Dimensionality Reduction through Sub-Space Mapping for Nearest Neighbour Algorithms

Terry R. Payne[1] and Peter Edwards[2]

[1] The Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15232, USA.
[2] Department of Computing Science, King's College, University of Aberdeen, Aberdeen, Scotland, AB24 3UE.

Abstract. Many learning algorithms make an implicit assumption that all the attributes present in the data are relevant to a learning task. However, several studies have demonstrated that this assumption rarely holds; for many supervised learning algorithms, the inclusion of irrelevant or redundant attributes can result in a degradation in classification accuracy. While a variety of different methods for dimensionality reduction exist, many of these are only appropriate for datasets which contain a small number of attributes (e.g. < 20). This paper presents an alternative approach to dimensionality reduction, and demonstrates how it can be combined with a Nearest Neighbour learning algorithm. We present an empirical evaluation of this approach, and contrast its performance with two related techniques; a *Monte-Carlo* wrapper and an *Information Gain*-based filter approach.

1 Introduction

The dimensionality of a supervised learning task can be characterised in many ways. A dataset contains a number of situations or instances, each of which contain several attributes and a class value. The attributes may be considered to be *predictor* (relevant) attributes, as they may be used to induce a classification hypothesis (sometimes represented as a set of rules or a decision tree) which is later used to predict the class of a new instance. However, other attributes may be considered as *irrelevant* attributes, as they contribute nothing to the classification task, and may even degrade the accuracy of the resulting classifications. The time taken to induce a concept description from a training set, and to predict the class of a new instance, is dependent on both the learning algorithm used and the number of attributes present (i.e. the number of dimensions used to describe the data).

Determining which of the attributes are relevant to the learning task (i.e. identifying attributes which predict the class value) is a central problem in machine learning. In the past, domain experts selected the attributes believed to be relevant to the learning task. However, in the absence of such background knowledge, automatic techniques are required to identify such attributes, as the presence of irrelevant attributes can reduce the performance of various learning

techniques. Nearest neighbour algorithms are especially prone to the inclusion of such attributes within datasets, as many utilise distance metrics that calculate an average similarity measure across all of the attributes [1]. In addition to this, the sample complexity (i.e. the number of instances required to learn a concept) grows exponentially with the number of irrelevant attributes [11], indicating that simple nearest neighbour algorithms may not scale up well if irrelevant attributes are present. For these reasons, various weighting techniques have been investigated in an attempt to reduce the contribution of irrelevant attributes within nearest neighbour algorithms [14].

A *redundant-attribute* set occurs when two or more relevant attributes exist, such that each makes an equal contribution towards learning some concept [10]. In general, only a single member of this redundant-attribute set is required when learning the concept. The inclusion of more than one member will not only increase the time taken to induce the concept description, but may place emphasis on the part of the concept description the attributes in the set represent, and thus reduce the influence of other relevant attributes [12]. The remaining attributes in this set are sometimes described as *redundant*.

In this paper, we present an alternative approach that can be used by machine learning algorithms to reduce the dimensionality of datasets. The instances in a dataset are represented as vectors within an instance space. An approximation of this space is then found, and the vectors are projected into this lower dimensional space. This is achieved by using the geometric technique, *Correspondence Analysis* [9], to identify and approximate the lower dimensional space (or *sub-space*). This sub-space can then be used by a nearest neighbour learning algorithm to perform class predictions for new instances. The two learning algorithms, *CA* and *CACP* utilise this approach to dimensionality reduction, and are described below.

2 Dimensionality Reduction for Machine Learning and Information Retrieval Systems

The dimensionality reduction techniques used by machine learning algorithms can be grouped into two broad categories: those that are instances of the *filter* model, where the selection technique is independent of the learning algorithm used to learn the concept hypothesis; and those that are instances of the *wrapper* model, where the learning algorithm is integral to the selection mechanism [10]. Both models perform a search within a space of attribute subsets to determine the optimal (or sub-optimal) subset for the classification task. The size of the search space is exponential; if there are n attributes in the original dataset, then there are a total of 2^n possible states in the search space. This exponential rise means that exhaustive, optimal searches are infeasible for all but simple problems involving few attributes. Therefore, most systems perform greedy or stochastic searches. Several studies have also shown that the wrapper model can identify better attribute sets, when compared with the filter model [10]. However, induction is performed at every search state visited. The number of instances, i,

in the training set and the control mechanism used to evaluate each state will also influence the length of time taken to determine the final attribute subset.

Dimensionality reduction techniques have also been utilised by a variety of Information Retrieval (IR) systems [18] to reduce the number of terms used to index documents. These techniques have also been applied to the problem of reducing the number of terms presented to learning algorithms for text categorisation problems [7]. Whilst some studies have omitted this stage, the number of unique terms (typically in the region of tens or hundreds of thousands) is prohibitively high for most machine learning algorithms. Many text categorisation systems employ *filter* based methods. *Latent Semantic Indexing* (LSI) [7] is an alternative approach for reducing the number of dimensions used to represent documents in many IR systems. LSI utilises an orthogonal decomposition technique to determine a smaller numeric representation for each document. A corpus is represented as a *term* $\times$ *document* matrix, where each row corresponds to a document, and each column to one of the terms appearing within the corpus. Thus, each document (i.e. row vector) is expressed as a point within some geometric space. An orthogonal decomposition technique is then applied to this matrix, resulting in a set of decomposed matrices that describe this space and the points within it. The space can then be approximated (by approximating the decomposed matrices) resulting in a lower dimensional representation of the points [15].

Various studies have demonstrated that LSI improved the performance of both IR and text categorisation systems. For example, Deerwester et. al. [7] achieved a reduction from 5000-7000 terms to 100 dimensions. Similar techniques have also been successfully applied to the problem of reducing the dimensionality of protein sequence data for presentation to neural networks [19]. The size of the input vectors presented to a backward propagation neural network was reduced from 9696 to 100, resulting in an overall improvement in the predictive accuracy of the neural network. These studies have demonstrated that LSI and the principles behind this method work for specific problems, but LSI's applicability to a broader range of classification tasks has not yet been investigated. For this reason, we have investigated a similar technique, based on *Correspondence Analysis* [9], and have developed two learning algorithms, *CA* and *CACP*, which combine variations of this technique with a Euclidean nearest neighbour learning algorithm. These algorithms have been applied to a variety of classification problems found in the UCI Machine Learning Database Repository [4], and to artificial data (described in Section 5).

3 Subspace Approximation through Correspondence Analysis

Correspondence analysis is a mathematical tool that is used to graphically present multi-dimensional data within low (e.g. two or three) dimensional data plots [14, 15]. This is achieved by identifying an approximation of the Euclidean space that contains the instances (which are represented as vectors). This ap-

proximation is used to project the vectors from a J-dimensional instance space into a K-dimensional sub-space, where J is the number of attributes of the dataset, and consequently the number of components of the vectors, and K (where $K \leq J$) is the rank of the approximated space.

The approximation is achieved by first determining an orthonormal basis for the instance space, and then removing those dimensions that have low singular values. *Singular Value Decomposition* (SVD) [16, 9] is normally used to perform the orthogonal decomposition, although other decomposition approaches, such as the ULV decomposition [3], can be used to replace SVD for this task. The SVD of a matrix $\mathbf{X}$ of I rows (i.e. instances) and J columns (i.e. attributes) can be expressed as:

$$\mathbf{X} = \mathbf{L}\,\mathbf{D}\,\mathbf{R}^{\top}$$

where $\mathbf{L}^{\top}\mathbf{L} = \mathbf{R}^{\top}\mathbf{R} = \mathbf{I}$ (the identity matrix). The orthonormal vectors of $\mathbf{R}$, called the right singular vectors, form an orthonormal basis for the rows of $\mathbf{X}$. The diagonal matrix $\mathbf{D}$ contains the singular values of $\mathbf{X}$, where the elements of $\mathbf{D} : d_1 \geq d_2 \geq \cdots \geq d_N > 0$, and $N \leq \min(I, J)$. A third matrix, $\mathbf{L}$, is also expressed, which forms an orthonormal basis for the columns of $\mathbf{X}$.

The sub-space approximation framework used by *CA* and *CACP* consists of two main routines: one that generates a mapping function between the original space and the transformed and approximated sub-space (Figure 1); and a routine that uses the mapping function to project instances from the original space into the new space [14, 15]. Data sets are presented to these routines as matrices, where each row of the matrix corresponds to an instance, and each column corresponds to one of the attributes of the dataset. The mapping function[1] consists of the basis $\mathbf{R}_{(K)}$ of the approximated sub-space, and a centroid, $\bar{\mathbf{y}}$. Instances, represented as vectors in the matrix $\mathbf{Y}$, are projected into the new space by translating them with respect to the centroid, $\bar{\mathbf{y}}$, and multiplying the translated vectors with the basis, $\mathbf{R}_{(K)}$. Thus, to determine a K-rank approximation of the dataset $\mathbf{Y}$:

1. Find the centroid vector $\bar{\mathbf{y}}$ for the training dataset $\mathbf{Y}$.
2. Translate the training dataset by the centroid vector into the matrix $\mathbf{X} = \mathbf{Y} - 1\bar{\mathbf{y}}^{\top}$.
3. Determine the basis $\mathbf{R}$ and the diagonal singular matrix $\mathbf{D}$ of $\mathbf{X}$ using singular value decomposition.
4. Select the K columns of $\mathbf{R}$ (or K rows of $\mathbf{R}^{\top}$) that correspond with the largest K singular values in the diagonal matrix $\mathbf{D}$.
5. Project the instances represented by the matrix $\mathbf{X}$ into the space characterised by $\mathbf{R}_{(K)}$, by multiplying $\mathbf{X}$ with $\mathbf{R}_{(K)}$.

Two algorithms have been developed based on the sub-space mapping approach described above. The first, *CA*, uses the function *generate_mapping* which ignores class information when generating the basis $\mathbf{R}$ from the training data. The second algorithm, *CACP*, exploits the class labels when determining the

[1] The details of these functions are described in greater detail in [14].

mapping function. The *generate_cpmapping* routine generates a single *prototype* point for each class, by finding the centroid of all the instances belonging to that class. Once all the prototype points have been found, they are used to generate the new basis, $\mathbf{R}$.

```
1  proc generate_mapping(Y, rank) ≡
2      y = get_centroid_vector(Y);
3      X = translate_data(Y, y);
4
5      [L, D, R] = SVD(X);
6
7      R_(K) = low_rank(D, R, rank);
8      map[basis] = R_(K);
9      map[centroid] = y;
10     return(map).
```

```
1  proc generate_cpmapping(Y, rank) ≡
2      y = get_centroid_vector(Y);
3      X = translate_data(Y, y);
4      P = get_class_prototypes(X);
5
6      [L, D, R] = SVD(P);
7
8      R_(K) = low_rank(D, R, rank);
9      map[basis] = R_(K);
10     map[centroid] = y;
11     return(map).
```

Fig. 1. The sub-space mapping algorithms, *generate_mapping* and *generate_cpmapping*, are used by *CA* and *CACP* respectively to map dataset $\mathbf{Y}$ to a sub-space of rank *rank*.

4 Experimental Design

Many machine learning systems incorporate, or utilise some form of dimensionality reduction to generate an optimal (or sub-optimal) subset of dimensions prior to induction. The sub-space approximation techniques described above project instances (represented as data points within some instance space) into a lower dimensional sub-space. To compare the benefits (in terms of predictive accuracy) of this approach with other attribute selection techniques, a suitable learning paradigm is required. Instance-based learning algorithms, which are sometimes referred to as Nearest Neighbour (NN) algorithms [6] are ideal, as the accuracy of these techniques degrades in the presence of irrelevant or redundant data [14]. They store and represent some or all the training instances as data points within a hyperdimensional instance space. The instance space is usually described by N dimensions, where each dimension corresponds to a single attribute of the dataset. New (unseen) instances are classified by determining their location within this instance space, and by identifying their nearest neighbour using some *distance* function. The class value of the nearest instance is then used to predict the class of the unseen instance.

To compare the effects of using correspondence analysis for dimensionality reduction with more traditional approaches to attribute selection, a wrapper based attribute selection method was implemented. The search method used was a stochastic search known as the *Monte Carlo* method [13]. This method was chosen as the number of search states visited can be controlled, and, unlike hill climbing approaches, it is not susceptible to local maxima [14]. It is also possible to show that as the number of states visited increases, so does the probability of finding an optimal solution [13]. This method searches for the best attribute subset by selecting a random subset and evaluating it. The evaluation

was performed using a leave-one-out cross validation with the nearest neighbour Euclidean distance learning algorithm on the training dataset.

A filter-based attribute selection method was also tested. The learning algorithm, *FNN* was implemented, which utilises the C4.5 decision tree learning algorithm [17] to identify relevant attribute subsets and remove the remaining attributes from the dataset. The modified dataset is then presented to a Euclidean nearest neighbour learning algorithm. C4.5 uses a *divide and conquer* approach to inducing decision trees, by recursively determining the attribute that best splits the data into homogeneously classified clusters of instances. As a consequence, many decision trees utilise a subset of the available attributes, which reduces the impact of irrelevant attributes on the target concept[2]. This behaviour has been exploited as an attribute selection mechanism in its own right, with the resulting attributes being tested with other learning algorithms [5].

5 Experimentation and Results

A 20-fold cross validation strategy was used to evaluate the performance of the learning algorithms on eleven numerical datasets (Table 1) from the UCI Machine Learning Database Repository [4]. Several of these datasets each contained an attribute corresponding to a unique identification value. These attributes were removed from the datasets to prevent them affecting the classification accuracy. For example, the *glass* dataset contains an ordered numeric identifier, which is highly correlated with the class (using Spearman's Rank Correlation, the coefficient is 0.958). To determine the lowest number of dimensions that achieve the highest accuracy, the *CA* and *CACP* algorithms varied the number of dimensions to approximate the sub-space for each dataset between 1 and n, where n was the total number of attributes available for the dataset. The results presented in the tables below refer to those tests that achieved the highest classification accuracy.

Table 1. UCI datasets used in this study.

balance	Balance Scale Weight & Distance	bupa	BUPA liver disorders
ionosp	JHU Ionosphere DB	glass	Glass Identification DB
pima	Pima Indians Diabetes DB	iris	Iris Plants DB
sonar	Sonar, Mines vs. Rocks	wine	Wine Recognition Data
wdbc	Wisconsin Diagnostic Breast Cancer	wiscon	Wisconsin Breast Cancer DB
wpbc	Wisconsin Prognostic Breast Cancer		

The results of the 20-fold cross validated tests for the five algorithms are given in Table 2. The results in the second column (*NN*) represent a baseline result, i.e. the result of the nearest neighbour algorithm when no dimensionality reduction

[2] The selection metrics utilised by decision tree learning algorithms will not necessarily select the optimal set of attributes [2].

technique is used. The wrapper method, *MC*, succeeded in reducing the number of attributes for ten of the eleven datasets. The number of attributes found for these datasets was typically half that of the original number of attributes. There was a significant increase in classification accuracy for the *iris* dataset (at the 5% confidence level) and *ionosp* dataset (at the 10% confidence level). However, there was a significant decrease in classification accuracy for the *pima* and *wiscon* datasets. No significant difference in classification accuracy was found between *NN* and *MC* for the remaining seven datasets. These results suggest that this wrapper algorithm can successfully reduce the number of attributes in most cases, with little or no loss in classification accuracy, and that in some cases the classification accuracy can increase.

Table 2. Classification accuracies for the UCI datasets for the learning algorithms tested. Results followed by † were significantly different at the 5% confidence level to the baseline (i.e. *NN*) result, whereas those followed by ‡ were significantly different at the 10% confidence level (using a one-tailed t-test in both cases). The number of dimensions selected for each dataset are given in parentheses.

	NN	MC	FNN	CA	CACP
bupa	61.98 (6)	↓ 60.38 (4)	61.98 (6)	61.98 (6)	61.98 (6)
ionosp	87.17 (34)	↑ 90.64‡ (14)	↑ 92.60† (9.6)	↑ 90.90† (22)	↑ 91.19† (11)
pima	70.99 (8)	↓ 67.96† (4)	70.99 (8)	70.99 (8)	70.99 (8)
sonar	85.96 (60)	↓ 83.68 (28)	↓ 82.32 (14)	↑ 86.96 (23)	↑ 86.00 (60)
wiscon	95.90 (9)	↓ 95.03‡ (5)	95.90 (6)	↑ 97.36† (6)	↑ 96.19 (3)
wdbc	95.40 (30)	↑ 96.11 (14)	↑ 95.43 (8)	↑ 96.65‡ (5)	↑ 96.29 (16)
wpbc	69.06 (33)	↑ 71.17 (15)	↑ 70.50 (14)	↑ 71.61† (16)	↑ 73.06 (15)
balance	78.10 (4)	78.10 (4)	78.10 (4)	↑ 78.12 (4)	↑ 88.95† (1)
glass	68.09 (9)	↑ 71.00 (5)	68.09 (9)	68.09 (8)	↑ 70.00 (8)
iris	96.16 (4)	↑ 98.13† (2)	↑ 98.13† (2)	96.16 (4)	↑ 96.70 (3)
wine	94.86 (13)	↓ 94.79 (7)	↑ 96.04 (4)	↑ 97.08‡ (6)	↑ 97.64† (6)

The filter method, *FNN*, succeeded in improving the classification accuracy with respect to that achieved by *NN* for five of the eleven datasets. The *iris* dataset is known to contain two relevant attributes (see Figure 2) and two irrelevant attributes [8]. The C4.5 decision trees utilised only the two relevant attributes, and thus *FNN* succeeded in successfully increasing the classification accuracy to 98.13%, whilst halving the number of dimensions used. All four relevant attributes in the *balance* dataset were successfully identified and utilised. Similarly, all the attributes found in the *bupa* and *pima* datasets appeared in the C4.5 decision trees, and as a result, there was no difference in classification accuracy or dimensionality for these datasets. Although there was a drop in classification accuracy for two of the remaining datasets, these results were not significant. The rejection of attributes had no effect on the results for the *wiscon* and *glass* datasets. This suggests that not all the attributes are required to represent the target hypothesis, and that the rejected attributes may be either irrelevant or redundant.

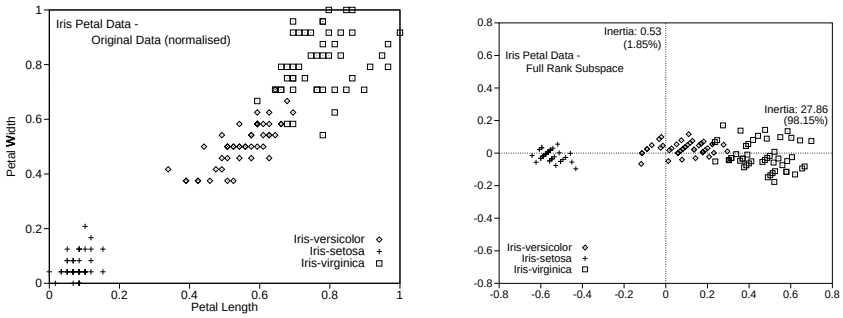

Fig. 2. Mapping the two most relevant attributes of the *iris* dataset into a two dimension sub-space.

Both *CA* and *CACP* reduce the number of dimensions required to represent the dataset for six of the eleven datasets. The effects of the two algorithms differed for the *balance*, *iris* and *sonar* datasets: *CA* failed to reduce the dimensionality of *balance* or *iris*; whereas *CACP* failed to reduce the dimensionality of *sonar*. As with *FNN*, neither dataset succeeded in reducing the dimensionality of the *bupa* or *pima* datasets. The sub-space mappings used by *CA* and *CACP* resulted in an increase in classification accuracy for most of the datasets, in addition to reducing the dimensionality. Both methods achieved higher accuracies than either the filter or wrapper methods for six datasets, but in most cases utilised more dimensions.

The result achieved by *CA* for the *balance* dataset suggests that when all the dimensions are present (i.e. no approximation is generated), the sub-space mapping may still affect the classification accuracy of the learning algorithm. This can be illustrated by examining the instance space for the *iris* dataset when only the petal attributes are used, and comparing it with a full rank (i.e. two dimension) sub-space generated by *CA* (Figure 2). In this case, the mapping function performs a rotation and a linear translation. The varying translation of each dimension has the effect of distorting the sub-space with respect to the original space, which is analogous to assigning relevance weights to each dimension.

All four methods (*MC*, *FNN*, *CA* and *CACP*) succeeded in reducing the number of attributes required for the majority of the datasets used in this study. The reductions in dimensionality for each dataset (given as a percentage of the original number of dimensions) are listed in Table 3. *MC* reduced the number of attributes by an average of 44.4%, and *FNN* by an average of 39.2%. In contrast, *CA* and *CACP* only reduced the dimensionality of the datasets by an average of 30.4%, and 36.4% respectively.

The results for the *iris* dataset suggest that the performance of *CA* and *CACP* may degrade in the presence of irrelevant attributes. To investigate this hypothesis, two further datasets were created, consisting of 100 instances each. The datasets each consist of two numeric attributes and a boolean class label. The

Table 3. The number of attributes used by each algorithm and the corresponding reduction in dimensionality (given as a percentage of the original number of dimensions).

	NN	MC		FNN		CA		CACP	
	attrs	attrs	% red.	attrs	% red.	attrs	% red.	attrs	% red.
bupa	6	4	33.3%	6	—	6	—	6	—
ionosp	34	14	58.8%	10	70.6%	22	39.3%	11	67.7%
pima	8	4	50.0%	8	—	8	—	8	—
sonar	60	28	53.3%	14	76.7%	23	61.7%	60	—
wiscon	9	5	44.4%	6	33.3%	6	33.3%	3	66.7%
wdbc	30	14	53.3%	8	73.3%	5	83.3%	16	46.7%
wpbc	33	15	54.6%	14	57.6%	16	51.5%	15	54.6%
balance	4	4	—	4	—	4	—	1	75.0%
glass	9	5	44.4%	9	—	8	11.1%	8	11.1%
iris	4	2	50.0%	2	50.0%	4	—	3	25.0%
wine	13	7	46.2%	4	69.2%	6	53.9%	6	53.9%
Average Reduction		**10 datasets 44.4%**		**7 datasets 39.2%**		**7 datasets 30.4%**		**8 datasets 36.4%**	

first dataset comprises of two linearly separable partitions. As *CACP* identifies and utilises class centroids, the second dataset contains four linearly inseparable partitions, two per class. Fifty additional irrelevant attributes were constructed, each containing a single random value for each instance. Various experiments were performed to investigate the behaviour of *CA* and *CACP* in the presence of irrelevant attributes. For each experiment, the two datasets containing the relevant attributes were combined with a random sample of irrelevant attributes, where the random sample increased in size from 0 to 50. Each dataset was then tested with *NN*, *CA* and *CACP*. This was repeated fifteen times for different combinations of irrelevant attributes.

Figure 3 illustrates the results obtained from experiments on the linearly separable dataset. The classification accuracy of all three algorithms falls exponentially, as the number of irrelevant attributes increase. The classification accuracies of both *NN* and *CA* are similar for datasets containing small numbers of irrelevant attributes. However, once the number of irrelevant attributes exceeds 14, the difference in classification accuracy between the two algorithms becomes small but significant (a one-tailed t-test shows significance at the 5% level), with *CA* achieving a slightly higher accuracy than *NN*. The number of dimensions used by *CA* varies as the number of irrelevant attributes in the dataset increases. There is no reduction in dimensionality for datasets with few irrelevant attributes. As the number of irrelevant attributes exceeds 8, the number of dimensions selected by *CA* increases slowly from 8 to 29.

The error rate of *CACP* is much lower than that achieved by either *CA* or *NN*. *CACP* achieved a mean accuracy of 74.74% with 49 additional attributes, whereas *CA* and *NN* achieved mean accuracies of 57.47% and 55.93% respectively. The presence of additional irrelevant attributes had little effect on the number of dimensions selected by *CACP* (three to five dimensions in most cases).

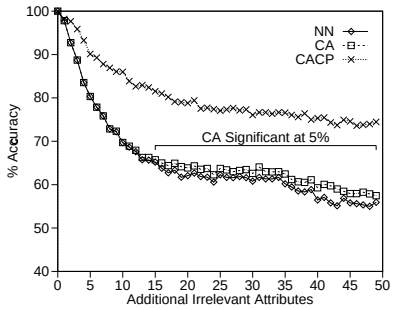

Fig. 3. The effects of additional irrelevant attributes for a linearly separable dataset on three learning algorithms.

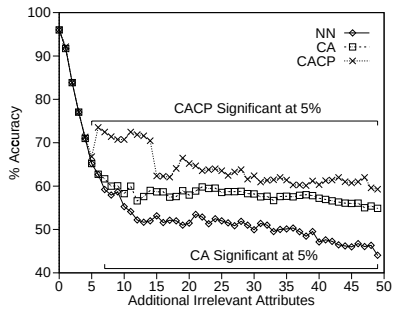

Fig. 4. The effects of additional irrelevant attributes for a linearly inseparable dataset on three learning algorithms.

The results for the three algorithms on the linearly inseparable datasets are shown in Figure 4. Although *CACP* achieved superior results for these datasets, the overall performance was much lower than with linearly separable data. However, this drop in accuracy for *CACP* may be due to the proximity of the centroids generated for each class. The initial drop in accuracy exhibited by *NN* is not surprising, as there is an additional boundary separating the points of the two classes, and a small number of points lie along this new boundary. However, the results after the addition of only a few attributes (e.g. 11 attributes) are little better than that achieved by pure chance, indicating that any contribution that the relevant attributes have to any classification hypothesis has been obscured by the effects of the irrelevant attributes. The results show an unusual increase in accuracy for *CACP* for datasets containing between 5 and 14 additional attributes. As yet, no explanation has been found for this behaviour.

The above experiments were repeated to investigate the behaviour of both *CA* and *CACP* in the presence of redundant attributes. In this case, 48 additional attributes were constructed. The values of the additional attributes were calculated in one of several ways: values were copied from one of the dimensions of the original datasets; or values were calculated by inverting one of the dimensions using the function $f(x) = 1 - x$. In addition, some of the attribute values were modified to introduce some variability to the similar dimensions. The function $f(x) = x \times (1 \pm rnd(\delta))$ was used, where $rnd(\delta)$ generates a small random number between 0 and δ; for this study we used $\delta = 0.05$.

All three algorithms achieved approximately 100% accuracy for the linearly separable dataset and 96.00% for the linearly inseparable dataset. A rank of two was always selected for *CA*, whereas the mean rank varied between one and four for *CACP*.

6 Conclusions

A number of attribute selection techniques that reduce the dimensionality of a dataset have been investigated in recent years. These techniques not only reduce the number of dimensions required to learn a hypothesis, but can result in an increase in classification accuracy. Various filter techniques have been proposed, but studies have shown that by including the learning algorithm in the selection process, better attribute subsets can be found. However, this wrapper approach does not scale up well to problems of more than a few attributes, due to the exponential increase in the size of the search space.

A technique known as Latent Semantic Indexing [7] has been used to reduce the dimensionality of large text-based corpora for some Information Retrieval systems. We have studied the underlying principles upon which LSI is based, and have developed two machine learning algorithms, *CA* and *CACP*, that combine these principles with a nearest neighbour learning algorithm. Both algorithms were found to reduce the number of dimensions required for the majority of datasets studied. In addition, the resulting classification accuracy increased for all but one of these reduced datasets. The techniques used by *CA* and *CACP* identified a new basis for a space that contained the instances in the training set, and then generated a lower dimension approximation to this space. The data points are represented by an attribute-by-instance matrix. Once this matrix has been decomposed, the rank of the matrix can be determined by the resulting diagonal matrix. This rank represents the number of linearly independent, orthogonal dimensions within a sub-space. Therefore, the addition of any duplicate attributes, or any linear combination of attributes will not result in an increase in rank, and so will be eliminated by the decomposition. If two or more attributes contain very similar but not identical values, then there will be additional orthogonal dimensions to express the slight deviations between them. Because the inertia of such dimensions will be small, a lower rank sub-space that excludes these dimensions will closely approximate the original sub-space.

CA and *CACP* appear to be very successful in removing redundant dimensions from the dataset. However, unlike many of the existing attribute selection techniques, they have little impact in reducing the effects of irrelevant attributes. The performance of the class projected variant *CACP* degrades at a slower rate than either *CA* or a simple nearest neighbour in the presence of irrelevant attributes. An investigation is required to determine the behaviour of this approach when used in conjunction with other attribute selection methods, such as weighted methods that identify and eliminate irrelevant attributes, but retain redundant ones. Further investigations are also required to compare this approach with constructive induction techniques, and more traditional statistical approaches such as Principal Components Analysis.

Acknowledgements

T.Payne acknowledges financial support provided by the UK Engineering & Physical Sciences Research Council (EPSRC).

References

1. D.W. Aha. Tolerating Noisy, Irrelevant and Novel Attributes in Instance-Based Learning Algorithms. *International Journal of Man-Machine Studies*, 36:267–287, 1992.
2. H. Almuallim and T.G. Dietterich. Learning With Many Irrelevant Features. In *Proceedings of the 9th National Conference on Artificial Intelligence (AAAI-91)*, pages 547–552. MIT Press, 1991.
3. M.W. Berry and R.D. Fierro. Low-Rank Orthogonal Decompositions for Information Retrieval Applications. *Numerical Linear Algebra with Applications*, 1(1):1–27, 1996.
4. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
5. C. Cardie. Using Decision Trees to Improve Case-Based Learning. In *Proceedings of the 10th International Conference on Machine Learning*, pages 25–32. San Francisco, CA:Morgan Kaufmann, 1993.
6. B. V. Dasarathy. *Nearest Neighbor(NN) Norms: NN Pattern Classification Techniques.* Los Alamitos, California:IEEE Computer Society Press, 1991.
7. S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.
8. R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.
9. M.J. Greenacre. *Theory and Applications of Correspondence Analysis.* London, UK:Academic Press, 1984.
10. G. 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. San Francisco, CA:Morgan Kaufmann, 1994.
11. P. Langley and W. Iba. Average-case Analysis of a Nearest Neighbor Algorithm. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 889–894. San Mateo, CA:Morgan Kaufmann, 1993.
12. P. Langley and S. Sage. Induction of Selective Bayesian Classifiers. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 399–406. Seattle, WA:Morgan Kaufmann, 1994.
13. H. Liu and R. Setiono. A Probabilistic Approach to Feature Selection - A Filter Solution. In *Proceedings of the 13th International Conference on Machine Learning*, pages 319–327. San Francisco, CA:Morgan Kaufmann, 1996.
14. T.R. Payne. *Dimensionality Reduction and Representation for Nearest Neighbour Learning.* PhD thesis, The University of Aberdeen, Scotland, 1999.
15. T.R. Payne and P. Edwards. Dimensionality Reduction through Correspondence Analysis. Technical Report AUCS/TR9910, Department of Computing Science, University of Aberdeen, Scotland., 1999.
16. W.H. Press. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1992.
17. J.R. Quinlan. *C4.5 Programs for Machine Learning.* San Mateo, CA:Morgan Kaufmann, 1993.
18. G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval.* New York: McGraw-Hill, 1983.
19. C. Wu, M. Berry, S. Shivakumar, and J. McLarty. Neural Networks for Full-Scale Protein Sequence Classification: Sequence Encoding with Singular Value Decomposition. *Machine Learning*, 21:177–193, 1995.