature selection is an integral part of the classification algorithm as happens for instance in the construction of decision trees [40] or in some types of neural network [27]. In the unsupervised context this general approach is a good deal more prominent. There are a number of clustering techniques that have dimension reduction as a by-product of the clustering process, for example; Non-negative Matrix Factorisation (NMF) [28], biclustering [20] and projected clustering [2]. These approaches have in common that they discover clusters in the data that are defined by a subset of the features and different clusters can be defined by different feature subsets. Thus these are implicitly *local* feature selection techniques.

The alternative to this is a *global* approach where the same feature subset is used to describe all clusters. A representative example of this is Q-α algorithm presented by Wolf and Shashua [47].

The Q-α Approach: A well motivated criterion of cluster quality is cluster coherence, in graph theoretic terms this is expressed by the notion of objects within clusters being well connected and individual clusters being weakly linked. The whole area of spectral clustering captures these ideas in a well founded family of clustering algorithms based on the idea of minimising the *graph-cut* between clusters [37].

The principles of spectral clustering have been extended by Wolf and Shashua [47] to produce the $Q - \alpha$ algorithm that simultaneously performs feature subset selection and discovers a good partition of the data. As with spectral clustering, the fundamental

data structure is the affinity matrix $\mathbf{A}$ where each entry $\mathbf{A}_{ij}$ captures the similarity (in this case as a dot-product) between data points i and j. In order to facilitate feature selection the affinity matrix for $Q-\alpha$ is expressed as $\mathbf{A}_\alpha = \sum_{i=1}^{p} \alpha_i \mathbf{m}_i \mathbf{m}_i^\mathsf{T}$ where $\mathbf{m}_i$ is the i^{th} row in the data matrix that has been normalised so to be centred on 0 and be of unit L_2 norm (this is the set of values in the data set for feature i). $\mathbf{m}_i \mathbf{m}_i^\mathsf{T}$ is the *outer*-product of $\mathbf{m}_i$ with itself. α is the weight vector for the p features – ultimately the objective is for most of these weight terms to be set to 0.

In spectral clustering $\mathbf{Q}$ is an $n \times k$ matrix composed of the k eigenvectors of $\mathbf{A}$ corresponding to the largest k eigenvalues. Wolf and Shashua show that the relevance of a feature subset as defined by the weight vector α can be quantified by:

$$Rel(\alpha) = trace(\mathbf{Q}^\mathsf{T} \mathbf{A}_\alpha^\mathsf{T} \mathbf{A}_\alpha \mathbf{Q}) \tag{21}$$

They show that feature selection and clustering can be performed as a single process by optimising:

$$\max_{\mathbf{Q}\alpha} trace(\mathbf{Q}^\mathsf{T} \mathbf{A}_\alpha^\mathsf{T} \mathbf{A}_\alpha \mathbf{Q}) \tag{22}$$

subject to $\alpha^\mathsf{T}\alpha = 1$ and $\mathbf{Q}^\mathsf{T}\mathbf{Q} = \mathbf{I}$.

Wolf and Shashua show that this can be solved by solving two inter-linked eigenvalue problems that produce solutions for α and $\mathbf{Q}$. They show that a process of iteratively solving for α then fixing α and solving for $\mathbf{Q}$ will converge. They also show that the process has the convenient property that the α_i weights are biased to be positive and sparse, i.e. many of them will be zero.

So the $Q-\alpha$ algorithm performs feature selection in the spirit of spectral clustering, i.e. the motivation is to increase cluster coherence. It discovers a feature subset that will support a partitioning of the data where clusters are well separated according to a graph-cut criterion.

4 Conclusions

The objective with this paper was to provide an overview of the variety of strategies that can be employed for dimension reduction when processing high dimension data. When feature transformation is appropriate then PCA is the dominant technique if the data is not labelled. If the data is labelled then LDA can be applied to discover a projection of the data that separates the classes. When feature selection is required and the data is labelled then the problem is well posed. A variety of filter and wrapper-based techniques for feature selection are described in section 3.1. The paper concludes with a review of unsupervised feature selection in section 3.2. This is a more difficult problem than the supervised situation in that the success criterion is less clear. Nevertheless this is an active research area at the moment and a variety of unsupervised feature selection strategies have emerged.

Acknowledgements

I would like to thank Derek Greene, Santiago Villalba and Anthony Brew for their comments on an earlier draft of this paper.

References

[1] A. Hyvärinen, J. Karhunen and E. Oja. *Independent Component Analyis*. John Wiley & Sons, Inc, 2001.

[2] C.C. Aggarwal, J.L. Wolf, P.S. Yu, C. Procopiuc, and J.S. Park. Fast algorithms for projected clustering. *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 61–72, 1999.

[3] D.W. Aha and R.L. Bankert. A comparative evaluation of sequential feature selection algorithms. *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, pages 1–7, 1995.

[4] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000.

[5] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[6] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

[7] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[8] K. Bryan, P. Cunningham, and N. Bolshakova. Biclustering of expression data using simulated annealing. In *CBMS*, pages 383–388. IEEE Computer Society, 2005.

[9] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[10] M. Devaney and A. Ram. Efficient feature selection in conceptual clustering. In Douglas H. Fisher, editor, *ICML*, pages 92–97. Morgan Kaufmann, 1997.

[11] M. Doyle and P. Cunningham. A dynamic approach to reducing dialog in on-line decision guides. In Enrico Blanzieri and Luigi Portinale, editors, *EWCBR*, volume 1898 of *Lecture Notes in Computer Science*, pages 49–60. Springer, 2000.

[12] R.C. Dubes. How many clusters are best?—an experiment. *Pattern Recognition*, 20(6):645–663, 1987.

[13] J.G. Dy and C.E. Brodley. Feature Selection for Unsupervised Learning. *The Journal of Machine Learning Research*, 5:845–889, 2004.

[14] D.H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2):139–172, 1987.

[15] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179 – 188, 1936.

[16] K. Fukunaga. *Introduction to statistical pattern recognition.* Academic Press, Inc, 2nd edition, 1990.

[17] M. A. Gluck and J. E. Corter. Information, uncertainty, and the utility of categories. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, pages 283–287, Hillsdale, NJ, 1985. Lawrence Earlbaum.

[18] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417 – 441, 1933.

[19] J. Handl, J. Knowles, and D.B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, 2005.

[20] JA Hartigan. Direct Clustering of a Data Matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

[21] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *NIPS*, 2005.

[22] X. He and P. Niyogi. Locality preserving projections. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *NIPS*. MIT Press, 2003.

[23] D. R. Heisterkamp. Building a latent semantic index of an image database from patterns of relevance feedback. In *ICPR (4)*, pages 134–137, 2002.

[24] R. Huang, Q. Liu, H. Lu, and S. Ma. Solving the Small Sample Size Problem of LDA. In *ICPR (3)*, pages 29–32, 2002.

[25] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the 11th International Conference on Machine Learning*, pages 121–129, New Brunswick, NJ, 1994. Morgan Kaufmann.

[26] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1-2):273–324, 1997.

[27] Y. LeCun, J. Denker, S. Solla, R. E. Howard, and L. D. Jackel. Optimal brain damage. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems II*, San Mateo, CA, 1990. Morgan Kauffman.

[28] D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[29] W. Liu, Y. Wang, S. Z. Li, and T. Tan. Null space approach of fisher discriminant analysis for face recognition. In Davide Maltoni and Anil K. Jain, editors, *ECCV Workshop BioAW*, volume 3087 of *Lecture Notes in Computer Science*, pages 32–44. Springer, 2004.

[30] J. Loughrey and P. Cunningham. Overfitting in Wrapper-Based Feature Subset Selection: The Harder You Try the Worse it Gets. *24th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (AI-2004)*, pages 33–43, 2004.

[31] J. Loughrey and P. Cunningham. Using early-stopping to avoid overfitting in wrapper-based feature subset selection employing stochastic search. In M. Petridis, editor, *10th UK Workshop on Case-Based Reasoning*, pages 3–10. CMS Press, 2005.

[32] G.J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley, 1997.

[33] S. Mika, B. Schölkopf, A. J. Smola, K. R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and De-Noising in Feature Spaces. In Michael J. Kearns, Sara A. Solla, and David A. Cohn, editors, *NIPS*, pages 536–542. The MIT Press, 1998.

[34] D. Mladenic. Feature subset selection in text-learning. In C. Nedellec and C. Rouveirol, editors, *ECML*, volume 1398 of *Lecture Notes in Computer Science*, pages 95–100. Springer, 1998.

[35] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. UCI repository of machine learning databases, 1998.

[36] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proc. Advances in Neural Information Processing*, 2001.

[37] A.Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14(2):849–856, 2001.

[38] S. K. Ng, Z. Zhu, and Y. S. Ong. Whole-genome functional classification of genes by latent semantic analysis on microarray data. In Yi-Ping Phoebe Chen, editor, *APBC*, volume 29 of *CRPIT*, pages 123–129. Australian Computer Society, 2004.

[39] J. Novovičová, A. Malík, and P. Pudil. Feature selection using improved mutual information for text classification. In A. L. N. Fred, T. Caelli, R. P. W. Duin, A. C. Campilho, and D. de Ridder, editors, *SSPR/SPR*, volume 3138 of *Lecture Notes in Computer Science*, pages 1010–1017. Springer, 2004.

[40] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[41] J. Reunanen. Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3:1371–1382, 2003.

[42] M. Saerens, F. Fouss, L. Yen, and P. Dupont. The principal components analysis of a graph, and its relationships to spectral clustering. *Proceedings of the 15th European Conference on Machine Learning (ECML 2004). Lecture Notes in Artificial Intelligence*, 3201:371–383, 2004.

[43] E. Sahouria and A. Zakhor. Content analysis of video using principal componets. In *ICIP (3)*, pages 541–545, 1998.

[44] P. Smaragdis, B. Raj, and M. Shashanka. A probabilistic latent variable model for acoustic modeling. In *Workshop on Advances in Models for Acoustic Processing at NIPS 2006*, 2006.

[45] M. Sugiyama. Local Fisher discriminant analysis for supervised dimensionality reduction. In William W. Cohen and Andrew Moore, editors, *ICML*, pages 905–912. ACM, 2006.

[46] M. West. Bayesian factor regression models in the "large p, small n" paradigm. *Bayesian Statistics*, 7:723–732, 2003.

[47] L. Wolf and A. Shashua. Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *Journal of Machine Learning Research*, 6:1855–1887, 2005.

[48] S. Wu and P.A. Flach. Feature selection with labelled and unlabelled data. *Proceedings of ECML/PKDD'02 Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning*, pages 156–167, 2002.