Modeling Local Nonlinear Correlations Using Subspace Principal Curves

Chandan K. Reddy* **and Mohammad S. Aziz**

Department of Computer Science, Wayne State University, Detroit, MI 48202, USA

Received 5 December 2009; revised 19 April 2010; accepted 10 July 2010
DOI:10.1002/sam.10086
Published online in Wiley Online Library (wileyonlinelibrary.com).

Abstract: While analyzing some of the complex real-world datasets, it is vital to identify local correlations in the subspaces. Some of the critical limitations of the subspace clustering techniques in identifying order revealing subspace correlation patterns motivate the need for more advanced subspace techniques. We formalize the problem of identifying local nonlinear correlations in high-dimensional data and build subspace models to capture such correlations. In this paper, we propose a new method for computing subspace principal curve models which can effectively capture these local patterns in the data. We demonstrate the results of the proposed method using several real-world datasets and highlight the advantages of our model compared to the other state-of-the-art techniques proposed in the literature. We also show the improved performance of the proposed algorithm in related problems such as missing data imputation and regression analysis compared to some of the state-of-the-art methods. © 2010 Wiley Periodicals, Inc. Statistical Analysis and Data Mining 3: 000–000, 2010

Keywords: subspace clustering; Apriori principle; biclustering; principal curves; nonlinear correlation

1. INTRODUCTION

Many practical applications produce data that contain thousands of records and several hundreds of features. Several high-dimensional data analysis techniques proposed in the literature do not reveal locally relevant correlations with respect to features and subsets of data points. In such high-dimensional feature spaces, it is critical to identify the subsets of data (and features) which form locally relevant subspace patterns and obtain local correlation information. This would enable the researchers to focus their attention on these local subsets and make it easy to identify the important and most informative aspects of the data. One of the many objectives of data exploration is to find correlations in the data, uncovering hidden patterns and trends in the data distribution, thus providing additional insights about the data [1,2]. However, it is a tedious task to identify the continuous structural patterns that capture the local correlations in the data within only a relevant set of features.

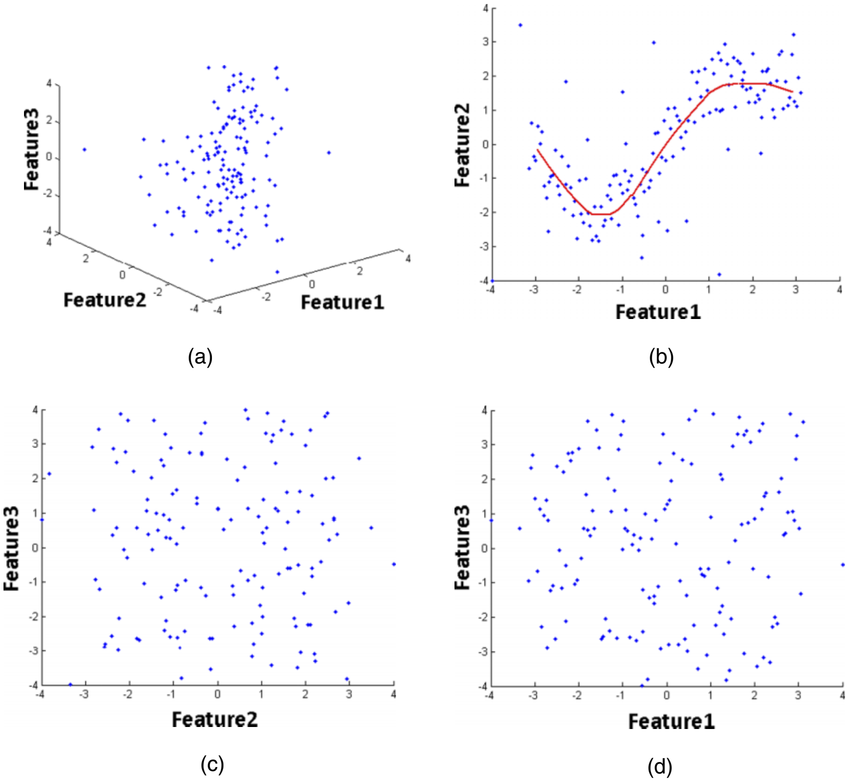
Though being successful in identifying local groupings, subspace clustering algorithms do not provide any continuous representation of local latent patterns in these subspaces. Two major drawbacks of subspace clustering algorithms that motivated the need for the proposed methodology are that the subspace algorithms do the following:

- simultaneously optimize the data and features to obtain localized clusterings and in this process, they tend to provide local dense clusters and do not preserve patterns (or correlations) in the data.
- yield a discrete set of clusterings which are hard to interpret. Especially when the end-user is looking for certain correlation patterns, it is important to extend the representation of these subspace clusters to continuous correlations in the data.

In this paper, we extend the notion of subspace clusters to '*subspace trends*' and develop a novel subspace principal curve method that captures local trends in feature subspaces. We extract subspace trends that represent the ordering and continuity information of the data points and have the potential to explain the linear or nonlinear correlations in the subspaces. These local correlation models developed here do not suffer from the above-mentioned problems of subspace clusters. They can be further analyzed with respect to their inherent properties such as coverage, continuity, length, ordering, and overlap. Analyzing individual trends can yield more information about the local structural arrangement of the data points along with some continuity information.

Let us consider a simple three-dimensional (3-D) dataset shown in Fig. 1. This dataset is generated as follows: Feature 2 value is a sine function of Feature 1 and some random noise is added to it. Feature 3 is randomly generated.

Correspondence to: Chandan K. Reddy (reddy@cs.wayne.edu)



(a)

(b)

(c)

(d)

Fig. 1 (a) No correlation in the original 3-D space. (b) An interesting correlation in only one of the 2-D subspaces. (c and d) No correlation in the other two 2-D subspaces.

Finally, ten noise points are included, where all the feature values are generated randomly in the range of −4 to 4. There is no prominent correlation that can characterize the data and reveal meaningful ordering in the original 3-D space. However, one of its subspaces (subspace which comprised of Features 1 and 2) contain a continuous correlation pattern which can be effectively represented using a subspace principal curve. Such local nonlinear correlations arise in many real-world datasets (as discussed in Section 5) and algorithms for extracting such complex subspace trends in the data have not been studied in the literature.

We propose an Apriori-based Subspace Principal Curve (SuPriC) algorithm to identify local subspace trends in the data. To the best of our knowledge, no work in the literature aims at computing the principal curves for the subspaces. The rest of this paper is organized as follows: Section 2 explains different methods proposed in the literature along with their shortcomings. Section 3 gives notations and some definitions required to comprehend the problem formulation. Section 4 describes the proposed SuPriC algorithm for building subspace principal curve models along with its computational complexity. Section 5 shows the experimental results of the proposed algorithm on several real-world datasets and compares the results with some of the state-of-the-art methods proposed in the literature. Finally, Section 6 concludes our discussion.

2. RELATED WORK

Our objective is to find a possible nonlinear correlation in the feature subspaces containing a sufficient number of datapoints in the dataset. Since the main objective here is to identify relevant and correlated subspaces, we discuss some of the dimensionality reduction, subspace clustering, principal curve and other approaches whose main goal is to compute an interesting subspace.

2.1. Dimensionality Reduction

Classical methods used for linear dimensionality reduction used in many practical applications are Principal Component Analysis (PCA) [3] and Multidimensional Scaling (MDS) [4]. Both these methods, though used widely in many applications, can only produce a linear mapping from a high-dimensional space into a low-dimensional space. Other methods such as Independent Component Analysis (ICA) [5] generate new feature spaces in which the data can be explained well. However, all these methods fail to find low-dimensional nonlinear correlations in the data since they always consider all the features and try to optimize a global objective measure that take into account all the features.

For many real-world problems, the underlying variability of the features creates a highly nonlinear inherent structure. For such datasets, nonlinear dimensionality reduction methods such as Locally Linear Embedding (LLE) [6], Laplacian Eigenmap (LE) [7], Isometric Mapping (Isomap) [8] and other manifold learning algorithms focus on preserving the inherent structural geometry of the data with respect to an intrinsic manifold. All these methods are geometry preserving dimensionality reduction methods which are able to identify the hidden structure of the entire dataset and preserve it in low-dimensional space. These are dimensionality reduction methods for linear and nonlinear embedding of the data points, but they can only interpret the hidden geometry of the data in a global sense. Although these methods succeed in identifying the global structure, they are essentially dimensionality reduction methods and are unable to extract the locally correlated structures that are present in the subspaces of the data. They only provide a guideline to generate a basis for some preliminary investigation about any positive correlations and cannot give any information about some of the subspace correlation patterns hidden in these datasets. To avoid the problems with high-dimensional feature spaces, some researchers use various feature selection methods that allow us to extract those informative features and separate them out from the redundant, repetitive, and noisy features [9]. These algorithms can only extract the relevant features, but they are not capable of selecting a subset of data points or provide any information about some of the local correlation structures of the data points within those feature sets.

2.2. Subspace Clustering

Given a set of data points in a multidimensional space, clustering [10] finds the most optimal partition of these points into groups such that the intracluster distance between the points is minimized and the intercluster distance is maximized. In many real-world problems, clusters may exist in different subspaces, comprised of different combinations of features [11]. Some of the data is correlated with respect to a particular subset of dimensions, and others are correlated with respect to a different subset of dimensions. In such cases, clustering is done in the subspaces rather than in the original spaces where cluster of points can possibly be found. The basic idea of the subspace clustering methods is to group the data into different partitions according to their connectedness or their correlation. Several subspace clustering algorithms have been proposed in the literature [12,13,14,15]. In order to avoid suboptimal cluster formation in high-dimensional feature spaces, subspace clustering algorithms find locally relevant clusters in a low-dimensional feature space. The subspace clustering is motivated by the fact that in high-dimensional space, the distance becomes meaningless and there is virtually no other point nearby. For obtaining the subspace principal curves, our method uses the Apriori principle [16]. We model the subspace trends in a bottom-up manner starting from two-dimensional (2-D) principal curves and extending them to higher-dimensional curves. There are some similarities between our approach and the CLIQUE algorithm [12] since both these methods use the Apriori principle. The key differences between the two algorithms are the following:

- The CLIQUE algorithm uses the concept of ‘density of a grid cell’ to make a k-dimensional cell desirable for further processing. Our algorithm uses the principal curve based objective measure for achieving this desirability (explained in Section 4). We claim that this measure is more suitable when the main goal is to identify the subspace trends (not subspace clusters).

- The result of the CLIQUE algorithm is a set of subspace clusters reported by a set of inequalities that describe the attribute ranges of the cells. On the other hand, our algorithm provides a continuous representation of the subspace trends that reveal better correlation information compared to the cluster representation.

We can compare these key differences to the methods proposed in the supervised learning literature. A rule-based classifier (such as RIPPER algorithm [17]) obtains a set of rules that make the distinction between different classes and a regression function models a set of continuous response values. Similarly, CLIQUE provides a set of rules to model the subspace clusters (in a discrete setting) whereas the subspace principal curves provide an optimal model for the correlation patterns in a continuous setting. In essence, most of subspace clustering algorithms (including the popular partitional approaches) optimize a criterion (such as centroid based distances or density based connectivity) which are not suitable to build subspace trends. For instance, the SUBCLU [18] algorithm looks for arbitrarily oriented subspace clusters and uses the monotonicity of the density connectivity whereas our algorithm finds the subspace principal curves and uses the antimonotonicity of a projection distance based objective criterion. SUBCLU can get more than one cluster in a subspace where as our goal is to characterize the data in terms of a continuous subspace correlations.

Bottom-up approaches [12,13,14] (such as CLIQUE) generally perform better [11] than the top-down approaches

[15,19,20] such as the PROCLUS method. PROCLUS [15] is a top-down subspace clustering algorithm that uses the medoid technique to find the appropriate sets of clusters and performs a locality analysis in order to find the set of dimensions associated with each medoid. In high-dimensional datasets, the bottom-up approaches converge much faster in the presence of low-dimensional clusters. In our framework, we used the bottom-up approach and incorporated the concept of Apriori principle since the goal here is also to find low-dimensional local correlations in a high-dimensional space.

2.3. Biclustering

Biclustering is a popular technique which allows simultaneous clustering of the rows and columns of a matrix. Given a set of m rows and n columns (i.e., an $m \times n$ matrix), biclustering algorithms find a subset of rows which exhibit similar behavior across a subset of columns, or vice versa. All of the biclustering methods strictly model linear correlations, but they fail to capture even negative or other complex correlations. Different models proposed in the literature [21,22,23,24] could not capture the subspace trends in the data, which is the main objective of this work. The proposed SuPriC algorithm can capture both linear and nonlinear (positive as well as negative) correlations present in subspaces. It can capture the correlated features using most data objects and is able to get trends that are present in the subspaces even when they are nonlinear and complex. The biclustering algorithms tend to unnecessarily split the data by following rigid constraints, whereas our method develops a single model to capture the subspace trend thus revealing an interesting correlation in a subspace.

2.4. Correlation Clustering

Correlation clustering [19,25,26] is a special type of clustering which defines the similarity between objects in terms of correlation between features, that is, it is a clustering approach which assigns two data points to the same cluster (no matter how far they are in the feature space) if they share the same correlation in the feature subspace. Most of these methods use eigendecomposition techniques and hence reveal a global linear fit of the data in the subspace. Some of these methods such as ORCLUS [19], 4C [25] use the density based clustering approach and assume that they are locally linearly but arbitrarily oriented. All these methods work in full-dimensional space to generate the initial orientation/cluster membership, whereas our method starts with a feature pair(subspace with only two features) and uses the Apriori principle to get feature subsets with more features.

2.5. Principal Curves

Principal curves are nonlinear summarizations of multidimensional data points represented by a smooth, one-dimensional curve. They are the one-dimensional representation of the data that are defined by a curve that passes through the most dense regions of the dataset, thus taking shape according to the distributions of the dataset. One of the pioneering works on principal curves was based on 'self-consistency' [27], that is, the curve should coincide at each position with the expected value of the data projecting to that position. Kegl *et al.* [28] improved the algorithm to achieve the minimum expected squared distance from points to their projection on the curve, which eliminates the estimation bias. A more probabilistic approach [29] defines principal curves as those minimizing a penalized log-likelihood measure. Although there have been several studies on improving the quality of the principal curves [30], none of them deal with fitting principal curves to the subspaces of features. Though principal surfaces which are multidimensional extensions to principal curves have been proposed in the literature, their poor interpretability in real-world problems pose a serious concern [31,32]. In this paper, we extended the concept of fitting principal curves to subspace principal curves, which can model the nonlinear subspace correlations in the data. For our work, we used the definition (and implementation) of the principal curves proposed by Verbeek *et al.* in [33] due to its simplicity and effectiveness in practice. To make this paper self-contained, we briefly describe the principal curve algorithm in the Appendix.

3. PRELIMINARIES

In this section, we will present the necessary notations and definitions that are required to comprehend our algorithm (Table 1).

DEFINITION 1: *(Softly Monotonic):* Let, X be a finite subset of $\Re$, $f\colon X \to \Re$ and $g_1\colon X \times X \to \{0,1\}$ where,

$$g_1(x_i, x_j) = \begin{cases} 1 & \text{if } x_i \le x_j \text{ and } f(x_i) > f(x_j) \\ 0 & \text{otherwise.} \end{cases}$$

and let $g_2\colon X \times X \to \{0,1\}$ where,

$$g_2(x_i, x_j) = \begin{cases} 1 & \text{if } \; x_i \le x_j \text{ and } f(x_i) < f(x_j) \\ 0 & \text{otherwise.} \end{cases}$$

Then function f is softly monotonic, if either

$$\textstyle\sum_{x_i, x_j \in X} g_1(x_i, x_j) < \kappa\,|X| \text{ or}$$
$$\textstyle\sum_{x_i, x_j \in X} g_2(x_i, x_j) < \kappa\,|X|.$$

Table 1. Notations used in this paper.

Notation	Description
n	Number of datapoints
m	Number of features
$Data$	The whole data matrix
F	Feature set $F = \{F_1, \ldots, F_m\}$
t	A candidate feature subset $t \subseteq F$
$Data(t)$	$Data(t) \subseteq Data$, where $Data(t)$ consists of data point associated with candidate feature set t
DP	Data Points $DP = \{DP_1, \ldots, DP_n\}$
X	Input dataset $X =$ (F,DP)
$F_i(DP_k)$	ith feature value of kth data point
f_i	Subset of features $f_i = (F_k, \ldots, F_*, \ldots, F_l) \subset F$
dp_i	Subset of data points $dp_i = (DP_k, \ldots, DP_*, \ldots, DP_r) \subset DP$
f^*	A curve representing principal or subspace principal curve
I	Indicator function

If the number of datapoints that violate the monotonicity is less than a threshold, then the function is called softly monotonic. Here, κ is a parameter that represents the lower bound on the fraction of datapoints to be considered and is less than 1. Note that we can consider the mapping function as the functional presentation of the corresponding subspace principal curve. In such a case, the antimonotonicity principle (which is needed to reduce the search space) holds, if that function (subspace principal curve) is softly monotonic.

DEFINITION 2: *(Monotonic Pair):* Let, $F_i(DP)$ denotes the set of values for feature F_i in dataset DP. A feature pair P $= (F_i, F_j)$ for a dataset (DP) is said to be a monotonic pair, if and only if, $F_i(DP) = f(F_j(DP))$ is softly monotonic where f is the mapping function from feature $F_j(DP)$ to $F_i(DP)$ for the data points in the dataset $\{DP\}$.

A monotonic pair is the basic component of a monotonic set. Any set of features is a monotonic set if all its proper subsets with cardinality ≥ 2 are also monotonic set. The objective of our algorithm is to identify all such monotonic sets in which there is a desired subspace principal curve. Later in this section, we prove that antimonotonicity holds for the monotonic sets, which will be used as the basis of our algorithm.

DEFINITION 3: *(Principal curve):* A curve f^* is called a principal curve of length L for X if f^* minimizes $\Delta(f)$ over all curves of length less than and equal to L where, $\Delta(f) = E[inf\,|X - f(t)|^2]$ = E[$|X - f(t_f(X))|$].

Here, $t_f(x)$ denotes the parameter value of t for which the distance between x and $f(t)$ is minimized [28]. We will use a k-segments algorithm [33] to compute the principal curve in different subspaces. Here, k is the number of segments that are joined to obtain the continuous nonlinear principal curve. If the nonlinearity of the principal curve is not high, then a small value of k is sufficient, but if the principal curve is highly nonlinear, k should be increased to reflect the nonlinearity.

DEFINITION 4: *(Desirable Principal curve):* A desirable principal curve is a principal curve for $X_1 \subseteq X$ if and only if $\|X_1\|>\delta\|X\|$ and $\Delta(f) = E[inf\,|X_1 - f(t)|^2] = E[|X_1 - f(t_f(X_1))|] \leq \tau$.

Here, τ is a parameter describing the upper bound of the objective function, and δ is a parameter describing the lower bound on the fraction of the dataset to be considered. A subspace principal curve is a principal curve when only a subset of features is considered.

DEFINITION 5: *(Subspace Principal curve):* A curve f^* is called a subspace principal curve of length L for X if f^* minimizes $\Delta(f)$ over all curves of length less than and equal to L where $\Delta(f) = E[inf\,|X - f(t)|^2] =$ E[$|X - f(t_f(X))|$] calculated over feature subset $M \subset F$.

DEFINITION 6: *(Desirable Subspace Principal curve):* A desirable subspace principal curve is a subspace principal curve if and only if $X_1 \subseteq X$ and $\|X_1\| > \delta\|X\|$ $\Delta(f) = E[inf\,|X_1 - f(t)|^2]$ = E[$\|X_1 - f(t_f(X_1))|$]$\leq \tau$.

For a curve f^* to be a desirable subspace principal curve, it must obey the constraint for both the subspace principal curve and the desirable principal curve. Immaterial of the scattering or orientation of the data in the subspaces, we will get a principal curve in every possible subspace. However, most of them will be noisy curves which do not convey any information unless there is a strongly correlated trend present in the subspace. We defined a desired subspace as a subspace where our objective function for a desirable subspace principal curve is lower than a certain threshold. We also show that the desired subspace follows antimonotonicity, which significantly reduces the complexity of computing these subspace principal curves. Theorem 1 indicates that the desired subspaces follow the Apriori principle. This is a vital component of our algorithm since antimonotonicity can significantly reduce the search space.

Apriori principle for desired subspace. If a feature set $M \subset F$ contains a desired subspace principal curve, then all of its subsets $M' \subset M$ must also contain a desired subspace principal curve.

THEOREM 1: Desired subspaces follow the Apriori principle, that is, M is a desired subspace if and only if all subsets of M are also desired subspaces.

Proof: As the feature space considered here is a metric space, $\Delta(f) = E[inf\,|X_1 - f(t)|^2] = E[|X_1 - f(t_f(X_1))|]$ in feature space M must be greater than $\Delta(f) = E[inf\,|X_1 - f(t)|^2] = E[|X_1 - f(t_f(X_1))|]$ in feature space $M_i \subset M$. So if $\Delta(f) = E[inf\,|X_1 - f(t)|^2] = E[|X_1 - f(t_f(X_1))|]$ $\leq$ threshold in feature space M, then $\Delta(f) = E[inf\,|X_1 - f(t)|^2] = E[|X_1 - f(t_f(X_1))|] \leq$ threshold in feature space $M_i \in M$, that is, if in any feature space the principal curve has a lower expected value than a certain threshold, then in all of its subspaces, the principal curve must have an expected value less than the threshold, thus holding the Apriori principle. In other words, it monotonically decreases with the increase in dimensions. ■

3.1. The Objective Function

The optimal principal curve for a given subspace is computed using $\Delta(f)$. However, it is not possible to determine a threshold based on which we can conclude whether the subspace principal curve is optimal for characterizing the subset of the data. To obtain a desirable subspace principal curve, we propose the following objective measure which is impartial to the number of dimensions and the number of data points:

$$p = \frac{w \times \mathrm{ssd} + (1 - w) \times \mathrm{len}}{\sqrt{\mathrm{nd}} \times \mathrm{nr}} \tag{1}$$

where ssd refers to the sum squared projection distances, len indicates the length of the principal curve, nd represents the number of dimensions and nr represents the number of data points. For each data point $x_{i,j}$, let p($x_{i,j}$) be its projection point on the principal curve. The L_2 norm based sum squared projection distance (ssd) is defined as follows:

$$\mathrm{ssd} = \sum_{i=1}^{m'} \sum_{j=1}^{n'} (x_{i,j} - p(x_{i,j}))^2 \tag{2}$$

where m' and n' are the number of data points and the number of features in the subspace that is being considered for fitting the principal curve. When $\Delta(f)$ is optimized, we get a principal curve that is described by the sequential projection points in the given subspace. Thus, len is calculated by the summation of the distance between two consecutive points. Since more number of data points will result in greater values for ssd and len, the result must be normalized by the number of data points (nr). The weight parameter w determines the trade-off between the importance given to ssd and len values. Here w is the parameter that signifies importance of the length of the principal curve and the projection distance. In our experiments, we fixed $w = 0.8$, thus giving more emphasis on the projection distances. This is due to the fact that len has less discriminatory power compared to ssd in terms of estimating a good subspace principal curve.

The numerator value of p also shows the Apriori principle since it is a combination of distance and length in a metric space where the length is calculated as the sum of distances between consecutive points in the curve. These two measures always increase with the increase in dimensionality. In a given space, if an optimal principal curve is generated using the weighted combination of ssd and len, then the distance of the points from the principal curve should be greater than the distance of those points from a curve in any of its subspace, even if the curve is just the projection of the mentioned principal curve in that subspace. Since the threshold for the optimal curve for a multidimensional space is fixed to be a constant value, we used $\sqrt{nd}$ in the denominator to nullify the effect of '*the density divergence problem*'. The density divergence problem [34] indicates that it is difficult to set a global threshold when the dimensionality varies because the data are naturally far apart in high-dimensional spaces. With higher dimensions, the distance between data points tends to get higher, which will affect the objective function and the corresponding threshold values. In order to nullify this effect, the square root of the number of features is used, which will reasonably reflect the increased distance in higher dimensionality. We used the fact that, with the same distance in every dimension, a d-dimensional distance is $\sqrt{nd}$ times the one-dimensional distance.

It should be noted that the use of the Apriori principle will significantly reduce the search space. In many real-world applications, the feature correlations of interest are typically low-dimensional and hence, there is no need to traverse the entire feature combination lattice and only a small portion of the entire lattice will be explored.

4. MODELING NONLINEAR SUBSPACE CORRELATIONS

In this section, we will describe the details of the proposed SuPriC algorithm. We will also analyze the effect of parameters and the computational complexity of the proposed method.

4.1. SuPriC Algorithm

SuPriC is a bottom-up approach which uses each feature pair as a building block for computing higher-dimensional subspaces. First, subspace principal curves for each feature pair are generated and the corresponding objective function values are calculated. From these objective function values of all the base cases, a threshold for eliminating some of the base cases is determined. For the feature pairs which are selected as good candidates, the algorithm removes the data points that are significantly distant from the corresponding subspace principal curve. Consequently, two-dimensional candidates are combined to generate higher-dimensional subspaces. If the Apriori criterion is satisfied, a principal curve is generated in that subspace with only those data points that are in both the candidate cases. If the objective function from the newly generated higher-dimensional subspace is less than the threshold, the new subspace is included in the final solution. For every subspace thus generated, distant data points are removed, and the remaining points are associated with that subspace in the final output. In this bottom-up approach, the higher-dimensional subspace principal curves are thus computed starting with the basic 2-D ones. Algorithm 1 describes our Apriori-based approach for identifying subspace principal curves.

4.1.1. SuPriC first calculates principal curves in 2-D subspaces

For each of the feature pairs, the optimal subspace principal curve is generated. This is the most critical and computationally intensive part of our algorithm. These base cases will provide an estimate of acceptability threshold FT. At this point, the objective function value p for each feature pair is computed. There is no such ideal threshold value since it depends on the particular dataset as well as the range of the feature values. Hence, the threshold value is set as $FT = \text{mean}(P) - 2 \times \text{stdev}(P)$, where P is an array of p values. This threshold value can identify feature pairs that are significantly better than the other feature pairs. If the data are really scattered in all the feature spaces, then we will not have any feature pair which satisfies $p < FT$. However, if some feature pair has a good subspace principal curves then the value of its p will be lower than FT. In this manner, FT is automatically calculated from the data and is not a user-defined parameter.

4.1.2. Finding the desirable feature pairs and corresponding data points

A subspace principal curve is desirable when the objective function value is lower than the threshold FT. We select all the feature sets with cardinality 2 based on the acceptability criteria $p < FT$ and, correspondingly, the data points for the corresponding feature pairs that form the subspace principal curve in that feature subspace.

Algorithm 1 SuPriC($Data$, w, DT)

```
Input: Data matrix ($Data$)
       Weight parameter for objective function($w$)
       Weight parameter for Data Threshold ($DT$)
Output: Set of Subspace Principal curves ($PC\_List$)
Procedure:
$F \leftarrow$ features(Data),$P \leftarrow \emptyset$
$c \leftarrow 2$
for each $\{u,v\} \subseteq F$ do
    $[sqd, len] \leftarrow$ PrincipalCurve(Data,$(u,v)$)
    $ssd \leftarrow sum(sqd)$
    $p[(u,v)] \leftarrow (w \times ssd + (1-w) \times len)/(\sqrt{2} \times |Data|)$
    $P \leftarrow P \cup p[(u,v)]$
end for
$FT \leftarrow$ mean $(P)$-2× std($P$)
for each pair of features $(u, v)$ do
    if $p[(u,v)] \leq FT$ then
        $t \leftarrow \{u, v\}$
        $Data(t) \leftarrow remove\_Outlier(Data, sqd, DT)$
        $PC\_List(c) \leftarrow PC\_List(c) \cup (Data(t), t)$
    end if
end for
repeat
    $c \leftarrow c+1$
    generate candidate set $C$ by including all $t$ such that
    $t = t_i \cup t_j$ and $Data(t) = Data(t_i) \cap Data(t_j)$ where
    $t_i, t_j \in PC\_List(c-1)$ and $\|t_i \cap t_j\|$ =c−2 and $(t - t_i)$
    $\times (t - t_j) \subset PC\_List(2)$
    for each $t \in C$ do
        $[sqd, len] \leftarrow$ PrincipalCurve(Data(t),t)
        $ssd \leftarrow sum(sqd)$
        $p[t] \leftarrow (w \times ssd + (1-w) \times len)/\sqrt{c} \times |Data(t)|$
        if $p[t] > FT$ then
            Remove $t$ from $C$
        end if
    end for
    for each desirable $c$-feature combination $(F_*)$ do
        $Data(F_*) \leftarrow remove\_Outlier(Data(F_*), sqd, DT)$
        $PC\_List(c) \leftarrow PC\_List(c) \cup (Data(F_*), \{F_*\})$
    end for
    $PC\_List{=}PC\_List \cup PC\_List(c)$
until $PC\_List(c) = \emptyset$
Return $PC\_List$
```

4.1.3. Finding the desirable feature sets of higher-dimensional subspaces

Using the desired feature space of cardinality 2, we use the Apriori principle to find the desired subspace principal curves for the subspace of cardinality 3. A 3-D subspace will be considered for generating subspace principal curve only if all of its lower-dimensional subspaces contain desirable subspace principal curves. In this new subspace, only the points that are not removed while selecting its subspaces are considered for further processing. In this manner, c-dimensional subspaces are iteratively generated from $c-1$ dimensional subspaces. Note that, when a c-dimensional candidate is generated from two $c-1$ dimensional candidates, $c-2$ features must be common amongst them (line 22). Note that $Data$ refers to the complete data matrix, $Data(t)$ refers to the data points that belong to the principal curve in feature subset t. When two feature subsets of $c-1$ dimensions are joined to generate a candidate feature subset of dimensionality c, the associated datapoints of the new feature subset will be the intersection of the data points associated with both the participating feature subsets (line 22). $PClist(c)$ will contain all the selected subspaces of cardinality c. When we cannot generate any new subspace of cardinality c from the $c-1$ subspaces, the new $PCList(c)$ remains empty, and we will output the $PCList$ which is the union of all the $PCList(i)$ where i varies from 2 to $c-1$ (lines 20–36). At each stage, whenever we select a feature subset as a desirable subspace, we remove the data points that are significantly distant from the curve, thus, the associated datapoints for that desirable subspace is reduced and relevant.

4.2. A Note on Parameters

The parameters used in our algorithm have predefined values which work well on a wide variety of datasets thus keeping the user involvement minimal. The parameter w was set to 0.8 for all our experiments, indicating that the length of the curve does not play a vital role in determining the quality of the curve in these datasets. The weight parameter DT for specifying a threshold value is required for removing outliers in the subspaces. Since there is no standard definition for a subspace outlier, we will remove the points that are distant from the principal curve. The parameter DT helps us to quantify the 'significantly distant' points and is calculated as mean + 2× standard deviation of the distances from the data points to the principal curve.

4.3. Computational Complexity

The running time of the SuPriC algorithm mainly depends on the principal curve algorithm whose complexity is $O(kn^2)$. Since there are $m(m-1)/2$ possible feature pair combinations and each evaluation needs $O(kn^2)$ time, the for loop (lines 6–11) has a complexity of $O(kn^2m^2)$ where n is the number of data point and m is the number of features. Line 12 will take $O(m^2)$ time since $O(m^2)$ objective functions are to be computed. Since the principal curve generation step already computed all the distances from the data points to the curve, the removal of outliers will consist of calculating the average and standard deviation of those distance values and therefore costs $O(n)$ processing time in total. Therefore, the for loop (lines 13–19) iterates $O(m^2)$ times in which the removal outliers (line 16) needs O(n) time in every iteration, thus taking $O(m^2n)$ time. The complexity of the next loop (lines 20–36) is motivated by the fact that in most cases, the number of desirable feature pair combinations reduces dramatically. If the total number of acceptable combinations is l and the number of new combinations with $k+1$ features is reduced by a factor of w (i.e., the branching factor is w), then the total number of acceptable feature sets of all levels is $(lw-1)/(w-1)$. Since l is not too large compared to w, this is a relatively small number. The removing outlier part has the complexity of $O(kn'^2)$ where n' is the number of acceptable data points. As the feature subset grows, the n' becomes smaller. Therefore, the overall complexity of the SuPriC algorithm is $O(kn^2m^2)$ seems to be high since this is both quadratic in terms of the number of data points and the number of features. However, it is important to note that no other algorithm addresses the issue of modeling nonlinear correlations in the subspaces. The main reason for this quadratic complexity is that we must check every possible feature pair combination, and there is no other way to get around this issue. However, for achieving scalability, we propose a simple approximation scheme which makes the algorithm work much faster. While calculating the objective function value of all the feature pairs, we can bypass the k-segments algorithm (which uses PCA). Rather, we can partition the data based on one feature value into k partitions (assuming k segments) and then use a linear regression in those k partitions. The objective function is the sum of those residuals. We can use the k-segments procedure in the subsequent stages. Thus, the complexity for finding the objective value for each pair becomes $O(knm^2)$ which will also be the overall complexity of the algorithm. This will be scalable to large-scale datasets since it is only linear in terms of the number of data points.

5. EXPERIMENTAL RESULTS

The proposed method was used to identify significant subspace principal curves in several real-world datasets. Since there is no prior work for computing nonlinear correlations in subspaces, we compared our work with different subspace clustering and feature correlation techniques. The code for the SuPriC algorithm was written in MATLAB Version 6.5 and the experiments were run on Pentium Dual Core 2.8 GHz machine.

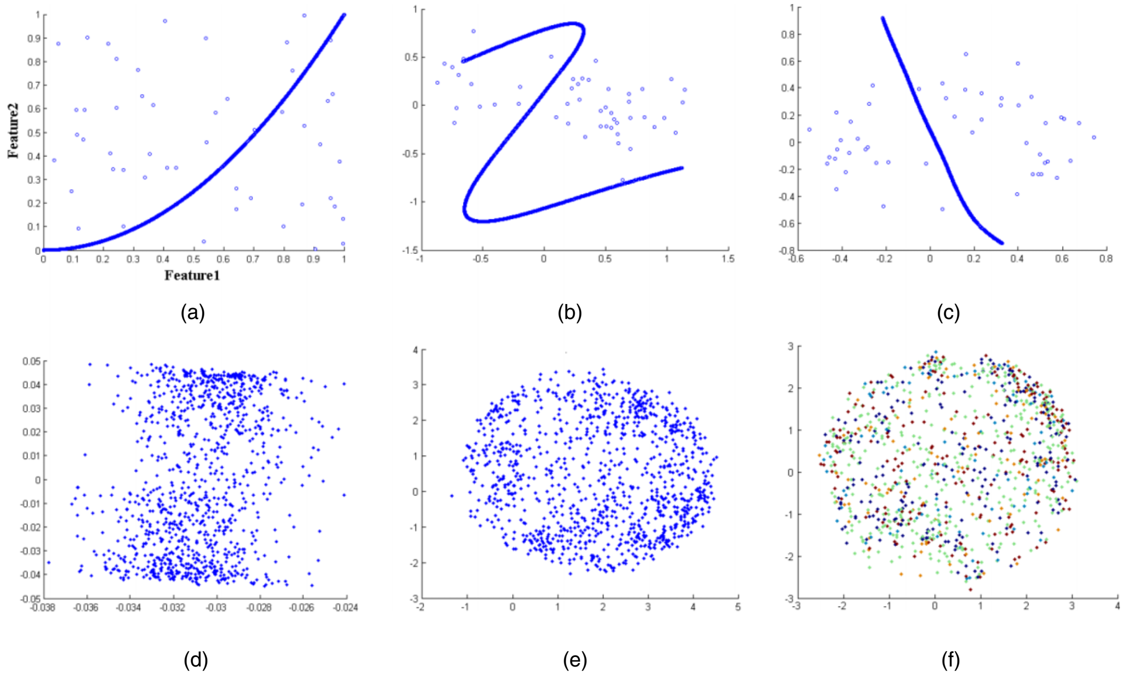
5.1. Synthetic Datasets

Our algorithm was tested successfully on various synthetic datasets that inherently contain embedded *subspace trends*. Several datasets were created with various embedded patterns hidden in the original global data space, and the algorithm was successfully able to fit the subspace principal curves. We chose to demonstrate the results on a representative synthetic dataset. We generated a synthetic dataset with 1050 × 58 dimensions. The first 1000 data points were generated in such a way that their first five features have the desired correlation. This is done by a parameter t which was randomly generated between 0 and 1. The five features were generated by the following components: $[t\ t^2\ sin(7t)\ 5t\ 3t]$. Three more features were generated using a second parameter t' with the component $[t'\ 5t'\ t'^2]$. The other 50 features are randomly generated noise. Fifty random noise points were also added. We expected ten subspace principal curves with two features and three features each, five subspace principal curves with four features, and one subspace principal curve with five features, a total of 26 subspace principal curves. We can also see four subspace principal curves (with the additional three correlated features) will make the number of possible subspace principal curves to 30. Our algorithm was able to identify all the desired principal curves that expressed the underlying correlation. We used the subspace clustering algorithms to investigate whether they can identify those subspaces. We found that they often miss the desired subspaces (see Table 2). This is mainly due to the fact that the goal of the subspace clustering algorithms is to partition the data points using some sort of distance (or density) criterion in the subspace. However, in our case, data is continuous along a nonlinear curve in a number of subspaces and do not necessarily form a dense cluster within a subset of data points. We can also see that, in such cases, subspace clustering tends to partition the data points and for each partition it reports some of the correlated features along with a few noisy features which aided to partition the data.

In panels (d)–(f) of Fig. 2, we can see that the embedded correlation in this data is not revealed by the traditional methods for dimensionality reduction such as Principal Component Analysis (PCA), Multidimensional Scaling (MDS), or other subspace clustering method (PROCLUS) [15] since they try to preserve as much information as possible (in terms of correlation) considering all the noisy features as well. Our method successfully extracted all the nonlinear subspace correlations present in the data, some of which are shown in panels (a)–(c) of Fig. 2.

Table 2. Comparison of Precision-Recall Statistics for synthetic dataset shown in Fig. 2.

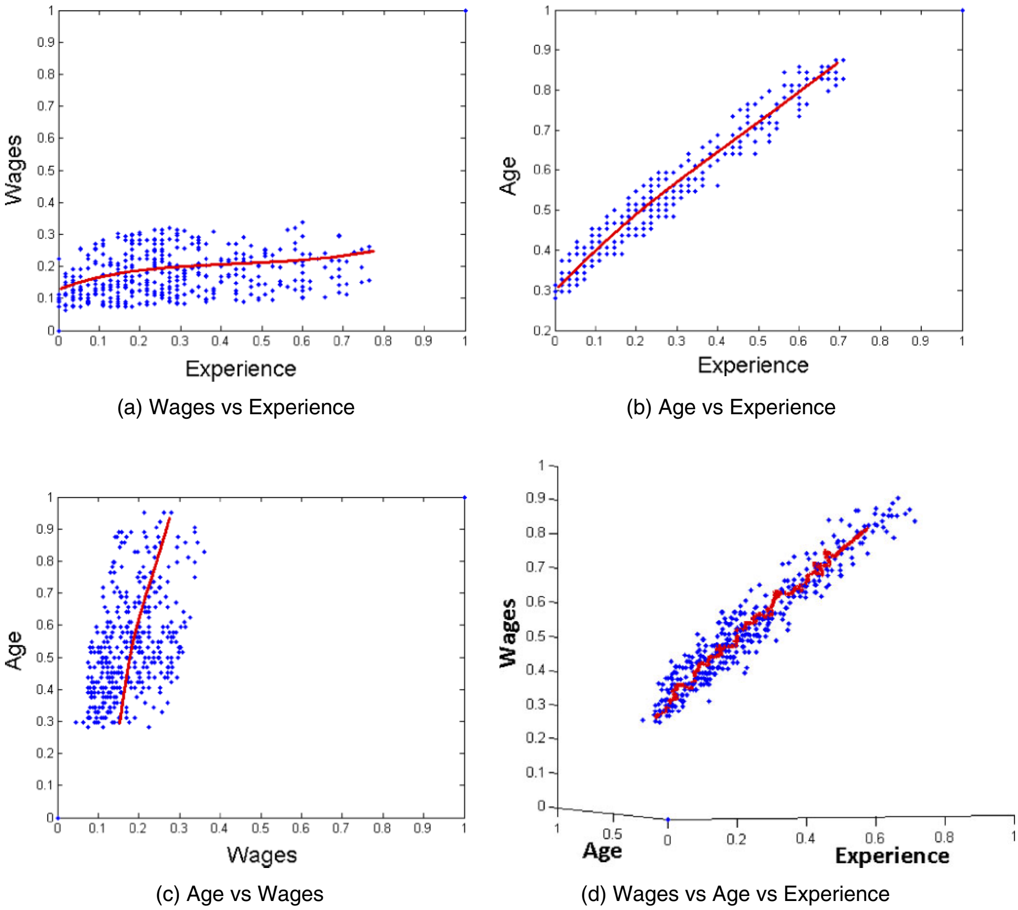
	CLIQUE [12]	SUBCLU [18]	PROCLUS [15]	SuPriC
Precision	50%	20%	40%	**93.75%**
Recall	30%	25%	23.33%	**100%**
F-measure	0.375	0.2222	0.2947	**0.9677**



(a) (b) (c)

(d) (e) (f)

Fig. 2 Subspace nonlinear correlations in the synthetic data successfully extracted by the SuPriC method. (a) Features 1 and 2, (b) first five features, (c) Features 6–8. Results from other methods (d) PCA, (e) MDS, (f) PROCLUS do not highlight the presence of any subspace correlation.



(a) Wages vs Experience

(b) Age vs Experience

(c) Age vs Wages

(d) Wages vs Age vs Experience

Fig. 3 Results on the Wages dataset. (a–c) 2-D feature combinations obtained by the SuPriC algorithm, (d) 3-D subspace principal curve that captures the correlation of the three attributes.

5.2. Real-World Datasets

Experiments were conducted using three machine learning datasets and several biological datasets.

(1) *Wages dataset*: The wages dataset contains the statistics of the determinants of Wages from the 1985 Current Population Survey. It contains 534 observations on 11 features sampled from the original Current Population Survey of 1985 and can be downloaded from the StatLib Data archive.[1] Out of these 11 features, four are numerical [EDUCATION: Number of years of education, EXPERIENCE: Number of years of work experience, WAGE: Wage (dollars per hour), and AGE: Age (years)]. The other seven features are categorical, which are then converted to the corresponding numerical values. We found that AGE and EXPERIENCE are the most correlated features based on our objective function value. Only 44 data points were eliminated from the principal trend. We also obtained a subspace principal trend with EXPERIENCE versus WAGES and with WAGES versus AGE. Since all of its subsets contain principal trends, we combined all the three features and identified a subspace trend with WAGES, EXPERIENCE, and AGE, containing 367 data points (see Fig. 3). This result is interesting because it gives some evidence that the pattern that SuPriC finds is meaningful and not just a spurious correlation.

(2) *Breast Cancer Dataset*: The Wisconsin Diagnostic Breast Cancer (WDBC) dataset[2] from the UCI Machine Learning Repository contains 32 features computed from a digitized image describing the characteristics of the cell nuclei present in the image with 569 data points. Excluding patient ID and diagnosis (class label) columns, we used only 30 features for our analysis. The 30 features contain three sets of attributes in which statistical values of some

[1] http://lib.stat.cmu.edu/datasets/CPS_85_Wages.

[2] http://archive.ics.uci.edu/ml/datasets.

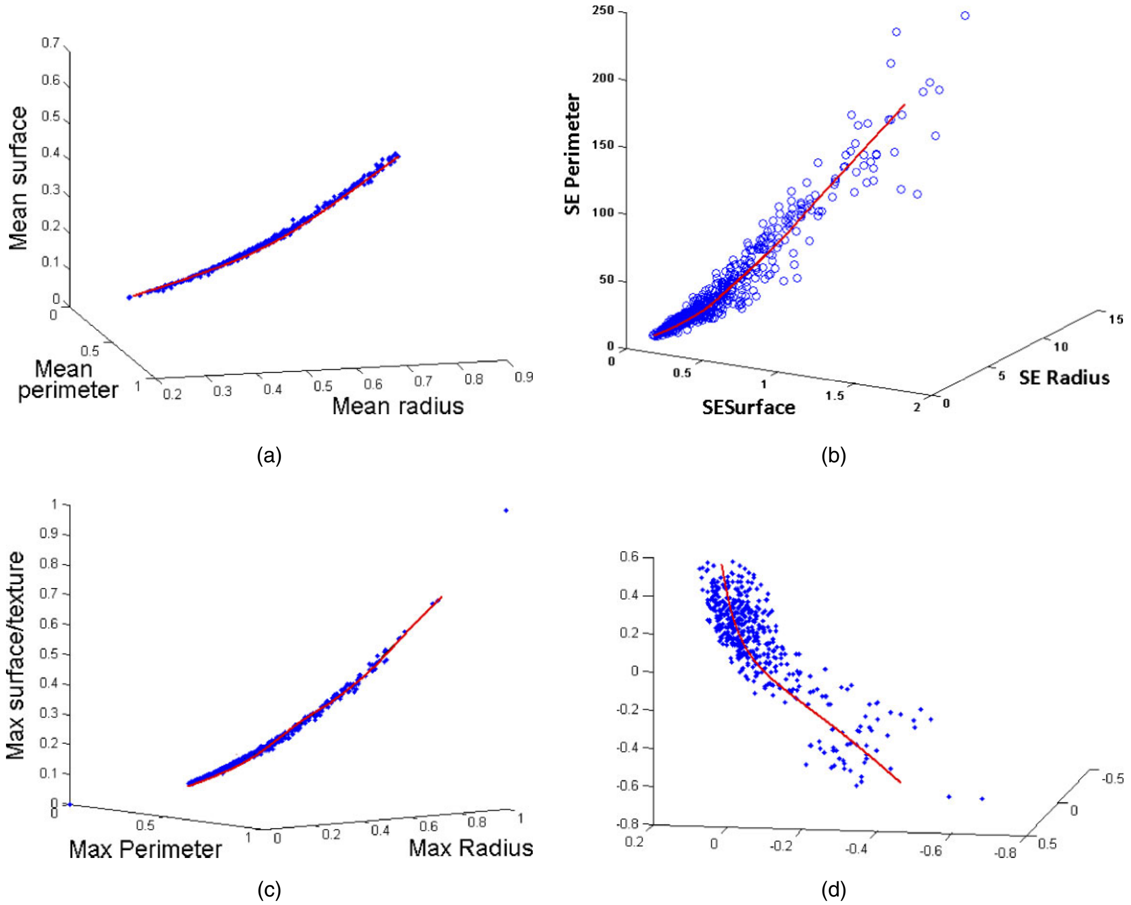


Fig. 4 Results on WDBC dataset. (a) A nonlinear correlation between mean perimeter, mean radius, and mean surface, (b) a nonlinear correlation between SEperimeter, SEradius, and SEsurface, (c) a nonlinear correlation between Max radius, Max perimeter and Max surface (d) a 9-D nonlinear correlation between all the features described in (a–c) projected onto a 3-D space using MDS.

tumor cell properties are stored. The first 10 features correspond to the mean values, the next 10 features correspond to the standard error values and the last 10 features correspond to the maximum values of the tumor properties such as radius, texture, perimeter, surface smoothness, etc. In this dataset, we see that mean radius and mean perimeter are the highly correlated features. Here, our algorithm obtained meaningful patterns, and the plots of the most significant 3-D subspace trends are shown in panels (a) and (b) of Fig. 4. We also found that there is a nine-dimensional (9-D) subspace trend obtained by combining the feature sets $(1, 3, 4)$ $(11, 13, 14)$ and $(21, 23, 24)$, indicating that the mean, standard error and maximum values of all these features are highly correlated compared to other sets of features. In panel (c) of Fig. 4, we show the result of multidimensional scaling (MDS) to reduce the dimensions of a 9-D subspace trend onto a 3-D plot. This particular result provides more insight about the *nonlinear subspace correlations* present in the data which are not revealed by any of the current techniques.

(3) *NBA dataset*: The NBA dataset[3] contains 28 features about 231 players from various teams playing in the NBA. There are many correlations among these features. Our method was able to find several subspace correlations. A few of them are listed below:

- The number of games played is correlated with the number of minutes the player gets per game, which means that dependable players play most matches and also get more time compared to others.
- The number of points accumulated is directly correlated with the two point scores, which means that two points are the bulk of all the points that are being scored.

[3] http://sports.espn.go.com/nba/teams.

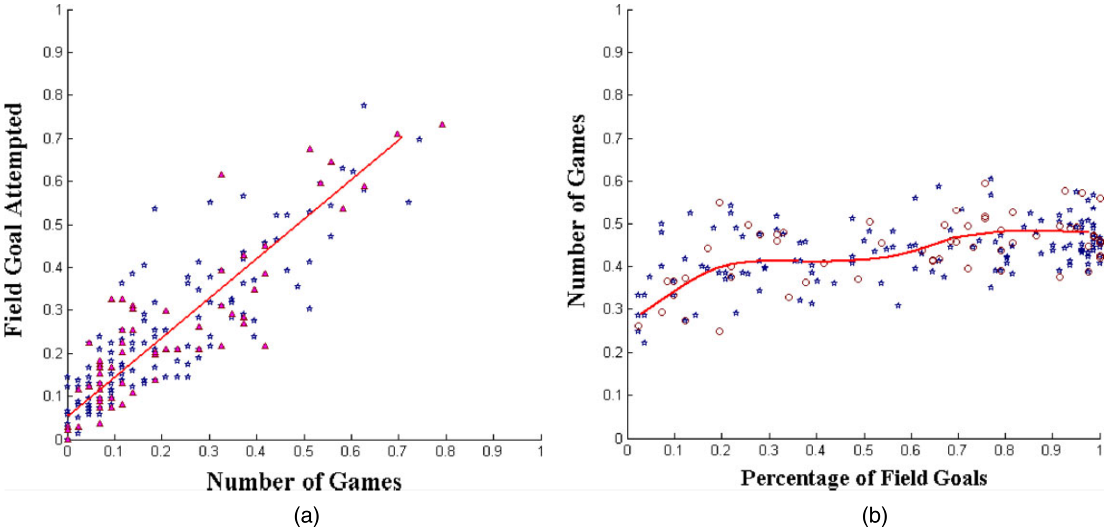


Fig. 5 Two subspace principal curves sharing a common feature ‘number of games’ and almost 70% of the data.

- In fact, we obtained a set of features that are strongly correlated: ‘total points scored’, ‘two points attempted’, ‘two points scored’, ‘two points percentage’, ‘points per shot’, and ‘adjusted field goal percentage’ were strongly correlated. Hence, any of the subsets of this feature set are also strongly correlated.
- The total minutes played per game tends to get higher turnover (the amount of occurrence a player loses the possession of the ball to the opponent) per game, which means that, irrespective of the player, time of stay in the game gives a higher probability of losing the ball.
- The number of games played is directly correlated with the percentage of field goals scored.

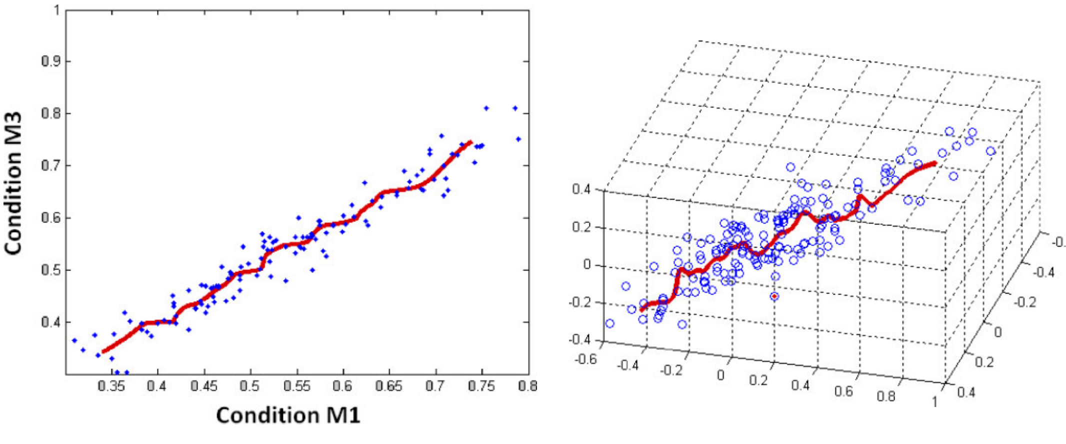
It should be noted that our method can also obtain subspace principal curves which might have some common features as well as data points. As shown in Fig. 5, the two subspace principal curves obtained share the attribute ‘number of games’ and 70% of the data. There was no desired subspace principal curve with all the three attributes. Hence, we can only describe the 2-D ‘*maximally desirable subspace principal curve*’ that does not become desirable when any other feature is added to the resulting feature combinations.

(4) *Mouse Gene Expression Dataset*: The mouse gene expression dataset [35] contains gene expression levels of 147 genes expressed in six different conditions, as shown in Fig. 6. We found a subspace trend with M1 and M2 (removing seven outlier points). Relaxing the threshold, we also obtained subspace principal curves with M1, M2, and M3 and with NC1, NC2, and NC3. In the next section, we show that the gene orderings belonging to the subspace trend reveal biologically more coherent groupings compared to the gene orderings obtained by fitting the principal curve directly.

(5) *Ecoli Gene Expression Dataset*: The Ecoli gene expression dataset [36] contains 102 genes expressed in eight different conditions. The best subspace trend is obtained in 90-1 and 150-1 condition by removing six outlier genes. Fig. 7 shows that the removal of the six data points with the highest projection distances from the principal curve can significantly improve the quality of the subspace trend.

(6) *Drosophila Gene Expression Dataset*: This dataset[4] contains the gene expression of the Drosophila melanogaster during its life cycle with the expression levels of 3944 genes that are evaluated for 57 sequential time periods [37]. Missing values (less than 1%) were replaced by zeros. We were able to extract subspace trends for the three sets of feature combinations $(1, 2, 3, 54, 55, 56, \text{and } 57)$, $(5, 54, 55, 56, 57)$ and $(9, 10, 11)$. It should be noted that the Features 1, 2, and 5 do not form any desired subspace, hence the feature combination with $(1, 2, 3, 5, 54, 55, 56, 57)$ features does not create a desired subspace principal curve. The SuPriC algorithm gave the ‘*maximally desirable subspace principal curves*’.

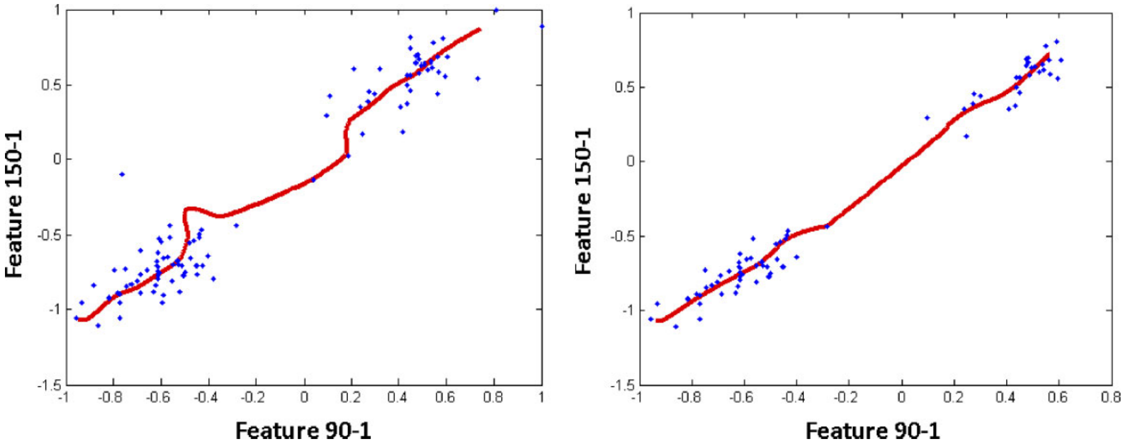
[4] http://genome-www5.stanford.edu.



(a) M1 vs M3

(b)

Fig. 6 Results on Mouse gene expression data. (a) The two highly correlated conditions (M1 and M3) that provide a good gene ordering that yield coherent groupings, (b) 3-D visualization of all the six nonlinearly correlated dimensions after reducing the dimensionality using MDS.



(a)

(b)

Fig. 7 Results on Ecoli gene expression data. (a) The subspace trend obtained before removing the outliers, (b) the subspace trend obtained after removing few outlier data points.

(7) *Gasch Gene Expression Dataset*: The Gasch gene expression dataset with 4532 samples and 250 features corresponding to model organism Yeast [38] was generated by merging the experiments of Spellman (gene expression measures relative to 77 conditions) [39] with the transcriptional responses of yeast to environmental stress (173 conditions).

(8) *SWSequence Dataset*: The SWSequence Data is a pair-wise similarity data with 3527 samples (rows) and 6349 features (columns) from the Smith–Waterman algorithm [40]. This data represents the homological functional relations that exist between genes belonging to the same functional classes. Each data value was computed from the Smith–Waterman $\log E$ values between a pair of yeast sequences that express the pair-wise similarities between the genes.

5.3. Comparison with Other Methods

5.3.1. Feature correlation methods [1,41]

The CARE algorithm proposed in [1] primarily finds the linear subspace correlations in the data and cannot identify the nonlinear correlations in the subspaces. In the presence of nonlinear correlations, it either breaks it into smaller segments or completely misses to identify such relations. In fact, the CARE algorithm found the most prominent (or trivial) correlations but missed some interesting linear correlations that are present in the global space with a relatively high percentage of data. The SuPriC algorithm was able to extract all the subspace correlations presented in [1] as well as others (see Fig. 8) except for the one which says that the total rebound is equal to the sum of the offensive rebound and the defensive rebound. This is because of the absence

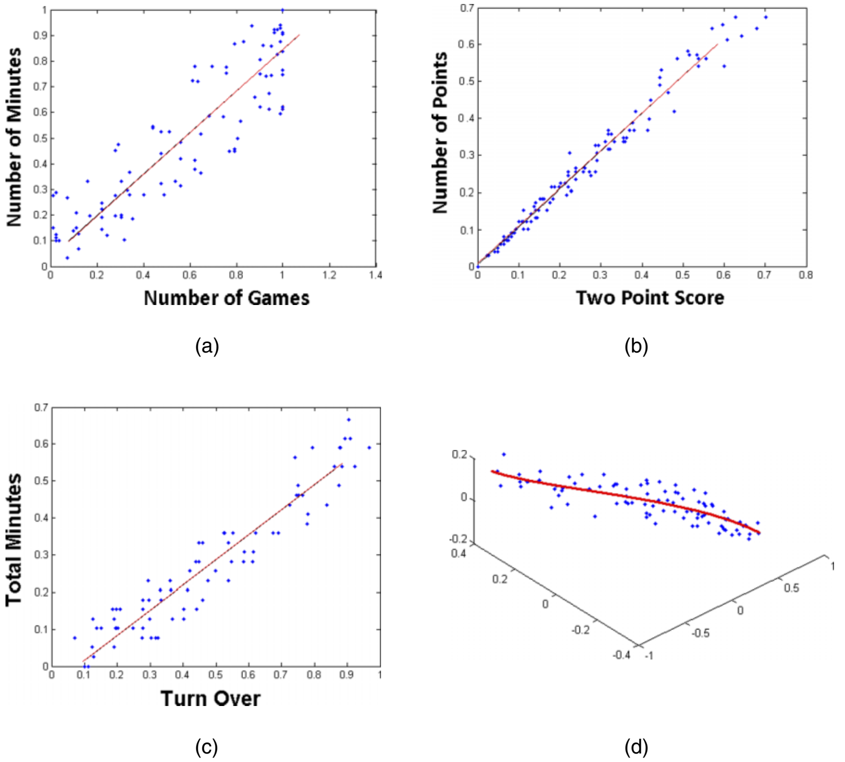


Fig. 8 (a–c) 2-D correlations present in different subspaces of the NBA dataset, (d) the nonlinear correlation between four features is visualized in 3-D after reducing the dimensionality using MDS.

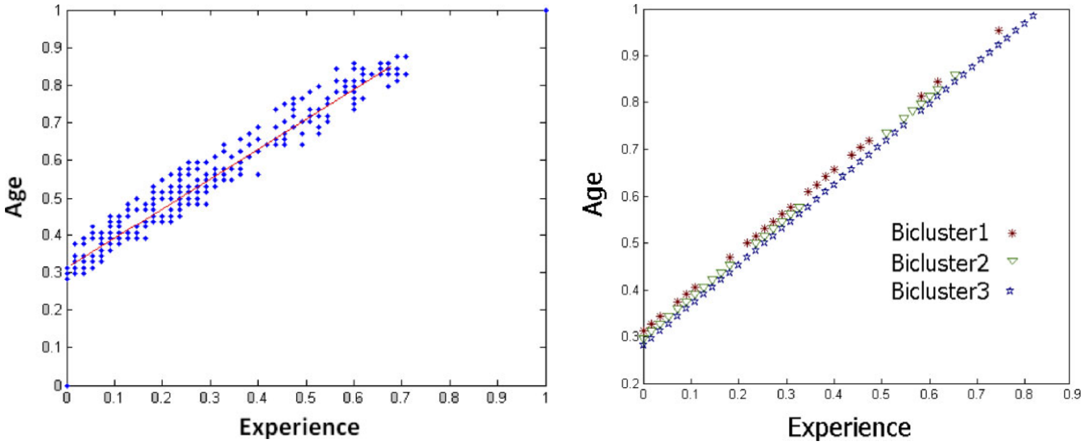
of any correlation between the defensive rebound with the offensive rebound for a large number of data points. However, finding such an obvious relation is not the goal of our algorithm. Rather, we are looking for a correlation that represents a significant amount of data points which is not evident otherwise, that is, the total rebound is not a true feature and can be calculated by summing the other two feature values. The REDUS algorithm [41] was recently proposed to find nonlinear correlations present in the data. In the WDBC dataset, REDUS was able to find only one nonlinear correlation (between mean radius, maximum radius and the mean of texture), whereas our method, was able to find much stronger correlations (between mean radius, mean surface, mean perimeter, maximum radius, maximum surface, maximum perimeter, standard error in radius, standard error in surface and standard error in perimeter) with these nine features in addition to the previously reported result (see Fig. 4). Also, one of the main drawbacks of the REDUS algorithm is its inability to capture overlapping correlations which are readily captured using our approach.

5.3.2. Biclustering algorithms [42]

We performed comparisons with a state-of-the-art biclustering method, namely, the Bivisu algorithm [42]. We used this biclustering algorithm on the wages dataset on the four continuous attributes and obtained three biclusters that are significant. One can clearly see that these biclusters do not convey the essential concept of a subspace correlation since they optimize the criteria of an additive model as shown in Fig. 9. Our algorithm groups the relevant data points in the subspace together indicating a stronger correlation pattern. This is the main advantage of the subspace trend compared to subspace clusters.

5.3.3. Principal curves [33]

For the mouse gene expression data, we also obtained the ordering of the genes by projecting them onto the subspace trend obtained by M1 and M3 and compared the ordering of the genes by projecting them onto the original principal curve fitted on the entire data (see Fig. 6). We can evaluate the biological validity of the gene orderings provided by normal and subspace trends by examining the biological homogeneity of neighboring genes in the trend. A good trend should arrange genes into contiguous clusters of functionally related genes. Therefore, if we divide trends into segments, better trends should exhibit a higher similarity between functions of genes in the segments. We divided the normal and subspace trends on the mouse dataset into the seven most natural segments using the change point analysis technique [43]. We then evaluated the segments using



(a) Age vs Experience

(b) Age vs Experience

Fig. 9 Comparison results on the Wages dataset. (a) Our method was able to capture the correlation between the AGE and EXPERIENCE attributes, (b) Fewer data points (compared to the number obtained using SuPriC) broke into three different biclusters for the AGE and EXPERIENCE attributes.

Table 3. Comparison of NRMSE values for missing value imputation in different real-world datasets.

	KNNImpute [45]	Local Least Squares [46]	Principal Curves [33]	SuPriC
Wages	0.415 (0.0212)	0.35 (0.0213)	0.33 (0.0217)	**0.278 (0.0191)**
WDBC	0.422 (0.0237)	0.367 (0.0215)	0.349 (0.0195)	**0.285 (0.0185)**
Ecoli	0.46 (0.0233)	0.427 (0.0204)	0.415 (0.0202)	**0.35 (0.0178)**
Mouse	0.409 (0.0221)	0.39 (0.0197)	0.349 (0.0188)	**0.332 (0.0182)**
Gasch	0.41 (0.023)	0.38 (0.0221)	0.35 (0.0203)	**0.33 (0.017)**
SWSeq	0.49 (0.0242)	0.48 (0.021)	0.465 (0.0213)	**0.455 (0.019)**

the Biological Homogeneity Index (BHI) proposed by [44], an external validation measure based on functional annotations ranging from 0 to 100 with higher values being better. We obtained functional annotations for the mouse dataset from the original authors containing nine functional groups. The functional groups were not necessarily disjoint, and many genes belonged to more than one functional group. The subspace trends obtained better BHI than the normal trends. The BHI for the subspace trend was 15.034, and for the normal trend was 14.014. Note that a single point increase in BHI is statistically significant since the effective ceiling for the BHI with the functional annotations used was approximately 20.

5.3.4. Missing data imputation [45,46]

Our method was also used to improve the quality of missing data imputation. We tested our algorithm on both synthetic and real-world datasets. The SuPriC algorithm was compared against the principal curve in the full-dimensional space along with two well-studied missing data imputation methods, namely, KNN imputation (KNNImpute) [45] and the Least Squares method [46]. First, some of the values at random locations in the data were intentionally left out and will be considered as the missing data. When the principal curves and SuPriC were used, the data points, containing entry missing, was filled using the average value of the corresponding features to begin with. We estimated the value for the missing feature using the other feature values on the principal curve. While using SuPriC some data points do not fall into any of the subspace principal curves. In such cases, we took the result from the full-dimensional principal curves. In all these cases, the Normalized Root Mean Square Error (NRMSE) between the result obtained, and the actual entry in those missing positions was calculated [45]. We performed the comparison using six real-world datasets with 3% missing values and observed that SuPriC performs better imputation compared to the other methods for missing value imputation (see Table 3). Here it is relevant to note that the missing values were real values; hence, we calculated NRMSE between the actual values in the missing positions and the corresponding imputed values. We performed the same experiments ten times using random missingness and reported the average NRMSE values.

Table 4. Comparison of RMSE values for regression in various real-world datasets.

	Isotonic regression	Lasso regression	SuPriC
Wages	0.059 (0.003)	0.061 (0.006)	**0.053 (0.0013)**
WDBC	0.056 (0.002)	0.058 (0.007)	**0.049 (0.0035)**
Ecoli	0.075 (0.002)	0.073 (0.005)	**0.069 (0.0023)**
Mouse	**0.042 (0.003)**	0.051 (0.008)	**0.042 (0.0031)**
Gasch	0.051 (0.002)	0.053 (0.01)	**0.047 (0.0026)**
SWseq	0.091 (0.006)	**0.079 (0.06)**	0.088 (0.0025)

5.3.5. Regression [47,48]

We also demonstrated the performance of the proposed algorithm in solving regression problems. After normalizing the data, we intentionally left out some of the features, individually estimated them using the SuPriC algorithm and compared the estimation with two standard regression techniques: Isotonic regression [48] and lasso regression [47]. For each of the features, the average of the Root Mean Square Error (RMSE) between the estimated value and the actual value is computed. tenfold cross validation was performed to reduce the bias in the test data selection. When the SuPriC algorithm was used, we examined if the data point belongs to any of the subspace principal curves. If so, the unknown feature value is calculated using other feature values in the corresponding subspace principal curve (if the unknown feature is a part of that subspace). In all the other cases, we took the value using the full-dimensional principal curve and the known feature values. This is especially useful because there is no general nonlinear regression model that one can specify directly. However, in the case of the principal curve and the subspace principal curve, we did not have to specify the model beforehand. We used our method for six real-world biological datasets, and the performance comparison of the SuPriC algorithm along with other methods in terms of the RMSE values is reported in Table 4. Our method, in general, outperformed both the methods since it exploits the correlated features and eliminates the noisy features and outliers where ever applicable.

6. CONCLUSION

This paper extends the notion of subspace clusters to subspace trends that can effectively model both linear and nonlinear local subspace correlations that often occur in complex real-world datasets. Many works proposed in the literature fail to effectively extract nonlinear subspace correlation patterns. In most of the real-world problems, one can rarely interpret the usefulness of principal curves in high-dimensional data. In this paper, we formalized the problem of modeling *subspace principal trends* for high-dimensional datasets and proposed SuPriC algorithm for identifying subspace principal curves that optimally represent these subspace trends in high-dimensional feature spaces. The SuPriC algorithm models the principal curves for subspaces rather than the complete feature space and provides a better exploratory analysis of high-dimensional data. The experimental results demonstrate the superiority of the proposed approach compared to other methods developed in the literature. We also demonstrate the improved performance of the proposed algorithm in problems such as missing data imputation and regression analysis compared to some of the standard approaches.

Acknowledgments

We would like to thank Mochan Shrestha and Tin-Kam Ho for providing comments about our work. We also thank the anonymous reviewers for their valuable feedback which significantly improved the quality of our paper.

APPENDIX

PRINCIPAL CURVES

The function *Prin_Curve* takes the subset of data points and returns the best fit principal curve with respect to a subset of the data. To achieve this, one can potentially use any efficient principal curve generating method proposed in the literature. To make this paper self-contained, we provide a high-level description of the k-segments algorithm (see Algorithm 2) used to generate principal curves [33] in our work. This function generates a principal curve for the given data in an incremental manner. First, it computes the principal component of the data covariance matrix and obtains the principal eigenvector corresponding to the largest eigenvalue. From this eigenvector, a segment that covers the projection of all the data points is taken. Then, it iteratively adds new line segments by taking out some of the data points corresponding to the existing segments. For adding a new segment, it first considers all the data points as potential zero length segment and assigns those data points for which this new segment is the closest. The data points assigned to the optimal new zero length segment are used to compute the principal component and eventually the new segment. In this iterative manner, one can add more and more segments and bring more nonlinearity to the curve. A polygonal line is formed by tailoring these segments using the Hamiltonian path algorithm (function call *Connect_Segments*). Finally, it smoothens the polygonal line and generates the principal curve. After the principal curve is obtained, the projection distances of all the data points onto the curve and the length of the curve are returned by this procedure. It should be noted that the number of segments k is a user-defined parameter and for obtaining linear correlation patterns, k must be set to 1. The higher values of k will obtain higher degree nonlinear correlation. Since a suitable value of k is unknown we modified the algorithm so that k is not a user parameter. In our approach, we start with a number of partitions to be 1 and incrementally improve the objective function value by adding more partitions. In most of the cases we tested on, we found that the number of segments is around 3. For datasets having more degree of subspace nonlinearity, it may go beyond three.

Algorithm 2 Prin_Curve($Data$, F)

```
Input: Data matrix ($Data$)
       Feature set ($F$)
Output: Optimal Principal curve ($S$)
        Projection distances of the data ($sqd$)
        Length of the principal curve ($len$)
Pseudocode:
$c \leftarrow |F|$
$v \leftarrow$ find_principal_component($Data$)
$v' \leftarrow$ define_segment_part($v$)
i←1
$V \leftarrow \{v'\}$
$p \leftarrow \infty$
while True do
  x← find_Optimum_Datapoint(Data,V)
  $PART \leftarrow$ partition_Data($Data$,V,x)
  V← Ø
  for each $part \in PART$ do
    $v \leftarrow$ find_principal_component($p$)
    $v' \leftarrow$ define_segment_part($v$)
    V←  V ∪ $v'$
  end for
  i←i+1
  $S \leftarrow Smooth\_Polygon(PL, Data)$
  $sqd \leftarrow Project(Data, S)$
  $ssd \leftarrow sum(sqd)$
  $len \leftarrow Length(S)$
  $q \leftarrow (w \times ssd + (1 - w) \times len)/\sqrt{c} \times |Data|$
  if $q < p$ then
    $p \leftarrow q$
  else
    exit while
  end if
end while
$PL \leftarrow Connect\_Segments$(V)
$S \leftarrow Smooth\_Polygon(PL, Data)$
$sqd \leftarrow Project(Data, S)$
$len \leftarrow Length(S)$
```

REFERENCES

[1] X. Zhang, F. Pan, and W. Wang, Care: Finding local linear correlations in high dimensional data, In IEEE 24th International Conference on Data Engineering (ICDE'08), 2008, 130–139.

[2] S. Pokharkar, and C. Reddy, Identifying information-rich subspace trends in high-dimensional data, In Proceedings of SIAM International Conference on Data Mining (SDM), 2009, 557–568.

[3] M. Turk, and A. Pentland, Eigenfaces for recognition, J Cogn Neurosci 3(1) (1991), 71–86.

[4] T. Cox, and M. Cox, Multidimensional Scaling, Chapman and Hall, 1994.

[5] A. Bell, and T. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Comput 7(6) (1995), 1129–1159.

[6] S. Roweis, and L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290(5500) (2000), 2323–2326.

[7] M. Belkin, and P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput 15(6) (2003), 1373–1396.

[8] J. Tenenbaum, V. Silva, and J. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290(5500) (2000), 2319–2323.

[9] L. Yu, and H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, In The Twentieth International Conference on Machine Leaning (ICML), 2003, 856–863.

[10] A. Jain, M. Murty, and P. Flynn, Data clustering: a review, ACM Comput Surv 31(3) (1999), 264–323.

[11] L. Parsons, E. Haque, and H. Liu, Subspace clustering for high dimensional data: a review, SIGKDD Explor 6(1) (2004), 90–105.

[12] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, In Proceedings of ACM SIGMOD International Conference on Management of Data, 1998, 94–105.

[13] C. Cheng, A. Fu, and Y. Zhang, ENCLUS: Entropy-based subspace clustering for mining numerical data, In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999, 84–93.

[14] S. Goil, H. Nagesh, and A. Choudhary, MAFIA: Efficient and scalable subspace clustering for very large data sets, In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999, 443–452.

[15] C. Aggarwal, J. Wolf, P. Yu, C. Procopiuc, and J. Park, Fast algorithms for projected clustering, In Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, 1999, 61–72.

[16] R. Agrawal, and R. Srikant, Fast algorithms for mining association rules, In Proceedings of VLDB, 1994, 487–499.

[17] W. Cohen, Fast effective rule induction, In Proceedings of International Conference on Machine Learning, 1995, 115–123.

[18] K. Kailing, H. Kriegel, and P. Kroger, Density-connected subspace clustering for high-dimensional data, In Proceedings of SIAM International Conference on Data Mining (SDM), 2004, 246–257.

[19] C. Aggarwal, and P. Yu, Finding generalized projected clusters in high dimensional spaces, In Proceedings ACM SIGMOD International Conference on Management of Data, 2000, 70–81.

[20] K. Woo, J. Lee, M. Kim, and Y. Lee, FINDIT: a fast and intelligent subspace clustering algorithm using dimension voting, Inform Softw Technol 46(4) (2004), 255–271.

[21] Y. Cheng, and G. Church, Biclustering of expression data, In Proceedings of International Conference on Intelligent Systems for Molecular Biology, 2000, 93–103.

[22] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein, Spectral biclustering of microarray data: coclustering genes and conditions, Genome Res 13(4) (2003), 703–716.

[23] H. Wang, W. Wang, J. Yang, and P. Yu, Clustering by pattern similarity in large data sets, In Proceedings ACM SIGMOD International Conference on Management of Data, 2002, 394–405.

[24] J. Yang, W. Wang, H. Wang, and P. Yu, delta-cluster: capturing subspace correlation in a large data set, In International Conference on Data Engineering (ICDE'02), 2002, 517–528.

[25] C. Bohm, K. Kailing, P. Kröger, and A. Zimek, Computing clusters of correlation connected objects, In Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, 2004, 455–466.

[26] M. S. Aziz, and C. K. Reddy, A robust seedless algorithm for correlation clustering, In Advances in Knowledge Discovery and Data Mining, 14th Pacific-Asia Conference (PAKDD), 2010, 28–37.

[27] T. Hastie, and W. Stuetzle, Principal curves, J Am Stat Assoc 84 (1989), 502–516.

[28] B. Kegl, A. Krzyzak, T. Linder, and K. Zeger, Learning and design of principal curves, IEEE Trans Pattern Anal Mach Intell 22(3) (2000), 281–297.

[29] R. Tibshirani, Principal curves revisited, Stat Comput 2 (1992), 182–190.

[30] J. Einbeck, G. Tutz, and L. Evers, Local principal curves, Stat Comput 15(4) (2005), 301–313.

[31] K. Chang, and J. Ghosh, A unified model for probabilistic principal surfaces, IEEE Trans Pattern Anal Machine Intell 23(1) (2001), 22–41.

[32] K. Chang, and J. Ghosh, Three-dimensional model-based object recognition and pose estimation using probabilistic principal surfaces, SPIE Appl Artif Neural Networks Image Process 3962(1) (2000), 192–203.

[33] J. Verbeek, N. Vlassis, and B. Krose, A *k*-segments algorithm for finding principal curves, Pattern Recognition Lett 23(8) (2002), 1009–1017.

[34] Y. Chu, J. Huang, K. Chuang, and M. Chen, On subspace clustering with density consciousness, In Proceedings of the 15th ACM international conference on Information and knowledge management (CIKM), 2006, 804–805.

[35] V. Bhattacherjee, P. Mukhopadhyay, S. Singh, C. Johnson, J. Philipose, C. Warner, R. Greene, and M. Pisano, Neural crest and mesoderm lineage-dependent gene expression in orofacial development, Differentiation 75(5) (2007), 463–477.

[36] W. Schmidt-Heck, R. Guthke, S. Toepfer, H. Reischer, K. Duerrschmid, and K. Bayer, Reverse engineering of the stress response during expression of a recombinant protein, In Proceedings of the EUNITE symposium, 2004, 407–412.

[37] M. Arbeitman, E. Furlong, F. Imam, E. Johnson, B. Null, B. Baker, M. Krasnow, M. Scott, R. Davis, and K. White, Gene expression during the life cycle of drosophila melanogaster, Science 297 (2002), 2270–2275.

[38] P. Gasch, T. Spellman, M. Kao, O. Carmel-Harel, B. Eisen, G. Storz, D. Botstein, P. Brown, and P. Silver, Genomic expression programs in the response of yeast cells to environmental changes, Mol Biol Cell 11 (2000), 4241–4257.

[39] P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher, Comprehensive identification of cell cycle-regulated genes of the yeast saccharomices cerevisiae by microarray hybridization, 9 (1998), 3273–3297.

[40] T. Smith, and M. Waterman, Identification of common molecular subsequences, J Mol Biol 147 (1981), 195–197.

[41] X. Zhang, F. Pan, and W. Wang, Redus: finding reducible subspaces in high dimensional data, In Proceeding of the 17th ACM Conference on Information and Knowledge Management (CIKM'08), 2008, 961–970.

[42] K. Cheng, N. Law, W. Siu, and A. Liew, Identification of coherent patterns in gene expression data using an efficient biclustering algorithm and parallel coordinate visualization, BMC Bioinform 9(210) (2008), 1–28.

[43] P. Green, Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, Biometrika 82 (1995), 711–732.

[44] S. Datta, and S. Datta, Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes, BMC Bioinform 7 (2006), 397–405.

[45] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. Altman, Missing value estimation for DNA microarrays, Bioinformatics 17(6) (2001), 520–525.

[46] H. Kim, G. Golub, and H. Park, Missing value estimation for DNA microarray expression data: local least squares imputation, Bioinformatics 21(2) (2005), 187–198.

[47] R. Tibshirani, Regression shrinkage and selection via the LASSO, J R Stat Soc 58(1) (1996), 267–288.

[48] W. Wu, M. Woodroofe, and G. Mentz, Isotonic regression: another look at the changepoint problem, 88(3) (2001), 793–804.