SLICE: A Novel Method to Find Local Linear Correlations by Constructing Hyperplanes*

Liang Tang[1], Changjie Tang[1], Lei Duan[1], Yexi Jiang[1], Jun Zhu[2]
[1] School of Computer Science, Sichuan University
610065 Chengdu, China
[2] National Center for Birth Defects Monitoring
610041 Chengdu, China
{tangliang, tangchangjie}@cs.scu.edu.cn

Abstract. Finding linear correlations in dataset is an important data mining task, which can be widely applied in the real world. Existing correlation clustering methods combine clustering with PCA to find correlation clusters in dataset. These methods may miss some correlations when instances are sparsely distributed. Previous studies are limited to find the primary linear correlation of the dataset. However, there may be some interesting local linear correlations exist in data subsets. This paper develops an efficient approach to seek multiple local linear correlations in dataset. The main contributions of this paper are: (1) analyzing the limitations of applying current methods on finding linear correlations in data subsets; (2) developing a novel algorithm, SLICE (significant local linear correlation searching), to find multiple local linear correlations in data subsets. The basic idea of SLICE is using a heuristic to construct hyperplanes that represent linear correlations; (3) conducting extensive experiments to show that SLICE is effective to find correct correlations in both synthetic and real-world datasets.

Keywords: Data Mining, Linear Correlation, Principal Component Analysis

1 Introduction

Linear correlations reveal the linear dependencies among several features in a dataset. Finding theses correlations is an interesting research topic, since it has many real-world applications. For example, in sensor network, a latent pattern, which is the linear correlation of multiple time series data received, can be used to detect the data evolution [12].

Principal component analysis (PCA) is able to capture linear correlations in a dataset [4]. It involves the computation of eigenvalue and eigenvectors of a data

* This paper has been accepted by the joint International Conferences on Asia-Pacific Web Conference (APWeb) and Web-Age Information Management (WAIM) 2009, and will appear in April, 2009.
This work was supported by the National Natural Science Foundation of China under grant No. 60773169 and the 11th Five Years Key Programs for Sci. &Tech. Development of China under grant No. 2006BAI05A01.

covariance matrix or singular value decomposition of a data matrix [4]. PCA assumes that all instances in a dataset are in the same correlation. However, instances collected from the real world may have different characteristics, so the linear dependencies among features may be different in different data subsets. For example, analysis on gene expression values in microarray data shows that some linear dependency among gene data only exists under certain situation [10]. In this case, it is not appropriate to apply PCA to analyze the dataset, since only certain part of the dataset shows the linear correlation.

Correlation clustering methods, such as 4C, try to seek clusters in a linear correlation [10, 11]. The most difference between correlation clustering and our work is that correlation clustering seeks the linear correlations of clusters, while the goal of our work is to seek linear correlations of tuples.

In [1], a method called CARE is proposed to find local linear correlations. This method adopts PCA to analyze the linear correlations on both feature subsets and instance subsets. CARE focuses on finding the primary linear correlations from the whole dataset. It is not suitable to find linear correlations that exist in data subsets.

This study focuses on finding multiple linear correlations in data subsets. We refer such correlations as *local linear correlations*. The main challenge for us is that the number of data subsets is so large that it is hard to enumerate all subsets in reasonable runtime. Thus, we design a method to search linear correlations by constructing hyperplanes. The time complexity of our proposed method is polynomial.

Our Contributions: (1) analyzing the limitations of applying current methods on finding linear correlations in data subsets; (2) developing a heuristic algorithm SLICE (**s**ignificant **l**ocal l**i**near **c**orrelation s**e**arching) to find multiple local linear correlations in data subsets. The basic idea of our method is using a heuristic to construct hyperplanes that represent linear correlations; (3) conducting extensive experiments to show that SLICE is effective to find correct correlations in both synthetic and real-world datasets.

Paper organization: the rest of this paper is organized as follows. Section 2 discusses related works. Section 3 states the concept of significant local linear correlation, and describes our method of searching significant local linear correlations. Section 4 comparatively analyzes the experimental results of our method and related methods. Finally, this Section 5 discusses future works, and concluding remarks.

2 Related Work

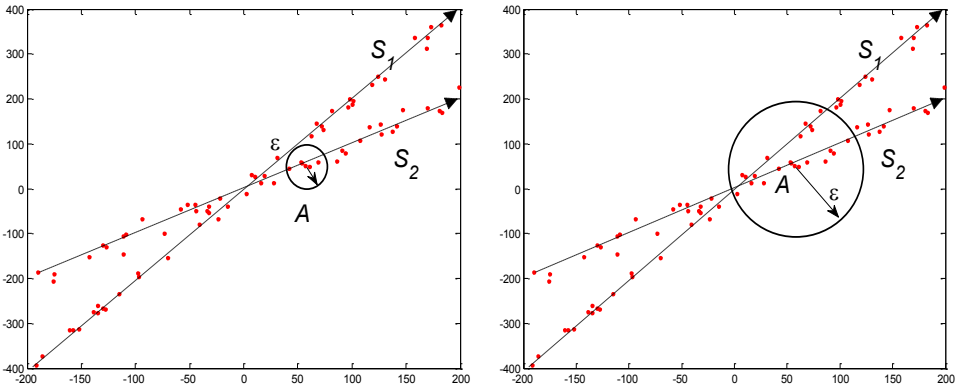
There are three previous studies related to this paper: PCA, correlation clustering, and CARE. We discuss them below.

Principal components analysis (PCA) is a useful method to detect linear correlations [1, 10, 12, 13]. PCA transforms the original data space to another orthogonal data space. Given a data matrix, PCA finds the eigenvectors and eigenvalues of that data matrix. The eigenvectors represent the directions with maximal variances of the data by performing singular value decomposition (SVD) to the data matrix [4]. There are other features transformation methods can be used, such as linear regression analysis (LRA) [5], linear discriminate analysis (LDA) [6],

principal component regression (PCR) [7], and supervise probabilistic PCA (SPPCA) [8].

Some clustering methods are developed to find data clusters in subspaces [2, 3, 9, 13]. They adopt PCA to project data points in some subspace, where the points of a cluster are close to each other. Recently, some correlations clustering methods are proposed to seek the linear correlations of clusters [10, 11]. One typical correlation clustering algorithm is 4C [10], which generates ε–neighborhoods first. However, it is hard to generate high quality ε–neighborhoods to find linear correlations, when the distribution of instances in the correlations is sparse. Figure 1 shows an example of this drawback of 4C.

Example 1. Figure 1 shows an example of a drawback of correlation clustering. Let S_1, S_2 be two local linear correlations, A be a ε–neighborhood. The points on S_1 and S_2 are distributed sparsely. If the value of parameter ε in algorithm 4C is small, few ε–neighborhoods can be generated. This situation is shown as Figure 1 (a). On the other hand, if ε is large, as shown in Figure 1 (b), ε–neighborhoods would contain points in S_1 and S_2, so they don't exhibit any linear dependency. As a result, few correlation clusters could be created in these ε–neighborhoods.



(a) small ε–neighborhood. **(b)** large ε–neighborhood.

Fig. 1 Two kinds of ε–neighborhoods

From Example 1, we can see that it is not suitable to apply 4C to finding local linear correlations.

CARE is a recently proposed method to find local linear correlations in both feature subsets and instance subsets [1]. For the problem of finding correlations in data subsets, it assumes that there is only one correlation in a dataset. So the limitation of CARE is that it can only find the primary linear correlations in dataset. CARE applies PCA to get the hyperplane of the whole dataset at first. Then it adopts a heuristic method to delete the "farthest" point to the hyperplane. This deletion process is repeated until the number of data points equals to the requirement defined by user. CARE may decrease the accuracy of results, since the deleted points may belong to some correct correlations. Moreover, there may be no global linear correlation can be found on the whole dataset.

3 Significant Local Linear Correlation Searching

3.1 Significant Local Linear Correlation Searching

Firstly, we give a brief overview of how to adopt PCA method to find the global linear correlation. Here, the "global" means the correlation is discovered from the whole dataset. Let $D = A \times B$ be a data matrix containing M tuples, each tuple has d features, where A is the set of features and B is the set of tuples. C_D is the covariance matrix of D. Let $\{\lambda_i\}$ $(1 \le i \le d)$ be the eigenvalues of C_D, where $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_N$. In order to get the linear correlations by PCA, we only need to find the smallest eigenvalues and their corresponding eigenvectors. Let a = $[a_1, a_2, \ldots, a_d]^T$ be the eigenvector of the smallest eigenvalue. The hyperplane $a_1x_1 + a_2x_2 + \ldots + a_dx_d = c$ established is exactly a global linear correlation, where c is a constant.

Example 2. Figure 2 shows a hyperplane of a linear correlation: $x_1 + 0.33x_2 - x_3 = 0$. The normal vector of this hyperplane is $[1, 0.33, -1]^T$, which is the eigenvector of the smallest eigenvalue (variance).

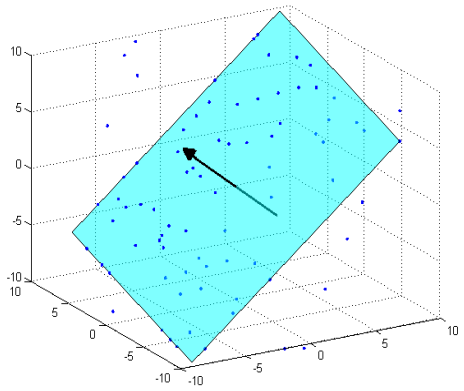


Fig. 2. An example of a hyperplane and its normal vector

We use the definition of strongly correlated feature subset proposed in [1] to measures both accuracy and significance of the local correlation.

Definition 1 (Significant Local Linear Correlation). Given a data matrix D containing M tuples, each tuple has d features. Let $S = \{x_{j1}, \ldots, x_{jm}\}$ $(0 \le j_t \le M, t = 1, 2, \ldots, m)$ is the subset of D, $\{\lambda_i\}(1 \le i \le d)$ be the eigenvalues of the covariance matrix of S, where $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_d$. The data subset S is in a *significant local linear correlation* if following conditions are true:

$$f(S,k) = \sum_{t=1}^{k} \lambda_t / \sum_{t=1}^{d} \lambda_t \le \varepsilon \qquad (1)$$

$$m / M \geq \delta \quad (2)$$

where ε, δ and k ($k \leq d$) are user defined parameters. The meanings of these user defined parameters are as same as in [1].

The accuracy is measured by the objective function $f(S, k)$ because the each eigenvalue λ_i indicates the variance on corresponding eigenvector's direction. Geometrically, $f(S, k)$ determines the "thickness" of the represented hyperplane. On the other hand, the significance is measured by parameter δ which determines the minimal size of the data subset. More tuples covered by the local linear correlation, more significant the correlation would be.

3.2 The Design of SLICE

In this section, we present our algorithm to find significant local linear correlations in dataset. We call our algorithm as SLICE (**s**ignificant **l**ocal l**i**near **c**orrelation s**e**arching).

The computation cost would be very large if each subset is enumerated and tested to make sure it is in a significant local linear correlation or not. Given a data matrix with M tuples, there are total 2^M data subsets to be tested, which is impractical for real-world applications. Due to this reason, we develop an approximate approach SLICE to find the significant linear correlations. SLICE cannot guarantee to find all existed significant linear correlations every time, but it is capable to reach them in a high probability within an acceptable computational cost in real datasets. Our extensive experimental results show that SLICE can discover more than 95% all linear correlations.

3.2.1 Heuristic Searching Method

Our SLICE adopts a heuristic to find the data subset within a significant local linear correlation. Suppose the dataset has M tuples and d features, the heuristic searching begins with an initial seed (d tuples), and let S be the set of the d tuples, then S absorbs the next "best" tuple in rest tuples iteratively until the value of $f(S, k)$ becomes larger than ε, which is introduced in Definition 1. If $|S|$ satisfies the requirement in Definition 1, the linear correlation established on S can be regarded as a *significant local linear correlation*. The concept of the "best" of a tuple will be discussed later in this section. We introduce the definition of distance from a tuple to a hyperplane firstly.

Definition 2 (Distance from a tuple to a hyperplane). Given a d dimensional data subset S and a tuple x. The distance from x to the hyperplane established on S, denoted as $d(x, S)$, is defined as follows.

$$d(x, S) = f(S \cup \{x\}, k) - f(S, k) \quad (3)$$

where $f(S, k)$ is defined in Definition 1.

According to Definition 1 and Definition 2, we can see that if the distance from a tuple to the hyperplane is small, the increase of $f(S, k)$ would be small after S absorbs this tuple. So, in the process of searching, the tuple with the minimal distance to the hyperplane is the "best" tuple to be absorbed. As a result, the accuracy of the hyperplane would be higher.

Our approach to find the "best" tuple is efficient and incremental. We maintain the mean and covariance matrix of S in whole searching to improve the efficiency of calculating the value of $f(S, k)$.

Lemma 1 (Incremental calculation for covariance and mean). Given a set of tuples $S = \{x_{j1}, ..., x_{jm}\}$, let $n = |S|$, x_0 be the mean of S and C_S be the covariance matrix of S. When S absorbs a new tuple x, the new mean x_0' and new covariance matrix C_S' can be obtained by following equations.

$$x_0' = \frac{n}{n+1} x_0 + \frac{1}{n+1} x \tag{4}$$

$$C_S' = \frac{n}{n+1}(C_S + x_0 x_0^T) + \frac{1}{n+1} x x^T - x_0' x_0'^T \tag{5}$$

By Lemma 1, the computational complexity of updating the covariance matrix of S in each iteration is $O(d^2)$ (d is the dimensionality). Considering the computational complexity of PCA, the time complexity of finding the minimum distance between a tuple and the hyperplane is $O(d^2+d^3)$. As the dimensionality d is a constant, the distance calculation can be finished in a constant time consuming. Thus, finding the "best" tuple in each step costs $O(n(d^2+d^3))$ at most.

3.2.2 Finding All Significant Local Linear Correlations

The second key point of SLICE is how to arrange the initial seeds to start the searching to find all significant local linear correlations. If the number of features is d and the size of dataset is M, there are C_M^d different initial seeds to be tested. It is easy to see that the computational cost is large when the dimensionality is large.

Fortunately, we can use a pruning strategy to set the initial seeds for searching. Then the whole process can converge very fast. The pruning strategy is based on following practical assumptions:

- A significant local liner correlation covers a large portion of tuples. That is, parameter δ should be large. The reason lies that correlations with few tuples may be caused by noise or coincidence of data distribution. Those correlations are meaningless to user.
- The overlaps of significant local linear correlations are all relatively small, which means parameter ε should be small. Parameter ε controls the "thickness" of hyperplane indicated by the significant local linear correlation in geometric view. As showed in Figure 3, if the hyperplanes are "thin", the intersection of portions is small. Intuitively, if the overlap of two significant local linear correlations is large, these two correlations should be very similar.

By these two assumptions, SLICE does not choose tuples as initial seeds which have been absorbed by a hyperplane. Although it cannot guarantee to find all significant local linear correlations, our experimental study shows that the probability for SLICE to reach all of them is larger than 95% (see Experimental Study Section).

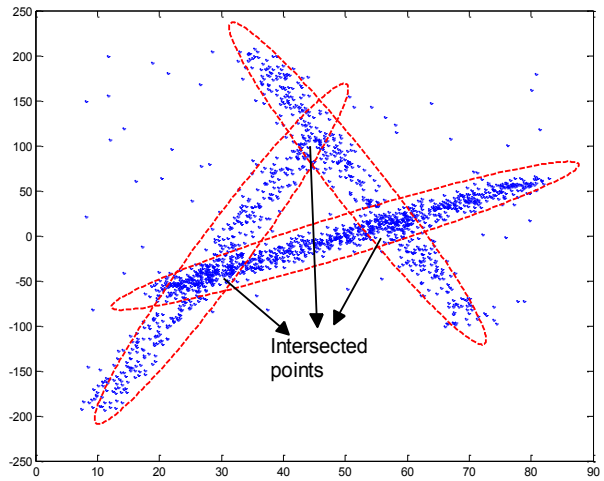


Fig. 3. Three hyperplanes with three overlapped parts

Algorithm 1 describes the implementation details of SLICE.

Algorithm 1 SLICE (D, ε, δ, k)

Input:
D: a d-dimensional data set D, ε, δ, k: user defined parameters.
Output:
SS: data subsets that are in a *significant local linear correlation*.
Begin

```
1    M ← |D|, C ← D
2    SS ← ∅
3    while |C| > d
4        seed ← ∅
5        while |seed| < d
6            p ← RANDOM(C, d)                    // select a tuple from C
7            seed ← seed ∪ {p}
8        end
9        S ← SEARCH(D, seed, ε, δ, k)    // call heuristic searching method
10       if |S| ≥ M·δ  then
11           SS ← SS ∪ {S}
12       end
13       C ← (C – S)
14   end
```

End.

Note: In Step 1, we maintain a set C of candidate initial seeds that to be invoked in heuristic searching. Subroutine RANDOM(C, d) in Step 6 is to select d tuples from the set C randomly. Subroutine SEARCH(D, *seed*, ε, δ, k) in Step 9 is to invoke the heuristic searching method discussed in the Subsection 3.2.1.

The time complexity of Algorithm 1 is $O(t \cdot n(d^2+d^3))$, where t is the times of invoking searching method. In the worst case, $t = n/d$. If ε is small, the number of tuples got in Subroutine SEARCH(D, *seed*, ε, δ, k) is small. In this case, t would be large. So we can see that t is associated with parameter ε. Our experimental results show that t is less than n in most cases.

4 Experimental Study

To evaluate the performance of SLICE, we test it on several synthetic datasets and a real world dataset. SLICE is implemented using Matlab 7.6. The experiments are performed on a 1.8GHz PC with 2G memory running Windows Server 2003 operating system.

The characteristics of experimental datasets are presented as follows.

- **Synthetic datasets**: we generate 500 distinct datasets. These synthetic datasets are categorized to 5 different groups. Each group consists of 100 different synthetic datasets, and has a distinct label, such as "D300F4C3". The label indicates the characteristics of each dataset in this group. For example, "D300F4C3" means each dataset in this group has 300 tuples with 4 features, and contains 3 predefined local linear correlations. All feature values of each tuple are generated in [-200, 200] randomly. Gaussian noise with mean 0 and variance 1.0 is added into each dataset.
- **Real world dataset:** we use NBA statistics dataset[1] to test the performance of SLICE. We use all 458 players' statistical scores in season 2006. The features in this dataset include *minutes*, *assists*, *rebounds* and so on, which are usually used for basketball specialists to evaluate basketball players in NBA. The meaning of each feature can be found at the website where we download the data.

4.1 Effectiveness Evaluation

In order to evaluate the results of algorithms, we conduct SLICE on these 500 synthetic datasets. We compare the discovered correlations with predefined correlations in datasets. If SLICE finds all predefined correlations, we mark this run as a *success*. We record the number of *success* times over all 500 synthetic datasets. We set k equals to 1. The values of ε, δ are list in Table 1. We choose these values

[1] http://sports.espn.go.com/nba/statistics

according to the dimensionality and predefined correlations of dataset. That is, different dataset has different these two parameter values. Table 2 shows the *success* rate[2] of SLICE on the 5 groups of synthetic datasets. From Table 2, we can conclude that our SLICE has high probability to reach all linear correlations in these synthetic datasets.

Table 1. Values of ε , δ for synthetic datasets

D300F3C3	D400F3C4	D600F3C4	D600F4C4	D500F4C5
ε =0.0006	ε =0.0006	ε =0.0006	ε =0.0001	ε =0.0001
δ =0.3	δ =0.2	δ =0.2	δ =0.2	δ =0.18

Table 2. *success* rates of SLICE for each group of datasets

D300F3C3	D400F3C4	D600F3C4	D600F4C4	D500F4C5
100%	95%	99%	97%	100%

4.1.1 Comparative evaluation on synthetic datasets

Next, we compare SLICE with 4C and CARE, since 4C and CARE are mostly recent works close to ours to the best of our knowledge.

a) Comparison with algorithm 4C: 4C is a kind of correlation clustering algorithm [10]. This algorithm, likes DBSCAN, clusters instances into $\varepsilon-$neighborhoods at first. In the synthetic datasets, each correlation intersects with another correlation. Therefore, for this kind of datasets, the generated $\varepsilon-$neighborhoods contain tuples in different correlations that would mislead searching the correct correlation clusters. In the experiment, we vary the parameters of 4C to get correlation clusters as large as possible. Figure 4 illustrates the best result got by 4C in dataset D300F3C3, where the parameters ε =300.0, μ =6, λ =2, δ =0.1 and κ =50. As shown in Figure 4, 3 significant local linear correlations found by SLICE are represented by 3 hyperplanes. They match the predefined correlations. However, only one correlation cluster is found by 4C on a part of one hyperplane. From the experimental results on dataset D300F3C3, we can see that no $\varepsilon-$neighborhood is created when the parameter ε is small. On the other hand, if parameter ε is large, the $\varepsilon-$neighborhoods would contain several hyperplanes' tuples, and no correlation cluster can be found. Due to the space limit, we only give the results in dataset D300F3C3 here (Figure 4). Similar results are found in other datasets.

b) Comparison with algorithm CARE: CARE is another type of method which is designed to find local linear correlations with both feature subset and tuple subset [1]. In this experiment, we use CARE to find linear correlations in tuple subset. Figure 5 illustrates the hyperplane found by CARE in dataset D300F3C3. CARE only finds one of three predefined correlations. Based on this result, we can see the main limitation of CARE is that it is only capable to find the primary linear correlation of

[2] We define the *success* rate is the percent of SLICE finds all predefined correlations over 500 datasets.

the dataset. Due to the space limit, we only give the results in dataset D300F3C3 here (Figure 5). Similar results are found in other datasets.

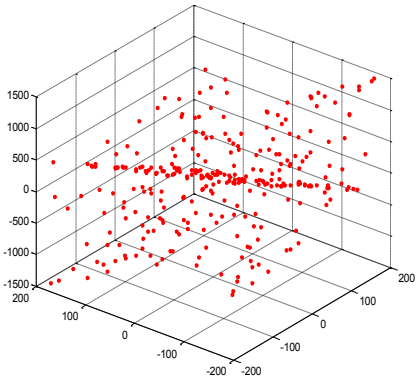


Fig. 4(a) D300F3C3 dataset.

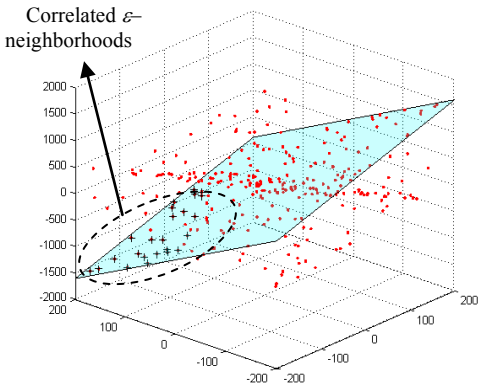


Fig. 4(b) Correlation found by 4C in D300F3C3 dataset

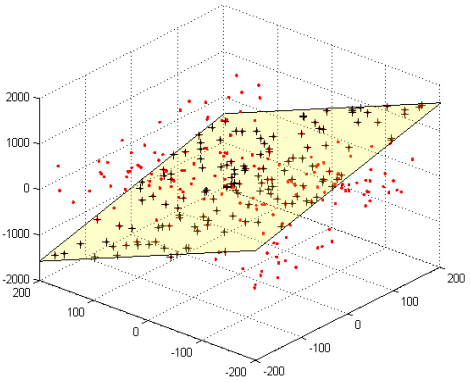


Fig. 4(c) Correlation found by CARE in D300F3C3 dataset

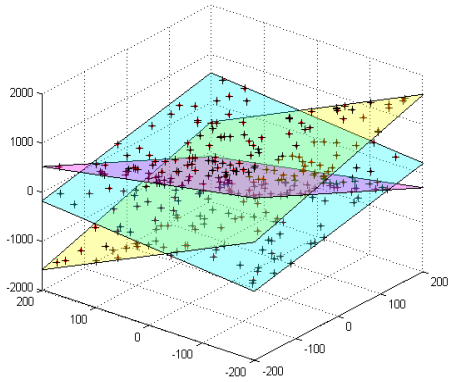


Fig. 4(d) Correlations found by SLICE in D300F3C3 dataset

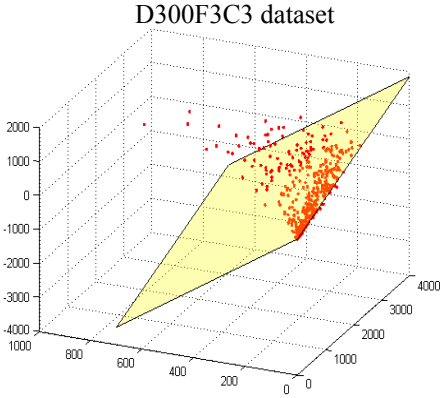


Fig. 5(a) Correlations found by CARE in NBA dataset.

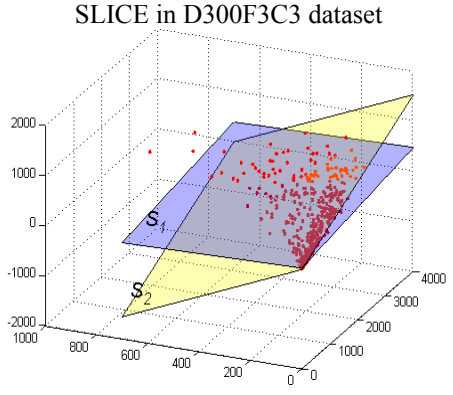


Fig. 5(b) Correlations found by SLICE in NBA dataset

4.1.2 Comparative evaluation on a real-world dataset

As the authors of [1] had demonstrated that CARE can better results than 4C, we just compare our SLICE with CARE in NBA dataset. In our experiments, we use SLICE and CARE to discover the linear correlations among *minutes*, *assists*, *rebounds* attributes. We use the same parameters in CARE and SLICE ($\varepsilon = 0.0006$, $\delta = 0.5$, $k = 1$). Table 3 lists the linear correlations discovered by CARE and SLICE.

Table 3. *success* rates of SLICE for each group of datasets

CARE	0.090645 * *minutes* – 0.976344 * *assists* – 0.196303 * *rebounds* = 0.796832
SLICE	0.157510 * *minutes* – 0.898574 * *assists* – 0.409580 * *rebounds* = 8.273140 0.112557 * *minutes* – 0.121851 * *assists* – 0.986146 * *rebounds* = -8.695694

From Table 3, we can see that SLICE finds two different linear correlations, while CARE just finds one linear correlation. In common sense, we know that there are different roles of players in a basketball team. Different players have different responsibilities in a match. For example, the player on Guard position is mainly in charge with assistance. So, we believe the results of SLICE are much closer to the real world compared the result of CARE.

4.2 Efficiency evaluation

We generate 10 datasets with different data sizes to evaluate the scalability of SLICE on data size. Moreover, we generate 10 datasets with different dimensionalities to evaluate the scalability of SLICE on feature size. All these synthetic datasets has 4 predefined local linear correlations. The parameter $k = 1$, and $\varepsilon = 0.0001$, $\delta = 0.2$.

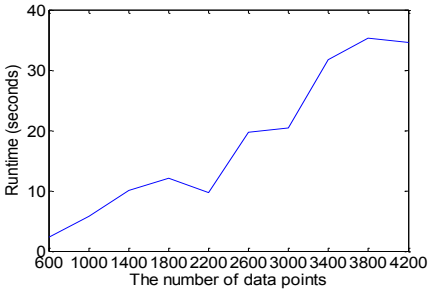


Fig. 6 Varying data size.

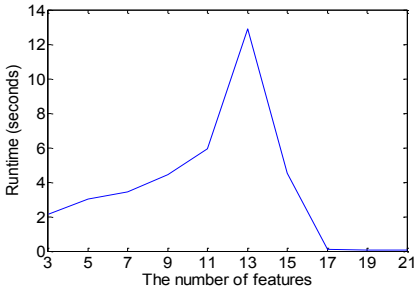


Fig. 7 Varying feature size

From Figure 6 we can see that the runtime of SLICE is increased linearly when the number of tuples is increased from 500 to 4200. From Figure 7, we can see that the runtime of SLICE is increased when the number of features is increased from 3 to 13. It is interesting to see that the runtime is decreased sharply when the number of features is larger than 13. We analyze the reason lies that the value of ε remains the same, so the local linear correlations may contain more tuples when the number of features is large than 13. Correspondingly, the times of invoking searching in SLICE

is decreased. Thus, we conclude that the value of ε should be smaller when there are more features in datasets.

5 Conclusions

Finding linear correlations in dataset has many real world applications. In this paper, we propose a method to find local linear correlations in data subsets. The main contributions of this paper include: (1) analyzing the limitations of applying current methods on finding linear correlations in data subsets; (2) developing a novel algorithm to find multiple local linear correlations in data subsets. The basic idea is using a heuristic to construct hyperplanes that represent linear correlations; (3) conducting extensive experiments to show that our algorithm is effective to find correct correlations in both synthetic and real-world datasets.

In future, we plan to integrate user interests in our method to find interesting local linear correlations. Furthermore, developing a method that can guarantee to find all correct linear correlations in polynomial time complexity in each execution is a challenging work.

References

1. Zhang, X., Pan, F., Wang, W.: CARE: Finding Local Linear Correlations in High Dimensional Data. In: the 24th IEEE International Conference on Data Engineering (ICDE), pp. 130-139 (2008)
2. Aggarwal, C., Yu, P.,: Finding Generalized Projected Clusters in High Dimensional Spaces. In: ACM SIGMOD 2000, pp. 70-81 (2000).
3. Aggarwal, C., Wolf, J., Yu, P.,: Fast Algorithms for Projected Clustering. In: ACM SIGMOD 1999, pp. 61-72 (1999)
4. Jolliffe, I.: Principal Component Analysis. Sprinter, New York (1986)
5. Lindman, H., R..: Analysis of Variance in Complex Experimental Designs. Wiley-Interscience (2001)
6. Kufunaga, K.: Introduction to statistical pattern recognition. Academic Press, San Diego, California (1990)
7. Mendenhall, W., Sincich, T.: A Second Course in Statistics: Regression Analysis. Prentice Hall (2002)
8. Yu, S., Yu, K., Kriegel H.-P., Wu, M.: Supervised Probabilistic Principal Component Analysis. In: ACM KDD 2006, pp. 464-473 (2006)
9. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In: ACM SIGMOD 1998, pp. 94-105 (1998)
10. Bohm, C., Kailing, K., Kroger, P., Zimek, A.: Computing Clusters of Correlation Connected Objects. In: ACM SIGMOD 2004, pp. 455-466 (2004)
11. Achtert, E., Bohm, C., Kriegel, H.-P., Kroger, P., Zimek, A.: Deriving Quantitative Models for Correlation Clusters. In: ACM KDD 2006, pp. 4-13 (2006)
12. Papadimitriou, S., Sun, J., Faloutsos, C.: Streaming Pattern Discovery in Multiple Time-Series. In: VLDB 2005, pp. 497-708 (2005)
13. Chakrabarti, K., Mehrotra, S.: Local Dimensionality Reduction: A New Approach to Indexing High Dimensional Spaces. In: VLDB 2000, pp. 89-100 (2000).