Finding High-Order Correlations in High-Dimensional Biological Data

Xiang Zhang, Feng Pan, and Wei Wang

Department of Computer Science
University of North Carolina at Chapel Hill

1 Introduction

Many real life applications involve the analysis of high dimensional data. For example, in bio-medical domains, advanced microarray techniques [1, 2] enables monitoring the expression levels of hundreds to thousands of genes simultaneously. By mapping each gene to a feature, gene expression data can be represented by points in a high dimensional feature space. To make sense of such high dimensional data, extensive research has been done in finding the latent structure among the large number of features. In general, existing approaches in analyzing high dimensional data can be summarized into 3 categories [3]: feature selection, feature transformation (or dimension reduction) and projected clustering.

The goal of feature selection methods [4–7] is to find a single representative subset of features that are most relevant for the task at hand, such as classification. The selected features generally have low correlation with each other but have strong correlation with the target feature.

Feature transformation methods [24, 9, 26, 8, 41, 43]summarize the dataset by creating linear/non-linear combinations of features in order to uncover the latent structure. The insight behind feature transformation methods is that a high dimensional dataset may exhibit interesting patterns on a lower dimensional subspace due to correlations among the features. The commonly used linear feature transformation methods include principal component analysis (PCA) [8], linear discriminant analysis (LDA), and their variants (see [9] for an overview). PCA is one of the most widely used feature transformation methods. It seeks an optimal linear transformation of the original feature space such that most variance in the data is represented by a small number of orthogonal derived features in the transformed space. PCA performs one and the same feature transformation on the entire dataset. It aims to model the global latent structure of the data and hence does not separate the impact of any original features nor identify local latent patterns in some feature subspaces.

Recently proposed projected clustering methods, such as [10, 11], can be viewed as combinations of clustering algorithms and PCA. These methods can be applied to find clusters of data points that may not exist in the axis parallel subspaces but only exist in the projected subspaces. The projected subspaces are usually found by applying the standard PCA in the full dimensional space.

Like other clustering methods, projected clustering algorithms find the clusters of points that are spatially close to each other in the projected space. However, a subset of features can be strongly correlated even though the data points do not form any clustering structure.

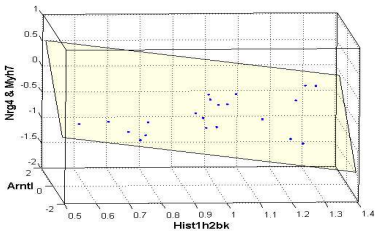
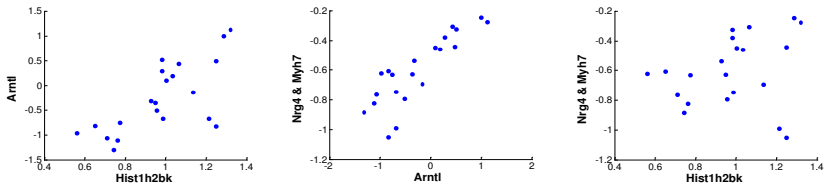


Fig. 1. A strongly correlated gene subset



(a) (Hist1h2bk, Arntl) (b) (Arntl, Nrg4&Myh7) (c) (Hist1h2bk,Nrg4&Myh7)

Fig. 2. Pair-wise correlations of a strongly correlated gene subset

1.1 Motivation

In many emerging applications, the datasets usually consist of thousands to hundreds of thousands of features. In such high dimensional dataset, some feature subsets may be strongly correlated, while others may not have any correlation at all. In these applications, it is more desirable to find the correlations that are hidden in feature subspaces. For example, in gene expression data analysis, a group of genes having strong correlation is of high interests to biologists since it helps to infer unknown functions of genes [1] and gives rise to hypotheses regarding the mechanism of the transcriptional regulatory network [2]. We refer to such correlation among a subset of features as a *local correlation* in comparison with the global correlation found by the full dimensional feature reduction methods. Since such local correlations only exist in some subspaces of the full dimensional space, they are invisible to the full feature transformation methods.

Recently, many methods [1, 12] have been proposed for finding clusters of features that are pair-wisely correlated. However, a set of features may have strong correlation but each pair of features only weakly correlated.

For example, Figure 1 shows 4 genes that are strongly correlated in the mouse gene expression data collected by the biologists in the School of Public Health at UNC. All of these 4 genes have same Gene Ontology (GO) [13] annotation *cell part*, and three of which, Myh7, Hist1h2bk, and Arntl, share the same GO annotation *intracelluar part*. The linear relationship identified by our algorithm is $-0.4(Nrg4) + 0.1(Myh7) + 0.7(Hist1h2bk) - 0.5(Arntl) = 0$. As we can see from the figure, all data points almost perfectly lay on the same hyperplane, which shows that the 4 genes are strong correlated. (In order to visualize this 3-dimensional hyperplane, we combine two features, Nrg4 and Myh7, into a single axis as $-0.4(Nrg4) + 0.1(Myh7)$ to reduce it to a 2-dimensional hyperplane.) If we project the hyperplane onto 2 dimensional spaces formed by each pair of genes, we find none of them show strong correlation, as depicted in Figures 2(a) to 2(c).

Projected clustering algorithms [10] have been proposed to find the clusters of data points in projected feature spaces. This is driven by the observation that clusters may exist in arbitrarily oriented subspaces. Like other clustering methods, these methods tend to find the clusters of points that are spatially close to each other in the feature space. However, as shown in Figure 1, a subset of features (genes in this example) can still be strongly correlated even if the data points are far away from each other. This property makes such strong correlations invisible to the projected clustering methods. Moreover, to find the projections of original features, projected clustering methods apply PCA in the full dimensional space. Therefore they cannot decouple the local correlations hidden in the high dimensional data.

In [45], an algorithm is proposed to find local linear correlations in high dimensional data. However, in real applications, the feature subspace can be either linearly or nonlinearly correlated. The problem of finding linear and nonlinear correlations in feature subspaces remains open.

For example, Figure 3 shows a data sets consisting of 12 features, $\{f_1, f_2, \cdots, f_{12}\}$, and 1000 data points. Embedded in the full dimensional space, features subspaces $\{f_1, f_2, f_3\}$ and $\{f_4, f_5, f_6\}$ are nonlinearly correlated, $\{f_7, f_8, f_9\}$ are linearly correlated. Features $\{f_{10}, f_{11}, f_{12}\}$ contain random noises.

Performing feature transformation methods to the full dimensional space cannot uncover these local correlations hidden in the full feature spaces. For example, Figure 4(a) shows the result of applying Principal Component Analysis (PCA)[8] to the full dimensional space of the example dataset shown in Figure 3. In this figure, we plot the point distribution on the first 3 principal components found by PCA. Clearly, we cannot find any pattern that is similar to the patterns embedded in the dataset. Similarly, Figure 4(b) shows the results of applying ISOMAP [43] to reduce the dimensionality of the dataset down to 3. There is also no desired pattern found in this low dimensional structure.

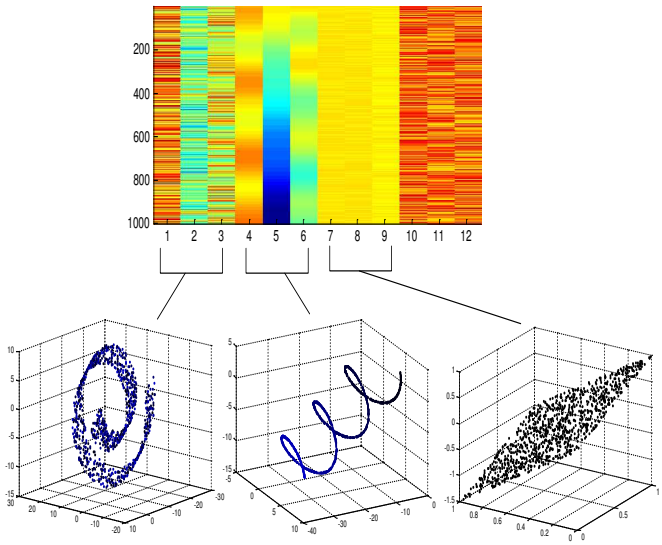
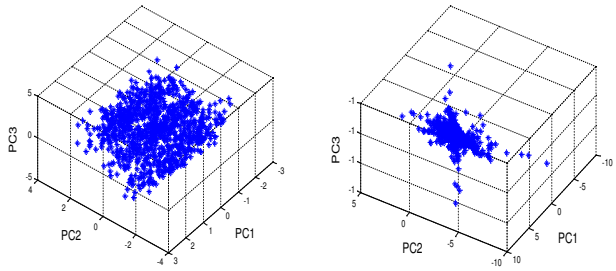


Fig. 3. An example dataset



(a) Result of PCA (b) Result of ISOMAP

Fig. 4. Applying dimensionality reduction methods to the full dimensional space of the example dataset

How can we identify these local correlations hidden in the full dimensional space?

This question is two-fold. First, we need to identify the strongly correlated feature subspaces, i.e., a subset of features that are strongly correlated and actually have low dimensional structures. Then, after these locally correlated feature subsets are found, we can apply the existing dimensionality reduction methods to identify the low dimensional structures embedded in them.

Many methods have been proposed to address the second aspect of the question, i.e., given a correlated feature space, finding the low dimensional embedding

in it. The first aspect of the question, however, is largely untouched. In this chapter, we investigate the first aspect of the question, i.e., identifying the strongly correlated feature subspaces.

1.2 Challenges and Contributions

(1) In this paper, we investigate the problem of finding correlations hidden in the feature subspaces of high dimensional data. The correlations can be either linear or nonlinear. To our best knowledge, our work is the first attempt to find local linear and nonlinear correlations hidden in feature subspaces.

For both linear and non-linea cases, we formalize the problem as finding *reducible subspaces* in the full dimensional space. We adopt the concept of PCA [8] to model the linear correlations. The PCA analysis is repeated applied on subsets of features. In the non-linear cases, we utilize intrinsic dimensionality [30] to detect reducible subspaces. Informally, a set of features are correlated if the intrinsic dimensionality of the set is smaller than the number of features. Various intrinsic dimensionality estimators have been developed [27, 31, 36]. Our problem formalization does not depend on any particular method for estimating the intrinsic dimensionality.

(2) We develop an efficient algorithm, CARE[1], for finding local linear correlations. CARE utilizes spectrum properties about the eigenvalues of the covariance matrix, and incorporates effective heuristic to improve the efficiency.

(3) We develop an effective algorithm REDUS[2] to detect non-linearly correlated feature subsets. REDUS consists of the following two steps.

It first finds the union of all reducible subspaces, i.e., the *overall reducible subspace*. The second step is to uncover the individual reducible subspaces in the overall reducible subspace. The key component of this step is to examine if a feature is strongly correlated with a feature subspace. We develop a method utilizing point distributions to distinguish the features that are strongly correlated with a feature subspace and those that are not. Our method achieves similar accuracy to that of directly using intrinsic dimensionality estimators, but with much less computational cost.

Extensive experiments on synthetic and real life datasets demonstrate the effectiveness of CARE and REDUS.

2 Related Work

Feature Transformation Feature transformation methods can be categorized into linear methods, such as Multi-Dimensional Scaling (MDS) [26] and Principal Component Analysis (PCA)[8], and non-linear methods, such as Local Linear Embedding (LLE) [41], ISOMAP [43], and Laplacian eigenmaps [24]. For high

[1] CARE stands for finding loCAl lineaR corrElations.
[2] REDUS stands for REDUcible Subspaces.

dimensional datasets, if there exist low dimensional subspaces or manifolds embedded in the full dimensional spaces, these methods are successful in identifying these low dimensional embeddings.

Feature transformation methods are usually applied on the full dimensional space to capture the independent components among all the features. They are not designed to address the problem of identifying correlation in feature subspaces. It is reasonable to apply them to the feature spaces that are indeed correlated. However, in very high dimensional datasets, different feature subspaces may have different correlations, and some feature subspace may not have any correlation at all. In this case, dimensionality reduction methods should be applied after such strongly correlated feature subspaces have been identified.

Feature selection Feature selection methods [4–7] try to find a subset of features that are most relevant for certain data mining task, such as classification. The selected feature subset usually contains the features that have low correlation with each other but have strong correlation with the target feature. In order to find the relevant feature subset, these methods search through various subsets of features and evaluate these subsets according to certain criteria. Feature selection methods can be further divided into two groups based on their evaluation criteria: wrapper and filter. Wrapper models evaluate feature subsets by their predictive accuracy using statistical re-sampling or cross-validation. In filter techniques, the feature subsets are evaluated by their information content, typically statistical dependence or information-theoretic measures. Similar to feature transformation, feature selection finds one feature subset for the entire dataset.

Subspace clustering Subspace clustering is based on the observation that clusters of points may exist in different subspaces. Many methods [18–20] have been developed to find clusters in axes paralleling subspaces. Recently, the projected clustering was studied in [10], inspired by the observation that clusters may exist in arbitrarily oriented subspaces. These methods can be treated as combinations of clustering algorithms and PCA. Similar to other clustering methods, these methods tend to find the clusters of points that are close to each other in the projected space. However, as shown in Figure 1, a subset of features still can be strongly correlated even if the data points are far away from each other. Pattern based bi-clustering algorithms have been studied in [1, 12]. These algorithms find the clusters in which the data points share pair-wise linear correlations, which is only a special case of linear correlation.

Intrinsic Dimensionality Due to correlations among features, a high dimensional dataset may lie in a subspace with dimensionality smaller than the number of features [27, 31, 36]. The intrinsic dimensionality can be treated as the minimum number of free variables required to define the data without any significant information loss [30]. For example, as shown in Figure 3, in the 3-dimensional space of $\{f_1, f_2, f_3\}$, the data points lie on a Swiss roll, which is actually a 2-dimensional manifold. Therefore, its intrinsic dimensionality is 2.

The concept of intrinsic dimensionality has many applications in the database and data mining communities, such as clustering [23, 32], outlier detection [38],

nearest neighbor queries [37], and spatial query selectivity estimation [25, 29]. Different definitions of intrinsic dimensionality can be found in the literature. For example, in linear cases, matrix rank [33] and PCA [8] can be used to estimate intrinsic dimensionality. For nonlinear cases, estimators such as *box counting dimension*, *information dimension*, and *correlation dimension* have been developed. These intrinsic dimensionality estimators are sometimes collectively referred to as *fractal dimension*. Please see [39, 42] for good coverage of the topics of intrinsic dimensionality estimation and its applications.

3 Problem Formalization

In this section, we utilize PCA and intrinsic dimensionality to formalize the problem of finding strongly correlated feature subspaces in linear and non-linear cases respectively .

Suppose that the dataset Ω consists of N data points and M features. Let $\Omega_P = \{p_1, p_2, \cdots, p_N\}$ denote the point set, and $\Omega_F = \{f_1, f_2, \cdots, f_M\}$ denote the feature set in Ω respectively. In the following sections, we define the linear and non-linear reducible subspaces.

3.1 Linear Reducible Subspace

A strongly linear-correlated feature subset is a subset of features that show strong linear correlation in a large portion of data points.

Definition 1. Strongly Linear-correlated Feature Subset
Let $\Omega' = \{\boldsymbol{f}_{i_1}, \cdots, \boldsymbol{f}_{i_m}\} \times \{\boldsymbol{p}_{j_1}, \cdots, \boldsymbol{p}_{j_n}\}$ be a submatrix of Ω, where $1 \leq i_1 < i_2 < \cdots < i_m \leq M$ and $1 \leq j_1 < j_2 < \cdots < j_n \leq N$. C_F is the covariance matrix of Ω'. Let $\{\lambda_l\}$ $(1 \leq l \leq n)$ be the eigenvalues of C_F and arranged in increasing order[3]*, i.e., $\lambda_1 \leq \lambda_2, \cdots, \leq \lambda_n$. The features $\{\boldsymbol{f}_{i_1}, \cdots, \boldsymbol{f}_{i_m}\}$ is a* ***strongly linear-correlated feature subset*** *if the value of the objective function $f(\Omega', k) = \frac{\Sigma_{t=1}^{k}\lambda_t}{\Sigma_{t=1}^{m}\lambda_t} \leq \eta$ and $n/N \geq \delta$, where k, η and δ are user specified parameters.*

Eigenvalue λ_l is the variance on eigenvector $\mathbf{v}_l$ [8]. The set of eigenvalues $\{\lambda_l\}$ of matrix $C_{\Omega'}$ is also called the *spectrum* of $C_{\Omega'}$ [21].

Geometrically, each $n \times m$ submatrix of Ω represents an m-dimensional space with n points in it. This m-dimensional space can be partitioned into two subspaces, S_1 and S_2, which are orthogonal to each other. S_1 is spanned by the k eigenvectors with smallest eigenvalues and S_2 is spanned by the remaining $m-k$ eigenvectors. Intuitively, if the variance in subspace S_1 is small (equivalently the variance in S_2 is large), then the feature subset is strongly linear-correlated. The input parameters k and threshold η for the objective function $f(\Omega', k) = \frac{\Sigma_{t=1}^{k}\lambda_t}{\Sigma_{t=1}^{m}\lambda_t}$

[3] In this chapter, we assume that the eigenvalues are always arranged in increasing order. Their corresponding eigenvectors are $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n\}$.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
p_1	0.7988	**3.8905**	0.6548	−0.6646	0.5536	1.3242	**−0.5532**	0.3158	**−1.1613**
p_2	0.8968	**1.3365**	−1.2484	0.5582	−1.5564	−0.1265	**0.2983**	1.3437	**−1.1098**
p_3	0.1379	**3.1503**	−0.5975	−1.1885	−0.2067	−0.7372	**−1.2266**	−2.2378	**0.2907**
p_4	−1.6191	**3.9939**	−0.4818	−0.7755	−0.4256	0.2137	**−0.1897**	1.2929	**−1.9102**
p_5	−1.6466	**−1.1069**	0.9834	0.271	0.4938	−0.4005	**−0.3017**	−0.3785	**1.3148**
p_6	0.4287	**−4.0447**	1.7621	1.535	−0.8709	0.0649	**0.957**	0.0025	**0.6653**
p_7	**−0.982**	**2.1536**	1.4274	−1.0523	**0.0798**	**−1.758**	**−0.5334**	**0.8846**	**−0.2751**
p_8	**−5.1084**	**2.7252**	0.9118	0.6256	**−0.5216**	**1.6867**	**−0.9011**	**0.5825**	**−0.023**
p_9	**10.9007**	**4.3305**	0.3268	−0.7976	**−1.4139**	**0.3274**	**−0.8926**	**−1.6142**	**−0.908**
p_{10}	**7.3744**	−0.8533	0.0696	−0.3135	**−0.3843**	**0.716**	0.2787	**−1.5037**	−1.0437
p_{11}	**−4.9437**	0.3456	−1.4998	−0.6022	**−0.4579**	**1.5986**	−0.7458	**0.5736**	0.3735
p_{12}	**9.7836**	0.1098	−0.4182	1.2591	**−0.2915**	**−2.0647**	1.6035	**−0.9105**	0.9015
p_{13}	**10.9429**	−1.133	−0.021	0.8585	**−0.3012**	**−0.7436**	0.5743	**−1.6313**	1.2785
p_{14}	**5.9643**	−0.6831	0.2284	−2.1053	**−1.5886**	**0.1762**	0.3207	**−0.3591**	−0.1285
p_{15}	**−2.0985**	−0.2779	−1.0082	−0.3609	**1.0943**	**0.5278**	−0.1514	**−0.3976**	0.6128

Fig. 5. An example dataset containing linear-correlated feature subsets

Feature subset	$\{\mathbf{f}_2, \mathbf{f}_7, \mathbf{f}_9\}$
Eigenvalues of $C_{\Omega'}$	$\lambda_1 = 0.001, \lambda_2 = 0.931, \lambda_3 = 2.067$
Input parameters	$k = 1$, $\eta = 0.004$ and $\delta = 60\%$
Objective function value	$f(\Omega', k) = 0.0003$

Table 1. An example of strongly linear-correlated feature subset

are used to control the strength of the correlation among the feature subset. The default value of k is 1. The larger the value of k, the stronger the linear correlation.

The reason for requiring $n/N \geq \delta$ is because a feature subset can be strongly linear-correlated only in a subset of data points. In our definition, we allow the strongly linear-correlated feature subsets to exist in a large portion of the data points in order to handle this situation. Note that it is possible that a data point may participate in multiple local correlations held by different feature subsets. This makes the local correlations more difficult to detect. Please also note that for a given strongly linear-correlated feature subset, it is possible that there exist multiple linear correlations on different subsets of points. In this chapter, we focus on the scenario where there exists only one linear correlation for a strongly linear-correlated feature subset.

For example, in the dataset shown in Figure 5, the features in submatrix $\Omega' = \{\mathbf{f}_2, \mathbf{f}_7, \mathbf{f}_9\} \times \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_9\}$ is a strongly linear-correlated feature subset when $k = 1$, $\eta = 0.004$ and $\delta = 60\%$. The eigenvalues of the covariance matrix,

Point subset $P_1 = \{\mathbf{p}_1, \cdots, \mathbf{p}_{15}\}$	
Feature subset V_1	$f(\Omega_1, k) = 0.1698$
Feature subset $V_1 \cup \{\mathbf{f}_9\}$	$f(\Omega_1', k) = 0.0707$
Feature subset $V_1 \cup \{\mathbf{f}_4, \mathbf{f}_9\}$	$f(\Omega_1^{''}, k) = 0.0463$

Table 2. Monotonicity with respect to feature subsets

$C_{\Omega'}$, the input parameters and the value of the objective function are shown in Table 1.

The spectrum of covariance matrix has a well-known theorem which is often called the *interlacing eigenvalues theorem*[4] [21].

Theorem 1. *Let* $\Omega' = \{\boldsymbol{f}_{i_1}, \cdots, \boldsymbol{f}_{i_m}\} \times \{\boldsymbol{p}_{j_1}, \cdots, \boldsymbol{p}_{j_n}\}$ *and* $\Omega^{''} = \{\boldsymbol{f}_{i_1}, \cdots, \boldsymbol{f}_{i_m}, \boldsymbol{f}_{i_{(m+1)}}\} \times \{\boldsymbol{p}_{j_1}, \cdots, \boldsymbol{p}_{j_n}\}$ *be two submatrices of* Ω. $C_{Omega'}$ *and* $C_{\Omega''}$ *are their covariance matrices with eigenvalues* $\{\lambda_l\}$ *and* $\{\lambda_l^{'}\}$. *We have*

$$\lambda_1^{'} \leq \lambda_1 \leq \lambda_2^{'} \leq \lambda_2 \leq \cdots \leq \lambda_{m-1} \leq \lambda_m^{'} \leq \lambda_m \leq \lambda_{m+1}^{'}.$$

Theorem 1 tells us that the spectra of $C_{\Omega'}$ and $C_{\Omega''}$ interleave each other, with the eigenvalues of the larger matrix bracketing those of the smaller one.

By applying the interlacing eigenvalues theorem, we have the following property for the strongly linear-correlated feature subsets.

Property 1. (Upward closure property of strongly linear-correlated feature subsets) Let $\Omega' = V' \times P$ and $\Omega^{''} = V^{''} \times P$ be two submatrices of Ω with $V' \subseteq V^{''}$. If V' is a strongly linear-correlated feature subset, then $V^{''}$ is also a strongly linear-correlated feature subset.

Proof. We show the proof for the case where $|V^{''}| = |V'| + 1$, i.e., V' is a subset of $V^{''}$ by deleting one feature from V'. Let $C_{\Omega'}$ and $C_{\Omega''}$ be the covariance matrices of Ω' and $\Omega^{''}$ with eigenvalues $\{\lambda_l\}$ and $\{\lambda_l^{'}\}$. Since V' is a strongly linear-correlated feature subset, we have $f(\Omega', k) = \dfrac{\Sigma_{t=1}^{k}\lambda_t}{\Sigma_{t=1}^{m}\lambda_t} \leq \eta$. By applying the interlacing eigenvalues theorem, we have $\Sigma_{t=1}^{k}\lambda_t \geq \Sigma_{t=1}^{k}\lambda_t^{'}$ and $\Sigma_{t=1}^{m}\lambda_t \leq \Sigma_{t=1}^{m+1}\lambda_t^{'}$. Thus $f(\Omega^{''}, k) = \dfrac{\Sigma_{t=1}^{k}\lambda_t^{'}}{\Sigma_{t=1}^{m+1}\lambda_t^{'}} \leq \eta$. Therefore, $V^{''}$ is also a strongly linear-correlated feature subset. By induction we can prove for the cases where V' is a subset of $V^{''}$ by deleting more than one feature.

The following example shows the monotonicity of the objective function with respect to the feature subsets. Using the dataset shown in Figure 5, let $\Omega_1 = V_1 \times P_1 = \{\mathbf{f}_2, \mathbf{f}_7\} \times \{\mathbf{p}_1, \cdots, \mathbf{p}_{15}\}$, $\Omega_1' = (V_1 \cup \{\mathbf{f}_9\}) \times P_1$, and $\Omega_1^{''} = (V_1 \cup \{\mathbf{f}_4, \mathbf{f}_9\}) \times P_1$. The values of the objective function, when $k = 1$, are shown in Table 2. It can

[4] This theorem also applies to Hermitian matrix [21]. Here we focus on the covariance matrix, which is semi-positive definite and symmetric.

Feature subset $V_2 = \{\mathbf{f}_2, \mathbf{f}_7, \mathbf{f}_9\}$	
Point subset P_2	$f(\Omega_2, k) = 0.0041$
Point subset $P_2 \cup \{\mathbf{p}_{10}\}$	$f(\Omega_2', k) = 0.0111$
Point subset $P_2 \cup \{\mathbf{p}_{14}\}$	$f(\Omega_2'', k) = 0.0038$

Table 3. No monotonicity with respect to point subsets

be seen from the table that the value of the objective function monotonically decreases when adding new features.

On the other hand, adding (or deleting) data points to a fixed feature subset may cause the correlation of the features to either increase or decrease. That is, the objective function is non-monotonic with respect to the point subsets. We use the following example to show the non-monotonicity of the objective function with respect to the point subsets. Using the dataset shown in Figure 5, let $\Omega_2 = V_2 \times P_2 = \{\mathbf{f}_2, \mathbf{f}_7, \mathbf{f}_9\} \times \{\mathbf{p}_1, \cdots, \mathbf{p}_9, \mathbf{p}_{11}\}$, $\Omega_2' = V_2 \times (P_2 \cup \{\mathbf{p}_{10}\})$, and $\Omega_2'' = V_2 \times (P_2 \cup \{\mathbf{p}_{14}\})$. The values of their objective functions, when $k = 1$, are shown in Table 3. It can be seen from the table that the value of the objective function f can either increase or decrease when adding more points.

We define the linear reducible subspace.

Definition 2. Linear Reducible Subspace
A submatrix $\Omega' = V \times P$ is a linear reducible subspace iff*: 1) Feature set V is strongly linear-correlated; 2) None of the feature subsets of V is strongly linear-correlated.*

3.2 Non-linear Reducible Subspace

PCA can only measure linear correlations. In this section, we extend the problem to non-linear correlations and non-linear reducible subspaces. In stead of specifically using "non-linear", we use the general terms, "correlation" and "reducible subspace", for both linear and non-linear cases.

We use intrinsic dimensionality to define correlated features (linear and non-linear). Given a submatrix $\Omega' = V \times P$, we use $ID(V)$ to represent the intrinsic dimensionality of the feature subspace $V \in \Omega_F$. Intrinsic dimensionality provides a natural way to examine whether a feature is correlated with some feature subspace: if a feature $f_a \in \Omega_F$ is strongly correlated with a feature subspace $V \subseteq \Omega_F$, then adding f_a to V should not cause much change of the intrinsic dimensionality of V. The following definition formalizes this intuition.

Definition 3. (Strong Correlation)
A feature subspace $V \subseteq \Omega_F$ and a feature $f_a \in \Omega_F$ have strong correlation, if

$$\Delta ID(V, f_a) = ID(V \cup \{f_a\}) - ID(V) \leq \epsilon.$$

In this definition, ϵ is a user specified threshold. Smaller ϵ value implies stronger correlation, and larger ϵ value implies weaker correlation. If V and f_a have strong correlation, we also say that they are strongly correlated.

Definition 4. (REDUNDANCY)
Let $V = \{f_{v_1}, f_{v_2}, \cdots, f_{v_m}\} \subseteq \Omega_F$. $f_{v_i} \in V$ is a redundant feature of V, if f_{v_i} has strong correlation with the feature subspace consisting of the remaining features of V, i.e.,

$$\Delta ID(\{f_{v_1}, \cdots, f_{v_{i-1}}, f_{v_{i+1}}, \cdots, f_{v_m}\}, f_{v_i}) \leq \epsilon.$$

We say V is a *redundant* feature subspace if it has at least one redundant feature. Otherwise, V is a *non-redundant* feature subspace.

Note that in Definitions 3 and 4, $ID(V)$ does not depend on a particular intrinsic dimensionality estimator. Any existing estimator can be applied when calculating $ID(V)$. Moreover, we do not require that the intrinsic dimensionality estimator reflects the exact dimensionality of the dataset. However, in general, a good intrinsic dimensionality estimator should satisfy two basic properties.

First, if a feature is redundant in some feature subspace, then it is also redundant in the supersets of the feature subspace. We formalize this intuition as the following property.

Property 2. For $V \in \Omega_F$, if $\Delta ID(V, f_a) \leq \epsilon$, then $\forall U\ (V \subseteq U \subseteq \Omega_F)$, $\Delta ID(U, f_a) \leq \epsilon$.

This is a reasonable requirement, since if f_a is strongly correlated with $V \subseteq U$, then adding f_a to U will not greatly alter its intrinsic dimensionality.

From this property, it is easy to see that, if feature subspace U is non-redundant, then all of its subsets are non-redundant, which is clearly a desirable property for the feature subspaces.

Corollary 1. *If $U \subseteq \Omega_F$ is non-redundant, then for $\forall V \subseteq U$, V is also non-redundant.*

The following property extend the concept of basis [35] in a linear space to nonlinear space using intrinsic dimensionality. In linear space, suppose that V and U contain the same number of vectors, and the vectors in V and U are all linearly independent. If the vectors of U are in the subspace spanned by the vectors of V, then the vectors in V and the vectors in U span the same subspace. (A span of a set of vectors consists of all linear combinations of the vectors.) Similarly, in Property 3, for two non-redundant feature subspaces, V and U, we require that if the features in U are strongly correlated with V, then U and V are strongly correlated with the same subset of features.

Property 3. Let $V = \{f_{v_1}, f_{v_2}, \cdots, f_{v_m}\} \subseteq \Omega_F$ and $U = \{f_{u_1}, f_{u_2}, \cdots, f_{u_m}\} \subseteq \Omega_F$ be two non-redundant feature subspaces. If $\forall f_{u_i} \in U$, $\Delta ID(V, f_{u_i}) \leq \epsilon$, then for $\forall f_a \in \Omega_F$, $\Delta ID(U, f_a) \leq \epsilon$ iff $\Delta ID(V, f_a) \leq \epsilon$.

Intuitively, if a feature subspace Y ($Y \subseteq \Omega_F$) is redundant, then Y should be reducible to some subspace, say V ($V \subset Y$). Concerning the possible choices of V, we are most interested in the smallest one that Y can be reduced to, since it represents the intrinsic dimensionality of Y. We now give the formal definitions of reducible subspace and its core space.

Definition 5. (Reducible Subspace and Core Space)
$Y \subseteq \Omega_F$ is a reducible subspace (linear or non-linear) if there exists a non-redundant subspace V ($V \subset Y$), such that
(1) $\forall f_a \in Y$, $\Delta ID(V, f_a) \leq \epsilon$, and
(2) $\forall U \subset Y$ ($|U| \leq |V|$), U is non-redundant.
We say V is the core space of Y, and Y is reducible to V.

Criterion (1) in Definition 5 says that all features in Y are strongly correlated with the core space V. The meaning of criterion (2) is that the core space is the smallest non-redundant subspace of Y with which all other features of Y are strongly correlated.

Among all reducible subspaces, we are most interested in the maximum ones. A maximum reducible subspace is a reducible subspace that includes all features that are strongly correlated with its core space.

Definition 6. (Maximum Reducible Subspace)
$Y \subseteq \Omega_F$ is a maximum reducible subspace if
(1) Y is a reducible subspace, and
(2)$\forall f_b \in \Omega_F$, if $f_b \notin Y$, then $\Delta ID(V, f_b) > \epsilon$, where V is the core space of Y.

Let $\{Y_1, Y_2, \cdots, Y_S\}$ be the set of all maximum reducible subspaces in the dataset. The union of the maximum reducible subspaces $OR = \bigcup_{i=1}^{S} Y_i$ is referred to as the *overall reducible subspace*.

Note that Definition 6 works for the general case where a feature can be in different maximum reducible subspaces. In this chapter, we focus on the special case where maximum reducible subspaces are non-overlapping, i.e., each feature can be in *at most one* maximum reducible feature subspace.

In the following sections, we present the CARE and REDUS algorithms which efficiently detect linear reducible subspaces and maximum (non-linear) reducible subspaces in high dimensional data.

4 The CARE Algorithm

In this section, we present the algorithm CARE for finding the linear reducible subspace (Definition 2). CARE enumerates the combinations of features to generate candidate feature subsets. To examine if a candidate is a linear reducible subspace, CARE adopts a 2-step approach. It first checks if the feature subset is strongly correlated on all data points. If not, CARE then apply point deletion heuristic to find the appropriate subset of points on which the current feature subset may become strongly correlated. In Section 4.1, we first discuss the overall procedure of enumerating candidate feature subsets. In Section 4.2, we present the heuristics for choosing the point subsets for the candidates that are not strongly correlated on all data points.

4.1 Feature Subsets Selection

For any submatrix $\Omega' = V \times \{\mathbf{p}_1, \cdots, \mathbf{p}_M\}$ of Ω, in order to check whether feature subset V' is strongly correlated, we can perform PCA on Ω' to see if its objective function value is lower than the threshold, i.e., if $f(\Omega', k) = \frac{\Sigma_{t=1}^{k} \lambda_t}{\Sigma_{t=1}^{m} \lambda_t} \leq \eta$.

Starting from feature subsets containing a single feature, CARE adopts depth first search to enumerate combinations of features to generate candidate feature subsets. In the enumeration process, if we find that a candidate feature subset is strongly correlated by evaluating its objective function value, then all its supersets can be pruned according to Property 1.

Next, we present an upper bound on the value of the objective function, which can help to speed up the evaluation process. The following theorem shows the relationship between the diagonal entries of a covariance matrix and its eigenvalues [21].

Property 4. Let Ω' be a submatrix of Ω and $C_{\Omega'}$ be the $m \times m$ covariance matrix of Ω'. Let $\{a_i\}$ be the diagonal entries of $C_{\Omega'}$ arranged in increasing order, and $\{\lambda_i\}$ be the eigenvalues of $C_{\Omega'}$ arranged in increasing order. Then $\Sigma_{t=1}^{s} a_t \geq \Sigma_{t=1}^{s} \lambda_t$ for all $s = 1, 2, \cdots, n$, with equality held for $s = m$.

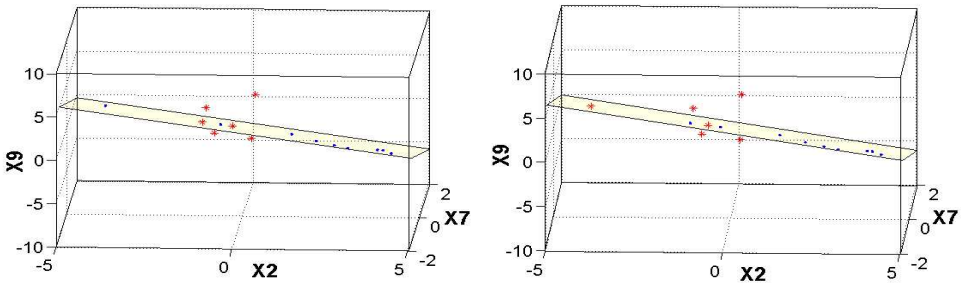
Applying Property 4, we can get the following proposition.

Proposition 1. *Let Ω' be a submatrix of Ω and $C_{\Omega'}$ be the $m \times m$ covariance matrix of Ω'. Let $\{a_i\}$ be the diagonal entries of $C_{\Omega'}$ and arranged in increasing order. If $\frac{\Sigma_{t=1}^{k} a_t}{\Sigma_{t=1}^{m} a_t} \leq \eta$, then we have $f(\Omega', k) \leq \eta$, i.e., the feature subset of Ω' is a strongly correlated feature subset.*

The proof of Proposition 1 is straightforward and omitted here. This proposition gives us an upper bound of the objective function value for a given submatrix of Ω. For any submatrix $\Omega' = V \times \{\mathbf{p}_1, \cdots, \mathbf{p}_N\}$ of Ω, we can examine the diagonal entries of the covariance matrix $C_{\Omega'}$ of Ω' to get the upper bound of the objective function. The computational cost of calculating of this upper bound is much less than that of evaluating the objective function value directly by PCA. Therefore, before evaluating the objective function value of a candidate feature subset, we can check the upper bound in Proposition 1. If the upper bound is no greater than the threshold η, then we know that the candidate is a strongly correlated feature subset without performing PCA on its covariance matrix.

4.2 Choosing the Subsets of Points

In the previous subsection, we discussed the procedure of generating candidate feature subsets. A feature subset may be strongly correlated only on a subset of the data points. As discussed in Section 3.1, the monotonicity property does not hold for the point subsets. Therefore, some heuristic must be used in order



(a) Successive point deletion (b) Distance-based point deletion

Fig. 6. Points deleted using different heuristics

to avoid performing PCA on all possible subsets of points for each candidate feature subset. In this subsection, we discuss the heuristics that can be used for choosing the subset of points.

A successive point deletion heuristic For a given candidate feature subset, if it is not strongly correlated on all data points, we can delete the points successively in the following way.

Suppose that $\Omega' = \{\mathbf{f}_{i_1}, \cdots, \mathbf{f}_{i_m}\} \times \{\mathbf{p}_1, \cdots, \mathbf{p}_N\}$ is a submatrix of Ω and $f(\Omega', k) > \eta$, i.e., the features of Ω' is not strongly correlated on all data points. Let $\Omega'_{\setminus \mathbf{p}_a}$ be the submatrix of Ω' by deleting point $\mathbf{p}_a$ ($\mathbf{p}_a \in \{\mathbf{p}_1, \cdots, \mathbf{p}_N\}$) from Ω'. This heuristic deletes the point $\mathbf{p}_a$ from Ω' such that $f(\Omega'_{\setminus \mathbf{p}_a}, k)$ has the smallest value comparing to deleting any other point. We keep deleting points until the number of points in the submatrix reaches the ratio $n/N = \delta$ or the feature subset of Ω' turns out to be strongly correlated on the current point subset.

This is a successive greedy point deletion heuristic. In each iteration, it deletes the point that leads to the most reduction in the objective function value. This heuristic is time consuming, since in order to delete one point from a submatrix containing n points, we need to calculate the objective function value n times in order to find the smallest value.

A distance-based point deletion heuristic In this subsection, we discuss the heuristic used by CARE. It avoids calculating objective function value n times for deleting a single point from a submatrix containing n points.

Suppose that $\Omega' = \{\mathbf{f}_{i_1}, \cdots, \mathbf{f}_{i_m}\} \times \{\mathbf{p}_1, \cdots, \mathbf{p}_N\}$ is a submatrix of Ω and $f(\Omega', k) > \eta$, i.e., the features of Ω' is not strongly correlated on all data points. Let S_1 be the subspace spanned by the k eigenvectors with the smallest eigenvalues and the S_2 be the subspace spanned by the remaining $m-k$ eigenvectors. For each point $\mathbf{p}_a$ ($\mathbf{p}_a \in \{\mathbf{p}_1, \cdots, \mathbf{p}_N\}$), we calculate two distances: d_{a_1} and d_{a_2}. d_{a_1} is the distance between $\mathbf{p}_a$ and the origin in sub-eigenspace S_1 and d_{a_2} is

the distance between $\mathbf{p}_a$ and the origin in sub-eigenspace S_2. Let the distance ratio $r_{\mathbf{p}_a} = d_{a_1}/d_{a_2}$. We sort the points according to their distance ratios and delete $(1-\delta)N$ points that have the largest distance ratios.

The intuition behind this heuristic is that we try to reduce the variance in subspace S_1 as much as possible while retaining the variance in S_2.

Using the running dataset shown in Figure 5, for feature subset $\{\mathbf{f}_2, \mathbf{f}_7, \mathbf{f}_9\}$, the deleted points are shown as red stars in Figures 6(a) and 6(b) using the two different heuristics described above. The reestablished linear correlations are $2\mathbf{f}_2 + 5.9\mathbf{f}_7 + 3.8\mathbf{f}_9 = 0$ (successive), and $2\mathbf{f}_2 + 6.5\mathbf{f}_7 + 2.9\mathbf{f}_9 = 0$ (distance-based). Note that the embedded linear correlation is $2\mathbf{f}_2 + 6\mathbf{f}_7 + 3\mathbf{f}_9 = 0$. As we can see from the figures, both methods choose almost the same point subsets and correctly reestablish the embedded linear correlation.

The distance-based heuristic is more efficient than the successive approach since it does not have to evaluate the value of the objective function many times for each deleted point.

As a summary of Section 4, CARE adopts the depth-first search strategy to enumerate the candidate feature subsets. If a candidate feature subset is not strongly correlated on all data points, then CARE applies the distance-based point deletion heuristic to find the subset of points on which the candidate feature subset may have stronger correlation. If a candidate turns out to be a linear reducible subspace, then all its supersets can be pruned.

5 The REDUS Algorithm

In this section, we present REDUS algorithm which detects the (non-linar) maximum reducible subspaces (Definition 6). We first give a short introduction to the intrinsic dimensionality estimator. Then we present the algorithms for finding the overall reducible subspace and the maximum reducible subspace respectively .

5.1 Intrinsic Dimensionality Estimator

To find the overall reducible subspace in the dataset, we adopt correlation dimension [39, 42], which can measure both linear and nonlinear intrinsic dimensionality, as our intrinsic dimensionality estimator since it is computationally more efficient than other estimators while its quality of estimation is similar to others. In practice, we observe that correlation dimension satisfies Properties 2 and 3, although we do not provide the proof here. In what follows, we give a brief introduction of correlation dimension.

Let Y be a feature subspace of the dataset, i.e., $Y \subseteq \Omega_F$. Suppose that the number of points N in the dataset approaches infinity. Let $dis(p_i, p_j, Y)$ represent the distance between two data points p_i and p_j in feature subspace Y. Let $B_Y(p_i, r)$ be the subset of points contained in a ball of radius r centered at point p_i in subspace Y, i.e.,

$$B_Y(p_i, r) = \{p_j | p_j \in \Omega_P, dis(p_i, p_j, Y) \leq r\}.$$

The average fraction of pairs of data points within distance r is

$$C_Y(r) = \lim_{N\to\infty} \frac{1}{N^2} \sum_{p_i \in \Omega_P} |B_Y(p_i, r)|.$$

The *correlation dimension* of Y is then defined as

$$ID(Y) = \lim_{r,r'\to 0} \frac{\log[C_Y(r)/C_Y(r')]}{\log[r/r']}.$$

In practice, N is a finite number. C_Y is estimated using $\frac{1}{N^2} \sum_{p_i \in Y_P} |B(p_i, r)|$. The correlation dimension is the growth rate of the function $C_Y(r)$ in log-log scale, since $\frac{\log[C_Y(r)/C_Y(r')]}{\log[r/r']} = \frac{\log[C_Y(r)] - \log[C_Y(r')]}{\log r - \log r'}$. The correlation dimension is estimated using the slope of the line that best fits the function in least squares sense.

The intuition behind the correlation dimension is following. For points that are arranged on a line, one expects to find twice as many points when doubling the radius. For the points scattered on 2-dimensional plane, when doubling the radius, we expect the number of points to increase quadratically. Generalizing this idea to m-dimensional space, we have $C_Y(r)/C_Y(r') = (r/r')^m$. Therefore, the intrinsic dimensionality of feature subspace Y can be simply treated as the growth rate of the function $C_Y(r)$ in log-log scale.

5.2 Finding Overall Reducible Subspace

The following theorem sets the foundation for the efficient algorithm to find the overall reducible subspace.

Theorem 2. *Suppose that $Y \subseteq \Omega_F$ is a maximum reducible subspace and $V \subset Y$ is its core space. We have $\forall U \subset Y$ ($|U| = |V|$), U is also a core space of Y.*

Proof. We need to show that U satisfies the criteria in Definition 6. Let $V = \{f_{v_1}, f_{v_2}, \cdots, f_{v_m}\}$ and $U = \{f_{u_1}, f_{u_2}, \cdots, f_{u_m}\}$.

Since $U \subset Y$, from the definition of reducible subspace, U is non-redundant, and for every $f_{u_i} \in U$, $\Delta ID(V, f_{u_i}) \leq \epsilon$. For every $f_a \in Y$, we have $\Delta ID(V, f_a) \leq \epsilon$. Thus from Property 3, we have $\Delta ID(U, f_a) \leq \epsilon$. Similarly, for every $f_b \notin Y$, $\Delta ID(V, f_b) > \epsilon$. Thus $\Delta ID(U, f_a) > \epsilon$.

Therefore, U is also a core space of Y.

Theorem 2 tells us that any subset $U \subset Y$ of size $|V|$ is also a core space of Y.

Suppose that $\{Y_1, Y_2, \cdots, Y_S\}$ is the set of all maximum reducible subspaces in the dataset and the overall reducible subspace is $OR = \bigcup_{i=1}^{S} Y_i$. To find OR, we can apply the following method. For every $f_a \in \Omega_F$, let $RF_{f_a} = \{f_b | f_b \in \Omega_F, b \neq a\}$ be the remaining features in the dataset. We calculate $\Delta ID(RF_{f_a}, f_a)$. The overall reducible subspace $OR = \{f_a | \Delta ID(RF_{f_a}, f_a) \leq \epsilon\}$.We now prove the correctness of this method.

Algorithm 1: REDUS

Input: Dataset Ω, input parameters ϵ, n, and τ,
Output: Y: the set of all maximum reducible subspaces

1 $OR = \emptyset$;
2 **for** *each* $f_a \in \Omega_F$ **do**
3 $\quad RF_{f_a} = \{f_b | f_b \in \Omega_F, b \neq a\}$;
4 $\quad$ **if** $\Delta ID(RF_{f_a}, f_a) \leq \epsilon$ **then**
5 $\quad\quad OR = OR \cup \{f_a\}$;
6 $\quad$ **end**
7 **end**
8 sample n points $P = \{p_{s_1}, p_{s_2}, \cdots, p_{s_n}\}$ from Ω.
9 **for** $d = 1$ *to* $|OR|$ **do**
10 $\quad$ **for** *each candidate core space* $C \subset OR$ $(|C| = d)$ **do**
11 $\quad\quad T = \{f_a | f_a$ is strongly correlated with C, $f_a \in OR, f_a \notin C\}$;
12 $\quad\quad$ **if** $T \neq \emptyset$ **then**
13 $\quad\quad\quad Y \leftarrow T$;
14 $\quad\quad\quad$ update OR by removing from OR the features in T;
15 $\quad\quad$ **end**
16 $\quad$ **end**
17 **end**
18 return Y.

Corollary 2. $OR = \{f_a | \Delta ID(RF_{f_a}, f_a) \leq \epsilon\}$.

Proof. Let f_y be an arbitrary feature in the overall reducible subspace. From Theorem 2, we have $\forall f_y \in Y_i \subseteq OR$, $\exists V_i \subset Y_i$ $(f_y \notin V_i)$, such that V_i is the core space of Y_i. Thus $\Delta ID(V_i, f_y) \leq \epsilon$. Since $f_y \notin V_i$, we have $V_i \subseteq RF_{f_y}$. From Property 2, we have $\Delta ID(RF_{f_y}, f_y) \leq \epsilon$.

Similarly, if $f_y \notin OR$, then $\Delta ID(RF_{f_y}, f_y) > \epsilon$.

Therefore, we have $OR = \{f_y | \Delta ID(RF_{f_y}, f_y) \leq \epsilon\}$.

The algorithm for finding the overall reducible subspace is shown in Algorithm 1 from Line 1 to Line 7. Note that the procedure of finding overall reducible subspace is linear to the number of features in the dataset.

5.3 Maximum Reducible Subspace

In this section, we present the second component of REDUS, i.e., identifying the maximum reducible subspaces from the overall reducible subspace found in the previous section.

Intrinsic Dimensionality Based Method From Definition 6 and Theorem 2, we have the following property concerning the reducible subspaces.

Corollary 3. *Let $Y_i \subseteq OR$ be a maximum reducible subspace, and $V_i \subset Y_i$ be any core space of Y_i. We have*

$$Y_i = \{f_a | \Delta ID(V_i, f_a) \leq \epsilon, f_a \in OR\}.$$

Therefore, to find the individual maximum reducible subspaces $Y_i \subseteq OR$ ($1 \leq i \leq S$), we can use any core space $V_i \subset Y_i$ to find the other features in Y_i. More specifically, a *candidate* core space of size d is a feature subset $C \subset OR$ ($|C| = d$). From size $d = 1$ to $|OR|$, for each candidate core space, let $T = \{f_a | \Delta ID(C, f_a) \leq \epsilon, f_a \in OR, f_a \notin C\}$. If $T \neq \emptyset$, then T is a maximum reducible subspace with core space of size d. The overall reducible subspace OR is then updated by removing the features in T. Note that the size of $|OR|$ decreases whenever some maximum reducible subspace is identified. We now prove the correctness of this method.

Corollary 4. *Any candidate core space is non-redundant.*

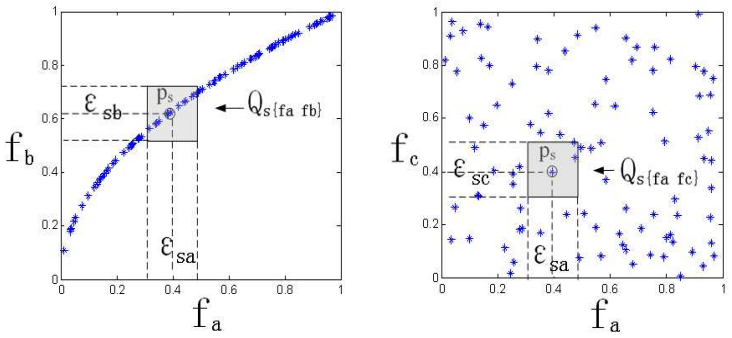
Proof. It is easy to see any candidate core space of size 1 is non-redundant. Now, assume that all candidate core spaces of size $d-1$ are non-redundant, we show all candidate core spaces of size d are non-redundant. We prove this by contradiction.

Let $V = \{f_{v_1}, f_{v_2}, \cdots, f_{v_d}\}$ be an arbitrary candidate core space of size d. Without loss of generality, assume that f_d is the redundant feature in V. Let $V' = \{f_1, f_2 \cdots, f_{v_{d-1}}\}$. We have $\Delta ID(V', f_{v_d}) \leq \epsilon$. Since $|V'| = d - 1$, V' is non-redundant according to the assumption. Moreover, we have $T = \{f_a | \Delta ID(V', f_a) \leq \epsilon, f_a \in OR, f_a \notin V'\} \neq \emptyset$, since $f_{v_d} \in T$. Therefore, $f_{v_d} \in T$ would have been removed from OR before the size of the candidate core spaces reaches d. This contradicts the assumption of f_{v_d} being in the candidate core space V. Therefore, we have that any candidate core space is non-redundant.

Corollary 5. *Let C be a candidate core space. If $\exists f_a \in OR$ such that $\Delta ID(C, f_a) \leq \epsilon$, then C is a true core space of some maximum reducible subspace in OR.*

Proof. Let $Y = \{f_y | \Delta ID(C, f_y) \leq \epsilon, f_y \in OR\}$. Following the process of finding OR, we know that Y includes all and only the features in Ω_F that are strongly correlated with C. Thus $\exists C \subset Y$, such that C satisfies Criterion (1) in Definition 5, and Criterion (2) in Definition 6. Moreover, according to Corollary 4, C is non-redundant. Hence C also satisfies Criterion (2) of Definition 5. Thus Y is a maximum reducible subspace with core space C.

In this method, for each candidate core space, we need to calculate $\Delta ID(C)$ and $\Delta ID(C \cup \{f_a\})$ for every $f_a \in OR$ in order to get the value of $\Delta ID(C, f_a)$. However, the intrinsic dimensionality calculation is computationally expensive. Since the intrinsic dimensionality estimation is inherently approximate, we propose in the following section a method utilizing the point distribution in feature subspaces to distinguish whether a feature is strongly correlated with a core space.



(a) strongly correlated features (b) uncorrelated features

Fig. 7. Point distributions in correlated feature subspace and uncorrelated feature subspace

Point Distribution Based Method After finding the overall reducible subspace OR, we can apply the following heuristic to examine if a feature is strongly correlated with a feature subspace. The intuition behind our heuristic is similar to the one behind the correlation dimension.

Assume that the number of data points N in the dataset approaches infinity, and the features in the dataset are normalized so that the points are distributed from 0 to 1 in each dimension. Let $p_s \in \Omega_P$ be an arbitrary point in the dataset, and $0 < l < 1$ be a natural number. Let ξ_{sy} represent the interval of length l on feature f_y centered at p_s. The expected number of points within the interval ξ_{sy} is lN. For d features $C = \{f_{c_1}, f_{c_2}, \cdots, f_{c_d}\}$, let Q_{sC} be the d-dimensional hypercube formed by the intervals ξ_{sc_i} ($f_{c_i} \in C$). If the d features in C are totally uncorrelated, then the expected number of points in Q_{sC} is $l^d N$. Let f_m be another feature in the dataset, and $C' = \{f_{c_1}, f_{c_2}, \cdots, f_{c_d}, f_m\}$. If f_m is determined by $\{f_{c_1}, f_{c_2}, \cdots, f_{c_d}\}$, i.e., f_m is strongly correlated with C, then C' has intrinsic dimensionality d. The expected number of points in the d-dimensional hypercube, $Q_{sC'}$, which is embedded in the $(d+1)$-dimensional space of C', is still $l^d N$. If, on the other hand, f_m is uncorrelated with any feature subspace of $\{f_{c_1}, f_{c_2}, \cdots, f_{c_d}\}$, then C' has dimensionality $d+1$, and the expected number of points in the $(d+1)$-dimensional hypercube $Q_{sC'}$ is $l^{(d+1)}N$. The difference between the number of points in the cubes of these two cases is $l^d(1-l)N$.

Figure 7(a) and 7(b) show two examples on 2-dimensional spaces. In both examples, $d = 1$ and $C = \{f_a\}$. In Figure 7(a), feature f_b is strongly correlated with f_a. Feature f_c is uncorrelated with f_a, as shown in Figure 7(b). The randomly sampled point p_s is at the center of the cubes $Q_{s\{f_a,f_b\}}$ and $Q_{s\{f_a,f_c\}}$. The point density in cube $Q_{s\{f_a,f_b\}}$ is clearly much higher than the point density in cube $Q_{s\{f_a,f_c\}}$ due to the strong correlation between f_a and f_b.

Therefore, for each candidate core space, we can check if a feature is correlated with it in the following way. We randomly sample n points $P = \{p_{s_1}, p_{s_2}, \cdots, p_{s_n}\}$

from the dataset. Suppose that $C = \{f_{c_1}, f_{c_2}, \cdots, f_{c_d}\}$ is the current candidate core space. For feature $f_a \in OR$ ($f_a \notin C$), let $C' = \{f_{c_1}, f_{c_2}, \cdots, f_{c_d}, f_a\}$. Let $\delta_{s_iC'}$ represent the number of points in the cube $Q_{s_iC'}$. $P' = \{p_{s_i} | \delta_{s_iC'} \geq l^{(d+1)}N\}$ is the subset of the sampled points such that the cube centered at them have more points than expected if f_a is uncorrelated with C. We say f_a is strongly correlated with C if $\frac{|P'|}{|P|} \geq \tau$, where τ is a threshold close to 1.

Concerning the choice of l, we can apply the following reasoning. If we let $l = (\frac{1}{N})^{\frac{1}{d+1}}$, then the expected number of points in the cube $Q_{s_iC'}$ is 1, if f_a is uncorrelated with C. If f_a is correlated with C, then the expected number of points in the cube $Q_{s_iC'}$ is greater than 1. In this way, we can set l according to the size of the candidate core space.

The second step of REDUS is shown in Algorithm 1 from Line 8 to Line 18. Note that in the worst case, the algorithm needs to enumerate all possible feature subspaces. However, in practice, the algorithm is very efficient since once an individual reducible subspace is found, all its features are removed. Only the remaining features need to be further examined.

6 Experiments

In this section, we present the experimental results of CARE and REDUS on both synthetic and real datasets. Both algorithms are implemented using Matlab 7.0.4. The experiments are performed on a 2.4 GHz PC with 1G memory running WindowsXP system.

6.1 Synthetic Data

We evaluate CARE and REDUS on different synthetic datasets.

CARE .

To evaluate the effectiveness of the CARE, we generate a synthetic dataset of 100 features and 120 points in the following way. The dataset is first populated with randomly generated points for each one of the 100 features. Then we embedded three local linear correlations into the dataset as described in Table 4. For example, on points $\{\mathbf{p}_1, \cdots, \mathbf{p}_{60}\}$ we create local linear correlation $\mathbf{f}_{50} - \mathbf{f}_{20} + 0.5\mathbf{f}_{60} = 0$. Gaussian noise with mean 0 and variance 0.01 is added into the dataset.

Point subsets	Local linear correlations
$\{\mathbf{p}_1, \cdots, \mathbf{p}_{60}\}$	$\mathbf{f}_{50} - \mathbf{f}_{20} + 0.5\mathbf{f}_{60} = 0$
$\{\mathbf{p}_{30}, \cdots, \mathbf{p}_{90}\}$	$\mathbf{f}_{40} - \mathbf{f}_{30} + 0.8\mathbf{f}_{80} - 0.5\mathbf{f}_{10} = 0$
$\{\mathbf{p}_{50}, \cdots, \mathbf{p}_{110}\}$	$\mathbf{f}_{15} - \mathbf{f}_{25} + 1.5\mathbf{f}_{45} - 0.3\mathbf{f}_{95} = 0$

Table 4. Local linear correlations embedded in the dataset

We first show the comparison of CARE and full dimensional PCA. We perform PCA on the synthetic dataset described above. To present the linear correlation discovered by PCA, we show the resulting hyperplanes determined by the three eigenvectors with the smallest eigenvalues. Each such hyperplane represents a linear correlation of all the features in the dataset. Due to the large number of features, we only show the features with coefficients with absolute values greater than 0.2. The linear correlations reestablished by full dimensional PCA are shown in Table 5. Clearly, these are not the local linear correlations embedded in the dataset.

Table 6 shows the local linear correlations reestablished by CARE, with $k = 1$, $\eta = 0.006$, $\delta = 50\%$, and $max_s = 4$. As can be seen from the table, CARE correctly identifies the correlations embedded in the dataset.

Eigenvectors	Linear correlations reestablished
$\mathbf{v}_1$ ($\lambda_1 = 0.0077$)	$0.23\mathbf{f}_{22} - 0.25\mathbf{f}_{32} - 0.26\mathbf{f}_{59} \approx 0$
$\mathbf{v}_2$ ($\lambda_2 = 0.0116$)	$0.21\mathbf{f}_{34} - 0.26\mathbf{f}_{52} \approx 0$
$\mathbf{v}_3$ ($\lambda_3 = 0.0174$)	$-0.22\mathbf{f}_6 - 0.29\mathbf{f}_8 + 0.22\mathbf{f}_{39}$ $-0.23\mathbf{f}_{72} + 0.26\mathbf{f}_{93} \approx 0$

Table 5. Linear correlations identified by full dimensional PCA

$\mathbf{f}_{50} - 0.99\mathbf{f}_{20} + 0.42\mathbf{f}_{60} = 0$
$\mathbf{f}_{40} - 0.97\mathbf{f}_{30} + 0.83\mathbf{f}_{80} - 0.47\mathbf{f}_{10} = 0$
$\mathbf{f}_{15} - 0.9\mathbf{f}_{25} + 1.49\mathbf{f}_{45} - 0.33\mathbf{f}_{95} = 0$

Table 6. Local linear correlations identified by CARE

Figure 8 shows the hyperplane representation of the local linear correlation, $\mathbf{f}_{40} - 0.97\mathbf{f}_{30} + 0.83\mathbf{f}_{80} - 0.47\mathbf{f}_{10} = 0$, reestablished by CARE. Since this is a 3-dimensional hyperplane in 4-dimensional space, we visualize it as a 2-dimensional hyperplane in 3-dimensional space by creating a new feature $(-0.83\mathbf{f}_{80}+0.47\mathbf{f}_{10})$. As we can see from the figure, the data points are not clustered on the hyperplane even though the feature subsets are strongly correlated. The existing projected clustering algorithms [10, 11] try to find the points that are close to each other in the projected space. Therefore, they can not find the strongly correlated feature subset as shown in this figure.

To evaluate the efficiency of CARE, we generate synthetic datasets as follows. Each synthetic dataset has up to 500K points and 60 features, in which 40 linear correlations are embedded. Gaussian noise with mean 0 and variance 0.01 is added into the dataset. The default dataset for efficiency evaluation contains 5000 points and 60 features if not specified otherwise. The default values for the parameters are: $k = 1$, $\eta = 0.006$, $\delta = 50\%$, and $max_s = 4$.

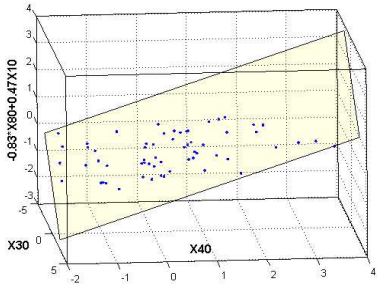
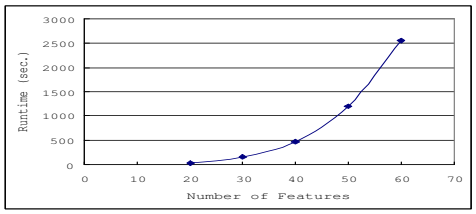
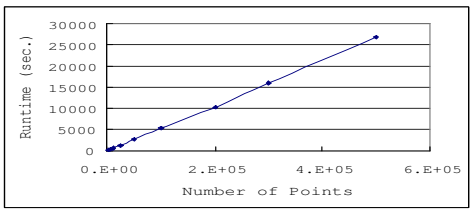


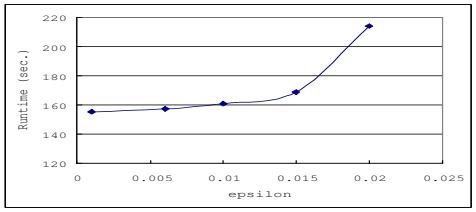
Fig. 8. The hyperplane representation of a local linear correlation reestablished by CARE



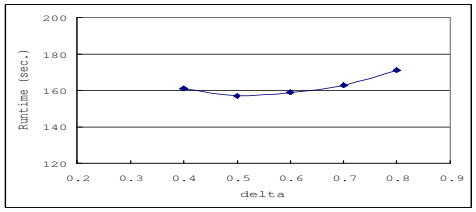
(a) Varying number of features



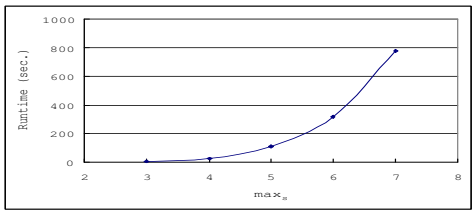
(b) Varying number of data points



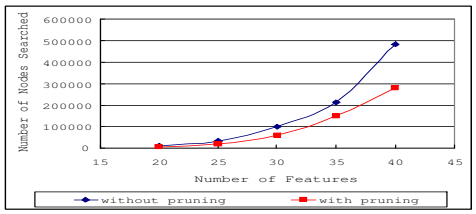
(c) Varying η



(d) Varying δ



(e) Varying max_s



(f) Pruning effect of the upper bound of the objective function value

Fig. 9. CARE Efficiency evaluation

Figures 9(a) to 9(f) show the efficiency evaluation results. Figure 9(a) shows that the running time of CARE is roughly quadratic to the number of features in the dataset. Note that the theoretical worst case should be exponential when the

algorithm has to check every subset of the features and data points. Figure 9(b) shows the scalability of CARE with respect to the number of points when the dataset contains 30 features. The running time of CARE is linear to the number of data points in the dataset as shown in the figure. This is due to the distance-based point deletion heuristic. As we can see from the figure, CARE finishes within reasonable amount of time for large datasets. However, since CARE scales roughly quadratically to the number of features, the actual runtime of CARE mostly depends on the number of features in the dataset.

Figure 9(c) shows that the runtime of CARE increases steadily until η reaches certain threshold. This is because the higher the value of η, the weaker the correlations identified. After certain point, too many weak correlations meet the criteria will be identified. Figure 9(d) demonstrates that CARE's runtime when varying δ. Figure 9(e) shows CARE's runtime with respect to different max_s when the datasets contain 20 features.

Figure 9(f) shows the number of patterns evaluated by CARE before and after applying the upper bound of the objective function value discussed in Section 4.

REDUS .

As shown in Algorithm 1, REDUS generally requires three input parameters: ϵ, n, and τ. In the first step of finding the overall reducible subspace, ϵ is the threshold to filter out the irrelevant features. Since features strongly correlated with some core space can only change intrinsic dimensionality a small amount, the value of ϵ should be close to 0. According to our experience, a good starting point is 0.1. After finding the reducible subspaces, the user can apply the standard dimensionality reduction methods to see if the are really correlated, and the adjust ϵ value accordingly to find stronger or weaker correlations in the subspaces. In all our experiments, we set ϵ between 0.002 to 0.25. In the second step, n is the point sampling size and τ is the threshold to determine if a feature is strongly correlated with a candidate core space. In our experiments, n is set to be 10% of the total number of data points in the dataset, and τ is set to be 90%.

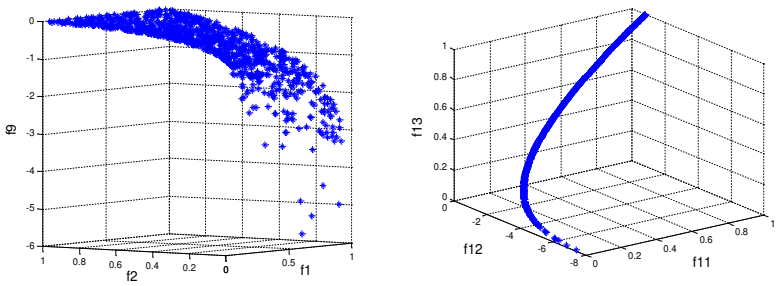
We generate two synthetic datasets.

REDUS Synthetic dataset 1: The first synthetic dataset is as shown in Figure 3. There are 12 features, $\{f_1, f_2, \cdots, f_{12}\}$, and 1000 data points in the dataset. 3 reducible subspaces: a 2-dimensional Swiss roll, a 1-dimensional helix-shaped line, and a 2-dimensional plane, are embedded in different 3-dimensional spaces respectively. The overall reducible subspace is $\{f_1, f_2, \cdots, f_9\}$. Let c_i $(1 \leq i \leq 4)$ represent constants and r_j $(1 \leq j \leq 3)$ represent random vectors. The generating function of the Swiss roll is: $t = \frac{3}{2}\pi(1 + 2r_1)$, $s = 21r_2$, $f_1 = t\cos(t)$, $f_2 = s$, $f_3 = t\sin(t)$. The roll is then rotated 45° counter clockwise on feature space $\{f_2, f_3\}$. The helix-shaped line is generated by: $f_4 = c_1 r_3$, $f_5 = c_2 \sin(r_3)$, $f_6 = c_2 \cos(r_3)$. The 2-dimensional plane is generated by $f_9 = c_3 f_7 + c_4 f_8$. The

remaining 3 features $\{f_{10}, f_{11}, f_{12}\}$ are random vectors consisting of noise data points.

In the first step, with $\epsilon = 0.25$, REDUS successfully uncovers the overall reducible space. The parameter setting for the second step is $\tau = 90\%$, and point sampling size 10%. We run REDUS 10 times. In all 10 runs, REDUS successfully identifies the individual maximum reducible subspaces from the overall reducible subspace.

REDUS Synthetic dataset 2: We generate another larger synthetic dataset as follows. There are 50 features, $\{f_1, f_2, \cdots, f_{50}\}$ and 1000 data points in the dataset. There are 3 reducible subspaces: $Y_1 = \{f_1, f_2, \cdots, f_{10}\}$ reducible to a 2-dimensional space, $Y_2 = \{f_{11}, f_{12}, \cdots, f_{20}\}$ reducible to a 1-dimensional space, and $Y_3 = \{f_{21}, f_{22}, \cdots, f_{30}\}$ reducible to a 2-dimensional space. The remaining features contain random noises. Figures 10(a) and 10(b) show two examples of the embedded correlations in 3-dimensional subspaces. Figure 10(a) plots the point distribution on feature subspace $\{f_1, f_2, f_9\}$ of Y_1, and Figure 10(b) plots the point distribution on feature subspace $\{f_{11}, f_{12}, f_{13}\}$ of Y_2.



(a) a correlation in Y_1 (b) a correlation in Y_2

Fig. 10. Examples of embedded correlations in synthetic dataset 2

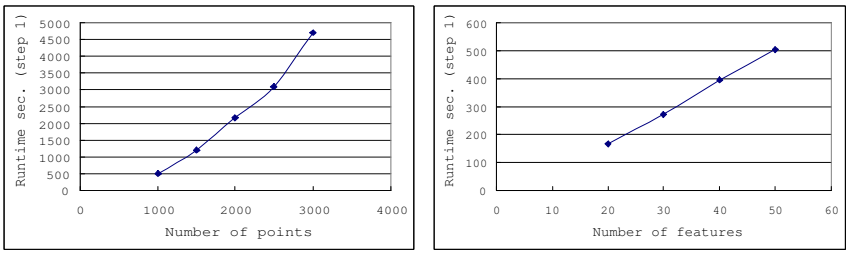
We apply REDUS on this synthetic dataset using various parameter settings. Table 7 shows the accuracy of finding the overall reducible subspace when ϵ taking different values. The recall is defined as $TP/(TP+FN)$, and the precision is defined as $TP/(TP+FP)$, where TP represents the number of true positive, FP represents the number of false positive, and FN represents the number of false negative. As we can see, REDUS is very accurate and robust to ϵ.

To evaluate the efficiency and scalability of REDUS, we apply it to **synthetic dataset 2**. The default dataset for efficiency evaluation contains 1000 points and 50 features if not specified otherwise. The default values for the parameters are the same as before.

Figure 11(a) shows the runtime of finding the overall reducible subspace when varying the number of data points. The runtime scales roughly quadratically. This is because when computing the correlation dimensions, we need to calculate

ϵ	**Precision**	**Recall**
0.06	83%	100%
0.05	91%	100%
0.04	96%	100%
0.03	100%	100%
0.02	100%	100%
0.01	100%	100%
0	100%	90%

Table 7. Accuracy of finding the overall reducible subspace when varying ϵ



(a) Varying number of points (b) Varying number of features

Fig. 11. REDUS Efficiency evaluation of finding the overall reducible subspace

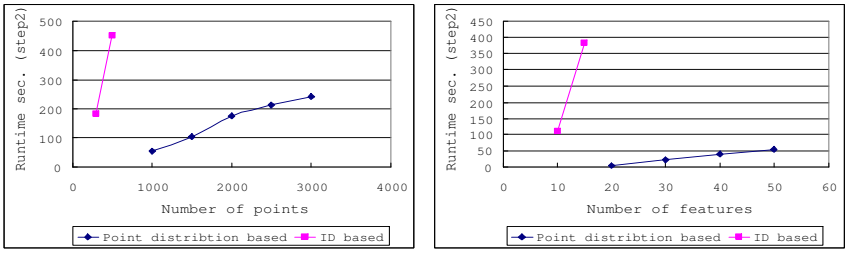
all pairwise distances between the data points, which is clearly quadratic to the number of points.

Figure 11(b) shows that the runtime of finding the overall reducible subspace is linear to the number of features. This is because REDUS only scans every feature once to examine if it is strongly correlated with the subspace of the remaining features. This linear scalability is desirable for the datasets containing a large number of features.

Figures 12(a) and 12(b) show the runtime comparisons between using the correlation dimension as intrinsic dimensionality estimator and the point distribution heuristic to identify the individual maximum reducible subspaces from the overall reducible subspaces. Since the calculation of intrinsic dimensionality is relatively expensive, the program often cannot finish in a reasonable amount of time. Using the point distribution heuristics, on the other hand, is much more efficient and scales linearly to the number of points and features in the dataset.

6.2 Real Data

We apply CARE on the mouse gene expression data provided by the School of Public Health at UNC. The dataset contains the expression values of 220 genes in 42 mouse strains. CARE find 8 strongly correlated gene subsets with parameter setting: $k = 1$, $\eta = 0.002$, $\delta = 50\%$, and $max_s = 4$. Due to the space limit, we show 4 of these 8 gene subsets in Table 8 with their symbols



(a) Varying number of points (b) Varying number of features

Fig. 12. REDUS Efficiency evaluation of identifying maximum reducible subspaces from the overall reducible subspace

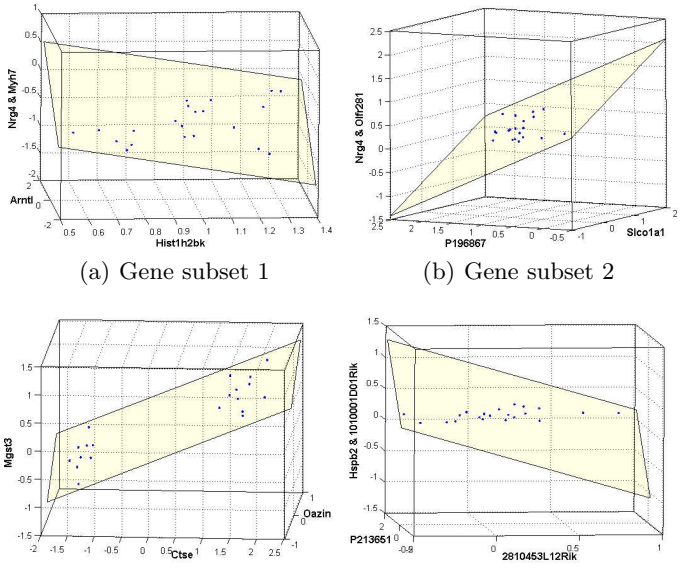
Subsets	Gene IDs	GO annotations
1	Nrg4	cell part
	Myh7	cell part; intracelluar part
	Hist1h2bk	cell part; intracelluar part
	Arntl	cell part; intracelluar part
2	Nrg4	integral to membrane
	Olfr281	integral to membrane
	Slco1a1	integral to membrane
	P196867	N/A
3	Oazin	catalytic activity
	Ctse	catalytic activity
	Mgst3	catalytic activity
4	Hspb2	cellular physiological process
	2810453L12Rik	cellular physiological process
	1010001D01Rik	cellular physiological process
	P213651	N/A

Table 8. Strongly correlated gene subsets

and the corresponding GO annotations. As shown in the table, genes in each gene subset have consistent annotations. We also plot the hyperplanes of these strongly correlated gene subsets in 3-dimensional space in Figures 13(a) to 13(d). As we can see from the figures, the data points are sparsely distributed in the hyperplanes, which again demonstrates CARE can find the groups of highly similar genes which cannot be identified by the existing projected clustering algorithms.

7 Conclusion

In this chapter, we investigate the problem of finding strongly correlated feature subspaces in high dimensional datasets. The correlation can be linear or nonlinear. Such correlations hidden in feature subspace may be invisible to the global



(a) Gene subset 1 (b) Gene subset 2

(c) Gene subset 3 (d) Gene subset 4

Fig. 13. Hyperplane representations of strongly correlated gene subsets

feature transformation methods. Utilizing the concepts of PCA and intrinsic dimensionality, we formalize this problem as the discovery of maximum reducible subspaces in the dataset. Two effective algorithms, CARE and REDUS, are presented to find the reducible subspaces in linear and non-linear cases respectively. The experimental results show that both algorithms can effectively and efficiently find these interesting local correlations. These methods are powerful tools for identifying potential transcriptional modules and thus play an important role in many modeling biological networks.

References

1. M. Eisen, P. Spellman, P. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl. Acad. Sci. USA*, vol. 95, pp. 14 863–68, 1998.
2. V. Iyer and et. al., "The transcriptional program in the response of human fibroblasts to serum," *Science*, vol. 283, pp. 83–87, 1999.
3. L. Parsons, E. Haque, and H. Liu, "Subspae clustering for high dimensional data: a review," *KDD Explorations*, 2004.
4. A. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, pp. 245–271, 1997.
5. H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*. Boston: Kluwer Academic Publishers, 1998.

6. L. Yu and H. Liu, "Feature selection for high-dimensional data: a fast correlation-based filter solution," *Proceedings of International Conference on Machine Learning*, 2003.
7. Z. Zhao and H. Liu, "Searching for interacting features," *the 20th International Joint Conference on AI*, 2007.
8. I. Jolliffe, *Principal component analysis*. New York: Springer, 1986.
9. T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*. Springer Verlag, 1996.
10. C. Aggarwal and P. Yu, "Finding generalized projected clusters in high dimensional spaces," in *SIGMOD*, 2000.
11. E. Achtert, C. Bohm, H.-P. Kriegel, P. Kroger, and A. Zimek, "Deriving quantitative models for correlation clusters," in *KDD*, 2006.
12. H. Wang, W. Wang, J. Yang, and Y. Yu, "Clustering by pattern similarity in large data sets," *SIGMOD*, 2002.
13. M. Ashburner and et al., "Gene ontology: tool for the unification of biology," *The gene ontology consortium, Nat. Genet.*, vol. 25, pp. 25–29, 2000.
14. H. R. Lindman, *Analysis of variance in complex experimental designs*. Wiley-Interscience, 2001.
15. K. Fukunaga, *Introduction to statistical pattern recognition*. Academic Press, San Diego, California, 1990.
16. W. Mendenhall and T. Sincich, *A Second Course in Statistics: Regression Analysis*. Prentice Hall, 2002.
17. S. Yu, K. Yu, H.-P. Kriegel, and M. Wu, "Supervised probabilistic principal component analysis," *KDD*, 2006.
18. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensinal data for data mining applications," *SIGMOD*, 1998.
19. C. Aggarwal, J. Wolf, P. Yu, C. Procopiuc, and J. Park, "Fast algorithms for projected clustering," *SIGMOD*, 1999.
20. C. Chen, A. Fu, and Y. Zhang, "Entropy-based subspace clustering for mining numerical data," *SIGKDD*, 1999.
21. R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge U. K.: Cambridge University Press, 1985.
22. A. Alizadeh and et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–11, 2000.
23. D. Barbara and P. Chen. Using the fractal dimension to cluster datasets. *KDD*, 2000.
24. M. Belkin and P. Niyogi. laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 2003.
25. A. Belussi and C. Faloutsos. Self-spacial join selectivity estimation using fractal concepts. *ACM Transactions on Information Systems*, 16(2):161–201, 1998.
26. I. Borg and P. Groenen. *Modern multidimensional scaling*. New York: Springer, 1997.
27. F. Camastra and A. Vinciarelli. Estimating intrinsic dimension of data with a fractal-based approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(10):1404–1407, 2002.
28. T. M. Cover and J. A. Thomas. *The Elements of Information Theory*. Wiley & Sons, New York, 1991.
29. C. Faloutsos and I. Kamel. Beyond uniformity and independence: analysis of r-trees using the concept of fractal dimension. *PODS*, 1994.

30. K. Fukunaga. Intrinsic dimensionality extraction. *Classification, Pattern recognnition and Reduction of Dimensionality, Volume 2 of Handbook of Statistics*, pages 347–360, P. R. Krishnaiah and L. N. Kanal eds., Amsterdam, North Holland, 1982.
31. K. Fukunaga and D. R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, 20(2):165–171, 1976.
32. A. Gionis, A. Hinneburg, S. Papadimitriou, and P. Tsaparas. Dimension induced clustering. *KDD*, 2005.
33. G. Golub and A. Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, Maryland, 1996.
34. M. Kendall and J. D. Gibbons. *Rank Correlation Methods*. New York: Oxford University Press, 1990.
35. D. C. Lay. *Linear Algebra and Its Applications*. Addison Wesley, 2005.
36. E. Levina and P. J. Bickel. Maximum likelihood estimation of intrinsic dimension. *Advances in Neural Information Processing Systems*, 2005.
37. B.-U. Pagel, F. Korn, and C. Faloutsos. Deflating the dimensionality curse using multiple fractal dimensions. *ICDE*, 2000.
38. S. Papadimitriou, H. Kitawaga, P. B. Gibbons, and C. Faloutsos. Loci: Fast outlier detection using the local correlation integral. *ICDE*, 2003.
39. S. N. Rasband. *Chaotic Dynamics of Nonlinear Systems*. Wiley-Interscience, 1990.
40. H. T. Reynolds. *The analysis of cross-classifications*. The Free Press, New York, 1977.
41. S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290 (5500):2323–2326, 2000.
42. M. Schroeder. *Fractals, Chaos, Power Lawers: Minutes from an Infinite Paradise*. W. H. Freeman, New York, 1991.
43. J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290 (5500):2319–2323, 2000.
44. A. K. H. Tung, X. Xin, and B. C. Ooi. Curler: Finding and visualizing nonlinear correlation. *SIGMOD*, 2005.
45. X. Zhang, F. Pan, and W. Wang. Care: Finding local linear correlations in high dimensional data. *ICDE*, 2008.