Big Data Research ••• (••••) •••–•••

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

A Near-Linear Time Subspace Search Scheme for Unsupervised Selection of Correlated Features ☆

Hoang-Vu Nguyen [a,*], Emmanuel Müller [a,b], Klemens Böhm [a]

[a] Karlsruhe Institute of Technology (KIT), Germany
[b] University of Antwerp, Belgium

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Correlation
Unsupervised feature selection
Subspace search
Outlier mining
Clustering
Classification

ABSTRACT

In many real-world applications, data is collected in high dimensional spaces. However, not all dimensions are relevant for data analysis. Instead, interesting knowledge is hidden in correlated subsets of dimensions (i.e., subspaces of the original space). Detecting these correlated subspaces independently of the underlying mining task is an open research problem. It is challenging due to the exponential search space. Existing methods have tried to tackle this by utilizing Apriori search schemes. However, their worst case complexity is exponential in the number of dimensions; and even in practice they show poor scalability while missing high quality subspaces.

This paper features a *scalable subspace search scheme (4S)*, which overcomes the efficiency problem by departing from the traditional levelwise search. We propose a new generalized notion of correlated subspaces which gives way to transforming the search space to a correlation graph of dimensions. We perform a direct mining of correlated subspaces in this graph, and then, merge subspaces based on the MDL principle in order to obtain high dimensional subspaces with minimal redundancy. We theoretically show that our search scheme is more general than existing search schemes. Our empirical results reveal that 4S in practice scales near-linearly with both database size and dimensionality, and produces higher quality subspaces than state-of-the-art methods.

1. Introduction

The notion of correlation is one of the key elements of statistics and is important for many areas of applied science. For instance, correlations have just recently been exploited in entropy-based dependency analysis to identify novel structures in global health and human gut microbiota data [38]. In data mining, correlations of dimensions often indicate the existence of patterns (e.g., for clustering [9], outlier mining [20], or as supervised feature selection for classification [16]), and are thus very important for knowledge discovery. In high dimensional data, however, patterns are often obscured in the full-space due to the presence of noisy features [7]. Instead, patterns can be found in (possibly overlapping) correlated subsets of dimensions, so-called correlated subspaces of the original data space. Detecting such subspaces, commonly referred to as subspace search, is crucial to unravel interesting knowledge and to understand high dimensional data.

Example 1. The facility management of our university stores indicator values of buildings, such as electricity, heating, gas, and water consumption per time unit. Each dimension is one indicator of a specific building. In such data, not all indicators of all buildings are correlated with each other. Instead, there are different subsets of correlated indicators, e.g., the heating indicators of office buildings, the ones of the Chemistry department, and so on. Overlap among subsets is possible since buildings can both be office buildings and belong to the Chemistry department. In practice, detecting subsets of correlated indicators is important for facility managers. This is because they can understand the energy consumption of the university better from such subsets. For instance, they can apply specialized data analytics on just those subsets to find anomalous measurements. An example would be an abnormally high heating value among the office buildings. Clearly, one cannot observe such patterns when indicators are uncorrelated or data is distributed randomly. Besides, it is preferable to find subsets with as many correlated indicators as possible, i.e., high dimensional subspaces. Returning redundant lower dimensional

☆ This article belongs to Scalable Computing for Big Data.

* Corresponding author.
E-mail addresses: hoang.nguyen@kit.edu (H.-V. Nguyen), emmanuel.mueller@kit.edu (E. Müller), klemens.boehm@kit.edu (K. Böhm).

http://dx.doi.org/10.1016/j.bdr.2014.07.004
2214-5796/

projections of the same subspace distracts users and misses the bigger picture.

Challenges

We observe three open challenges that have to be tackled for scalable subspace search, in particular w.r.t. dimensionality of data. First, it is unclear how to decide which subspaces have high correlations. Existing methods [9,19,20,33], using an Apriori-style search scheme, impose a restrictive monotonicity on the correlation model: A relevant subspace has to be relevant in all of its lower dimensional projections. However, this is only for efficiency reasons and may cause poor results in terms of quality. Second, one needs a scalable scheme to navigate through the huge search space. For a data set with 40 dimensions, the total number of subspaces is 2^{40} (more than 1 trillion). Looking at databases in practice (e.g., our facility management stores 540 dimensions), the search space is astronomically large. Obviously, this makes brute-force search impractical. Apriori-style methods, though employing the monotonicity restriction, still suffer from poor efficiency in practice due to (a) their expensive mining of correlated dimension pairs and (b) their level-by-level processing that generates a tremendous number of candidates. Third, the final set of subspaces should be free of redundancy, i.e., it should contain high dimensional subspaces rather than their low dimensional fragments. However, existing methods using the monotonicity restriction, detect fragmented subspaces of low quality which are redundant projections of the same high dimensional subspace.

Contributions

We address these challenges by proposing a *scalable subspace search scheme (4S)*. In general, we depart from the traditional Apriori search scheme and its monotonicity restriction. In particular, we make scalable subspace search feasible by creating a new generalized notion of correlated subspaces: We define a subspace to have a high correlation if its member dimensions are *all pairwise correlated*. We later establish a relationship between our notion and the well-known total correlation [9] and prove that our notion is more general than existing ones. That is, given the same correlation measure, all subspaces found by Apriori-style methods are also discovered by 4S. As a result, we expect 4S to discover not only subspaces found by such methods, but also interesting subspaces missed by them.

4S starts exploring the search space by computing pairwise correlations of dimensions. To ensure scalability, we devise two new efficient computation methods for this task. The first method takes advantage of two upper bounds of correlation scores to early prune non-candidate pairs, and hence, reduce computational cost. The second method uses AMS Sketch [6] to derive unbiased estimators of correlation scores, which in turn can be computed efficiently. To our knowledge, we are first to apply the theory of AMS Sketch for efficient computation of pairwise correlations of continuous random variables. Based on the pairwise correlations computed, we map the subspace search problem to efficient mining of maximal cliques in a correlation graph (with theoretical justifications). Hence, we get rid of the levelwise search of Apriori-style methods and directly construct higher dimensional subspaces by maximal clique mining. Due to this non-levelwise processing, 4S neither requires to compute correlations of each subspace candidate nor to check an excessive number of its lower dimensional projections. To address the fragmentation issue of high dimensional correlated subspaces, we transform the problem to an MDL-based merge scheme of subspaces and merge the detected subspaces accordingly. While MDL is an established notion for model selection, its deployment to subspace search is new.

Overall, our contributions include:

- A generalized notion of correlated subspaces, relaxing restrictions of traditional models.
- A scalable subspace search scheme, including efficient pairwise correlation computation and direct construction of high dimensional correlated subspaces.
- An MDL-based merge of subspaces to reconstruct fragmented subspaces and remove redundancy.

Paper organization

The road map of this paper is as follows. In Section 2, we introduce the main notions used. In Section 3, we discuss the properties of correlation measures and propose our own measure. In Section 4, we formally review the Apriori search scheme. In Section 5, we point out the practical requirements for subspace search in the era of big data research. In Section 6, we give an overview of 4S, with details on mining pairwise correlations in Section 7, mining higher dimensional subspaces in Section 8, and subspace merge in Section 9. We study the speed up of 4S in Section 10, followed by our extensive experiments in Section 11. We discuss related work in Section 12 and conclude the paper in Section 13.

Please note, that a preliminary version of this paper was published in [31]. We build upon that work and make the following new contributions: We include a formal discussion of correlation measures, and introduce our own measure in Section 3. We discuss requirements of subspace search for the era of big data research in Section 5, and derive our search scheme out of these requirements. We provide proofs on the correctness of our pruning rules in Section 7.1, a proof on the NP-hardness of the subspace search problem in Section 6, and introduce a formal formulation of our subspace merge in Section 9. Further, we include more experiments and discuss the experiment setup in more details. Results on a new real-world data set provide more insight into the practical impact of our work (cf., Section 11.3). More detailed analysis on the subspace merge and the reduction of redundancy is now provided in Section 11.4.

2. Preliminaries

Consider a database **DB** of size N and dimensionality D. The set of dimensions is denoted as the full-space $F = \{X_1, \ldots, X_D\}$. Each dimension X_i has a continuous domain $dom(X_i)$ and w.l.o.g., we assume $dom(X_i) = [-v, v] \subseteq \mathbb{R}$ with $v \geq 0$. We write $p(X_i)$ for the probability density function (pdf) of X_i. We also write $p(x_i)$ as a short form for $p(X_i = x_i)$. We let $P(X_i)$ stand for the cumulative distribution function (cdf) of X_i, and write $P(x_i)$ as a short form for $P(X_i \leq x_i)$.

A subspace S is a non-empty subset of F. Its dimensionality is written as $|S|$. The subspace lattice of **DB** consists of $D-1$ layers $\{\mathcal{L}_i\}_{i=2}^{D}$. Single dimensional subspaces are excluded since one is interested in correlations of two or more dimensions. Every layer $\mathcal{L}_i$ contains $\binom{D}{i}$ subspaces, each having i dimensions.

We aim at mining subspaces across all lattice layers whose member dimensions are highly correlated. Note that the search space is huge. For a dataspace with D dimensions the total number of possible subspaces is $O(2^D)$. For one subspace, one needs $O(D \cdot N)$ time to process, e.g., to compute the correlation. An overall complexity of $O(D \cdot N \cdot 2^D)$ makes brute-force search impractical. Even more sophisticated search schemes have severe scalability problems (see Section 4). Hence, we will propose a new scalable solution (see Section 6).

Please cite this article in press as: H.-V. Nguyen et al., A Near-Linear Time Subspace Search Scheme for Unsupervised Selection of Correlated Features, Big Data Research (2014), http://dx.doi.org/10.1016/j.bdr.2014.07.004

3. Correlation measure

To mine correlated subspaces, we need a *multivariate* correlation measure *Corr* for subspace assessment. Consider a d-dimensional subspace S. Without loss of generality, we assume that $S = \{X_1, \ldots, X_d\}$. In principle, the correlation score of S, denoted as both $Corr(S)$ and $Corr(X_1, \ldots, X_d)$, quantifies to which extent its joint probability function differs from the product of its marginal probability functions. The larger the difference, the higher $Corr(S)$ is. Hence, if all dimensions of S are statistically independent, $Corr(S) = 0$. Formally, we expect that

$$Corr(S) = Corr(X_1, \ldots, X_d) \sim diff\left(p(X_1, \ldots, X_d), \prod_{i=1}^{d} p(X_i)\right) \tag{1}$$

with *diff* being an instantiation of a difference function.

Our goal is to have a correlation measure that captures both linear and non-linear correlation. The measure should also permit direct calculation on empirical data without having to estimate probability density functions, or rely on discretization as in [9,38]. To this end, there are several options in the literature. Arguably, one of the most popular multivariate correlation measures is total correlation [10]. It is based on Shannon (differential) entropy and is widely used in many fields [24,40,44]. The formal definition of total correlation is as follows.

Definition 1. The total correlation of $\{X_1, \ldots, X_d\}$ is

$$T(X_1, \ldots, X_d) = \sum_{i=2}^{d} H(X_i) - H(X_i|X_1, \ldots, X_{i-1})$$

where $H(X_i)$ is the Shannon (differential) entropy of X_i, and $H(X_i|X_1, \ldots, X_{i-1})$ is the conditional entropy of X_i given $X_1, \ldots, X_{i-1}$.

Total correlation detects both linear and non-linear correlation. However, for continuous data it requires the pdfs, which in general are not available at hand and needs estimation [10]. Thus, total correlation in general is not directly computable on empirical data.

To address the issue and to achieve our goal, we propose a new correlation measure which is based on cdfs, is non-parametric (no prior assumption on the data distribution is required), and permits computation on empirical data in closed form. It is defined as follows.

Definition 2. The correlation score of $S = \{X_1, \ldots, X_d\}$ is

$$Corr(X_1, \ldots, X_d) = \int_{-v}^{v} \ldots \int_{-v}^{v} \big(P(x_1, \ldots, x_d) - P(x_1) \cdots P(x_d)\big)^2 dx_1 \cdots dx_d.$$

The lemma below immediately follows.

Lemma 1. *$Corr(X_1, \ldots, X_d) \geq 0$ with equality iff $p(X_1, \ldots, X_d) = p(X_1) \cdots p(X_d)$.*

According to Lemma 1, one can see that our correlation measure meets the expected property of a correlation measure laid out in Eq. (1). Further, we prove that it can be computed in closed form on empirical data. Let $\{X_i(1), \ldots, X_i(N)\}$ be realizations of X_i. We have:

Theorem 1.

$$Corr(X_1, \ldots, X_d) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \prod_{k=1}^{d} \big(v - \max\big(X_k(i), X_k(j)\big)\big) - \frac{2}{N^{d+1}} \sum_{i=1}^{N} \prod_{k=1}^{d} \sum_{j=1}^{N} \big(v - \max\big(X_k(i), X_k(j)\big)\big) + \frac{1}{N^{2d}} \prod_{k=1}^{d} \sum_{i=1}^{N} \sum_{j=1}^{N} \big(v - \max\big(X_k(i), X_k(j)\big)\big).$$

Proof. Our proof is based on [42]. In particular, let $ind(\alpha)$ be an indicator function with value 1 if α is true and 0 otherwise. It holds that

$$P(a_1, \ldots, a_d) = \int_{-v}^{v} \ldots \int_{-v}^{v} ind(x_1 \leq a_1) \cdots ind(x_d \leq a_d) \times p(x_1, \ldots, x_d) dx_1 \cdots dx_d. \tag{2}$$

Using empirical data, Eq. (2) becomes:

$$P(a_1, \ldots, a_d) = \frac{1}{N} \sum_{i=1}^{N} \prod_{k=1}^{d} ind\big(X_k(i) \leq a_k\big).$$

Likewise: $P(a_k) = \frac{1}{N} \sum_{i=1}^{N} ind(X_k(i) \leq a_k)$. Therefore, $Corr(X_1, \ldots, X_d)$ equals to:

$$\int_{-v}^{v} \ldots \int_{-v}^{v} \left(\frac{1}{N} \sum_{i=1}^{N} \prod_{k=1}^{d} ind\big(X_k(i) \leq x_k\big) - \prod_{k=1}^{d} \frac{1}{N} \sum_{i=1}^{N} ind\big(X_k(i) \leq x_k\big) \right)^2 dx_1 \cdots dx_d. \tag{3}$$

Expanding Eq. (3), we have:

$$\int_{-v}^{v} \ldots \int_{-v}^{v} \left(\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \prod_{k=1}^{d} ind\big(\max\big(X_k(i), X_k(j)\big) \leq x_k\big) - \frac{2}{N^{d+1}} \sum_{i=1}^{N} \prod_{k=1}^{d} \sum_{j=1}^{N} ind\big(\max\big(X_k(i), X_k(j)\big) \leq x_k\big) + \frac{1}{N^{2d}} \prod_{k=1}^{d} \sum_{i=1}^{N} \sum_{j=1}^{N} ind\big(\max\big(X_k(i), X_k(j)\big) \leq x_k\big) \right) dx_1 \cdots dx_d$$

Bringing the integrals inside the sums, we obtain:

$$\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \prod_{k=1}^{d} \int_{-v}^{v} ind\big(\max\big(X_k(i), X_k(j)\big) \leq x_k\big) dx_k - \frac{2}{N^{d+1}} \sum_{i=1}^{N} \prod_{k=1}^{d} \sum_{j=1}^{N} \int_{-v}^{v} ind\big(\max\big(X_k(i), X_k(j)\big) \leq x_k\big) dx_k + \frac{1}{N^{2d}} \prod_{k=1}^{d} \sum_{i=1}^{N} \sum_{j=1}^{N} \int_{-v}^{v} ind\big(\max\big(X_k(i), X_k(j)\big) \leq x_k\big) dx_k$$

by which we arrive at the final result. □

Please cite this article in press as: H.-V. Nguyen et al., A Near-Linear Time Subspace Search Scheme for Unsupervised Selection of Correlated Features, Big Data Research (2014), http://dx.doi.org/10.1016/j.bdr.2014.07.004

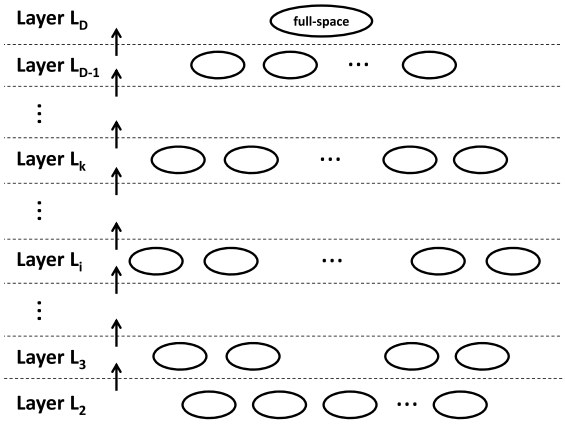


Fig. 1. Example showing the subspace lattice exploration of Apriori approach APR.

Algorithm 1: APR.

1 $CAND_2$ = Set of $min(MAX\_NUM, \binom{D}{2})$ subspaces in $\mathcal{L}_2$ with the highest correlation;
2 $OUT = CAND_2$;
3 $i = 2$;
4 **while** $CAND_i \neq \emptyset$ **do**
5 $\quad CAND_{i+1} = \{S \in \mathcal{L}_{i+1} : \forall S' \subset S \wedge |S'| = i \Rightarrow S' \in CAND_i\}$;
6 $\quad CAND_{i+1}$ = Top $min(MAX\_NUM, |CAND_{i+1}|)$ subspaces of $CAND_{i+1}$ with the highest correlation;
7 $\quad OUT = OUT \cup CAND_{i+1}$;
8 $\quad i = i + 1$;
9 Return OUT;

Using Theorem 1, computing our measure *Corr* on empirical data is straightforward. Thus, we will use *Corr* as the correlation measure in 4S. In fact, one can also plug *Corr* into existing subspace search schemes. However, in the next section we point out why these search schemes are not suited to big data applications.

4. Existing search schemes

Existing methods explore the search space based on the Apriori principle (APR) using a correlation measure for subspace assessment, e.g., total correlation as used in [9].

For APR, one can either keep a top number of subspaces at each layer (beam-based) or impose a threshold on the subspace correlation (threshold-based). Recently, [20,33] point out that the beam-based scheme allows more intuitive parameterization than the threshold-based one. Thus, for better presentation, we stick to the former. However, our discussion is also applicable to the threshold-based scheme [9,19].

We illustrate the lattice exploration of APR in Fig. 1. Its pseudocode is in Algorithm 1. APR starts at layer $\mathcal{L}_2$ (Line 1). For each layer $\mathcal{L}_i$ visited, APR computes the total correlation $T(S)$ for each candidate subspace $S \in \mathcal{L}_i$. The top $min(MAX\_NUM, \binom{D}{i})$ subspaces $CAND_i$ with the highest total correlation are selected (Lines 1 and 6). *MAX_NUM* is the beam size. $CAND_i$ is also used to determine which subspaces to examine in the next layer $\mathcal{L}_{i+1}$. In particular, a subspace S^{i+1} in $\mathcal{L}_{i+1}$ is considered iff all of its i-dimensional projections are in $CAND_i$ (Line 5). This is known as the monotonicity restriction, which causes redundant processing: To reach one subspace, one needs to generate and examine all of its lower dimensional projections, even though not all of them are relevant.

APR stops when either there is no more layer to explore, or the set of candidate subspaces in the current layer is empty. Assume that *MAX_NUM* is set such that APR reaches layer $\mathcal{L}_k$. We have:

Lemma 2. *The time complexity of APR is* $O\left(\Delta \cdot \sum_{i=2}^{k} \binom{D}{i}\right)$ *where* Δ *is the cost of computing the correlation of each subspace.*

Proof. For each layer $\mathcal{L}_i$ $(i \geq 2)$ with $\binom{D}{i}$ subspaces, the worst case time complexity to compute the correlation for all of its subspaces is $O\left(\Delta \cdot \binom{D}{i}\right)$. Thus, the overall time complexity is $O\left(\Delta \cdot \sum_{i=2}^{k} \binom{D}{i}\right)$. □

Regarding Δ, we have $\Delta = \Theta(N)$ with total correlation [9], and $\Delta = \Theta(N^2)$ with our *Corr* measure.

Since the monotonicity property imposes strict restrictions on high-level layers (i.e., high k), APR tends not to reach high dimensional subspaces. To resolve the issue, *MAX_NUM* must be very large. However, this causes APR to process many candidate subspaces at each layer visited. Further, to process a subspace, APR requires to examine exponentially many lower dimensional projections to ensure that they all have high correlation. These cause its runtime to become very high. Even when *MAX_NUM* is kept low, APR still suffers from poor scalability due to its expensive mining of $\mathcal{L}_2$, in particular, $O(D^2 \cdot \Delta)$. Further, setting *MAX_NUM* to low values fails to offset the monotonicity restriction. This prevents APR from discovering high dimensional subspaces. Only lower dimensional fragments of correlated subspaces are detected. Thus, the quality of subspaces is impacted.

Another drawback of using APR is that the higher the layer visited, the more likely it is that the curse of dimensionality occurs. This is because most existing multivariate correlation measures, including our *Corr* measure, suffer from reduction of discriminative power in high dimensional spaces—a phenomenon which has been demonstrated empirically in [33].

In summary, APR(a) is inefficient, (b) tends to miss high dimensional correlated subspaces, (c) fragments them into many redundant lower dimensional subspaces, and (d) is prone to the curse of dimensionality.

5. Subspace search for the era of big data research

Detecting interesting relationships among dimensions (i.e., correlated subspaces) in high dimensional spaces lies at the heart of big data analytics. It is important for humans to understand their data, e.g. in the domains of genomics, physics, political science, economics, etc. [38]. However, in practice this is challenging because more and more dimensions are being added to scientific and industrial databases in order to capture the increasing complexity of information. In such cases it is naturally better to present a succinct set of correlated subspaces than letting users face with the daunting task of exploring the data by themselves. Furthermore, detecting correlated subspaces also helps to steer the focus of users to particular views of data where they will likely discover interesting patterns.

In general, to address the subspace search problem, one has to handle a huge search space, which is exponential in the number of dimensions. This is also the reason why we believe that detecting correlated subspaces is a subject of big data research: The 'big' aspect of data is not only in its physical size, but also in the virtual size of any search space contingent on its structure. In our case, methods have to tackle the exponential search space of all subspaces.

Before getting to any specific solution, it is important to specify the desired properties of a good subspace search scheme. First, to explore a search space of large volume, one typically expects many of its parts to be pruned out to achieve scalability. In other words, it is crucial that the search algorithm is able to avoid unnecessary processing of non-promising parts of the search space. This is the point where APR-based methods do not deliver the expected scalability: They have to process very many non-candidate subspaces

Please cite this article in press as: H.-V. Nguyen et al., A Near-Linear Time Subspace Search Scheme for Unsupervised Selection of Correlated Features, Big Data Research (2014), http://dx.doi.org/10.1016/j.bdr.2014.07.004

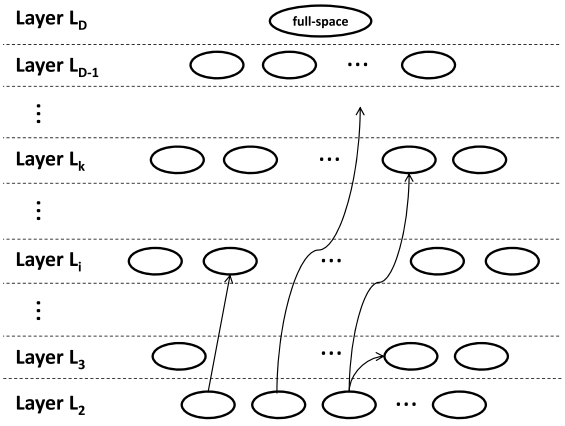


Fig. 2. Example showing the subspace lattice exploration of 4S.

before reaching the relevant ones. As a consequence, they do not deliver good results on data sets with medium amounts of dimensions and records (see Section 11). Second, to avoid 'sequential' processing, a search scheme should have the capability of quickly identifying promising subspaces. By quickly, we imply a 'jump' processing as in [27,28], i.e., the ability to jump to higher dimensional subspaces by exploiting the statistics collected from subspaces of much lower dimensionality. To meet the efficiency requirement, the method to combine low dimensional subspaces (e.g., two dimensional ones) must, of course, be efficient. Lastly, the scheme for combining subspaces should be reliable, or in other words, the promising subspaces it detects should be verifiable to be the actual correlated subspaces.

With respect to the requirements mentioned above, there is not yet a satisfactory solution for scalable subspace search. The work here in turn is our first step towards addressing this shortcoming. In particular, we deploy a search scheme whose main goal is to efficiently reach the promising subspaces, without spending time for irrelevant ones. Further, by means of theoretical analysis using information theory, we verify its reliability, and hence, its quality.

Next, we summarize our solution to fulfill these requirements and provide more details in the subsequent sections.

6. Overview of 4S processing

We illustrate the lattice exploration of 4S in Fig. 2 and contrast it to the APR scheme depicted in Fig. 1. To avoid the exponential runtime in the data dimensionality, 4S does not explore the subspace lattice in a levelwise manner. Instead, 4S initially mines subspaces of high correlations in $\mathcal{L}_2$. They are then combined to directly create higher dimensional subspaces. In short, 4S works in three steps. First, we compute the correlation of each pair of dimensions and only keep the top K pairs (i.e., subspaces of $\mathcal{L}_2$) with the largest correlations. Setting K is explained in Section 8.

Second, we construct an undirected correlation graph $\mathcal{G}_\mathcal{D}$ representing our search space of subspaces. Its nodes are the dimensions, connected by an edge iff their correlation is in the top K values. Following our new notion of correlated subspaces, we mine maximal cliques of this correlation graph. They serve as candidate subspaces. We also prove that these candidate subspaces are likely mutually correlated. The toy example in Fig. 3 displays a correlation graph for a 10-dimensional data set. There are 45 possible subspaces in $\mathcal{L}_2$; $K = 10$ of which are picked to construct $\mathcal{G}_\mathcal{D}$. From $\mathcal{G}_\mathcal{D}$, 4S finds three maximal cliques (subspaces): $S_1 = \{1, 2, 3, 4\}$, $S_2 = \{1, 3, 4, 5\}$, and $S_3 = \{7, 8\}$.

Third, mining maximal cliques on $\mathcal{G}_\mathcal{D}$ may also produce subspaces that are projections of the same subspaces due to the restriction on pairwise correlations (i.e., through K). For instance, in

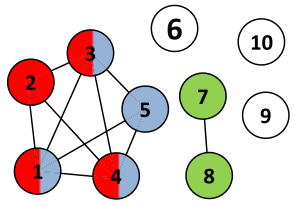


Fig. 3. Example of correlation graph.

Fig. 3, dimension 5 is connected to all dimensions in S_1 except for dimension 2. This leads to the detection of two separate subspace fragments S_1 and S_2 that have high overlap with each other. It would make sense to merge S_1 and S_2 to create the larger subspace $\{1, 2, 3, 4, 5\}$. This also helps us to cope with real-world data where perfect pairwise correlation between dimensions of correlated subspaces may not always be fulfilled. Thus, we propose to merge similar subspaces using an MDL-based approach. Following this step, we obtain higher dimensional subspaces with minimal redundancy.

Overall, in contrast to APR, we can reach high dimensional correlated subspaces with our scalable search scheme, which consists of: (a) scalable computation of $\mathcal{L}_2$, (b) scalable mining of $\mathcal{L}_k$ with $k > 2$, and (c) subspace merge. While APR needs to impose the Apriori monotonicity restriction on all layers for the sake of efficiency, we only require that dimensions of subspaces are pairwise correlated (i.e., restriction on $\mathcal{L}_2$).

There is one remark that we want to highlight regarding our 4S processing scheme. That is, the subspace search under our problem transformation (i.e., after the computation of pairwise correlations in $\mathcal{L}_2$) is NP-hard. We note that the input of the subspace search problem for database **DB** consists of: (a) the set of all dimensions $\mathcal{F}$, and (b) the set of dimension pairs $\mathcal{P} \subseteq \mathcal{F} \times \mathcal{F}$ with the largest correlations to be kept. We prove the NP-hardness of the subspace search problem by giving a polynomial reduction of the classic Maximal Cliques Mining (MCM) problem, which is a NP-hard problem [18], to subspace search (SS): MCM $\leq_p$ SS.

Theorem 2. *The subspace search problem under the setting of* 4S *is NP-hard.*

Proof. First, we map the input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of MCM to an input $(\mathcal{F}, \mathcal{P})$ of SS. The mapping is straightforward: For each node $t \in \mathcal{V}$, we add a dimension X_t to the set of dimensions $\mathcal{F}$. After that, for each edge $e = (t, t') \in \mathcal{E}$, we add the pair $(X_t, X_{t'})$ into $\mathcal{P}$. Obviously, our mapping is done in polynomial time. Now we have to show that the subspace search result (under our notion of correlated subspaces) on $(\mathcal{F}, \mathcal{P})$ corresponds to a solution of MCM in $\mathcal{G}$. This is easy to show as by our convention, each correlated subspace $S \subseteq \mathcal{F}$ has to satisfy:

(a) $\forall (X_t, X_{t'}) \subset S \Rightarrow (X_t, X_{t'}) \in \mathcal{P}$;
(b) no proper superset of S is a correlated subspace, i.e., S is maximal.

In other words, S represents a maximal clique of $\mathcal{G}$. Thus, solving SS for the constructed instance $(\mathcal{F}, \mathcal{P})$ leads to a valid solution of MCM for $\mathcal{G}$. □

Following Theorem 2, without making any assumption on the structure of the search space, i.e., the correlation graph $\mathcal{G}_\mathcal{D}$, SS is NP-hard. To overcome the issue, we instead impose restrictions on $\mathcal{G}_\mathcal{D}$. In particular, we heuristically make $\mathcal{G}_\mathcal{D}$ sparse by implicitly limiting the maximal degree of its nodes. We accomplish this by carefully setting K (more details are in Section 8). By enforcing

Please cite this article in press as: H.-V. Nguyen et al., A Near-Linear Time Subspace Search Scheme for Unsupervised Selection of Correlated Features, Big Data Research (2014), http://dx.doi.org/10.1016/j.bdr.2014.07.004

$\mathcal{G}_{\mathcal{D}}$ to be sparse, we are able to apply exact and efficient algorithms [18], which have good performance on sparse graphs. Next, we introduce the details of 4S (see Sections 7–10), including our analysis showing that 4S reliably identifies correlated subspaces and is more general than APR in Section 8. We empirically show that 4S produces subspaces of higher quality than existing methods in Section 11.

7. Scalable computation of $\mathcal{L}_2$

In $\mathcal{L}_2$, we need to compute the correlation score of all pairs of dimensions. To this end, for two dimensions X and Y, we have:

Lemma 3.

$$\begin{aligned}
&Corr(X,Y)\\
&= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\big(v-\max(x_i,x_j)\big)\big(v-\max(y_i,y_j)\big)\\
&\quad - \frac{2}{N^3}\sum_{i=1}^{N}\left(\sum_{j=1}^{N}\big(v-\max(x_i,x_j)\big)\right)\left(\sum_{j=1}^{N}\big(v-\max(y_i,y_j)\big)\right)\\
&\quad + \frac{1}{N^4}\sum_{i=1}^{N}\sum_{j=1}^{N}\big(v-\max(x_i,x_j)\big)\sum_{i=1}^{N}\sum_{j=1}^{N}\big(v-\max(y_i,y_j)\big).
\end{aligned}$$

Proof. The proof is obtained by simply applying Theorem 1 for $d=2$. □

Following Lemma 3, we need to compute three terms, referred to as T_1, T_2, and T_3, and

$$Corr(X,Y) = \frac{1}{N^2}T_1 - \frac{2}{N^3}T_2 + \frac{1}{N^4}T_3.$$

To compute $Corr(X,Y)$, we need $O(N^2)$ time. For D-dimensional data sets, the total runtime required to explore layer $\mathcal{L}_2$ becomes $O(D^2N^2)$. This is a serious problem for any data set. To tackle the issue, we introduce two new approaches, *MultiPruning* and *Sketching*, to boost efficiency regarding both N and D. *MultiPruning* calculates the exact correlation. However, it still has issues regarding efficiency for large data sets. *Sketching* in turn trades accuracy for efficiency. Yet it is still better than APR (see Section 11). Note that *Corr* deploys the same estimator as other quadratic measures of independence [36], such as [1,43]. The difference only lies in different kernels employed. Thus, the ideas of *MultiPruning* and *Sketching* are also applicable to other measures of the same category. In other words, our method is not limited to one correlation measure.

7.1. MultiPruning

MultiPruning aims at reducing the runtime by applying pruning rules for $Corr(X,Y)$ based on two upper bounds of T_1. It uses the fact that we only keep the top K pairs of dimensions with the largest correlation. Let $\{(x_{s(i)}, y_{s(i)})\}_{i=1}^{N}$ be $\{(x_i,y_i)\}_{i=1}^{N}$ sorted in descending order w.r.t. X. The upper bounds of T_1 are as follows.

Theorem 3 (Cauchy–Schwarz bound).

$$T_1 \le \sum_{i=1}^{N}\sqrt{\sum_{j=1}^{N}\big(v-\max(x_i,x_j)\big)^2\sum_{j=1}^{N}\big(v-\max(y_i,y_j)\big)^2}.$$

Proof. Applying the Cauchy–Schwarz inequality, we have that for each $i\in[1,N]$:

$$\begin{aligned}
&\left(\sum_{j=1}^{N}\big(v-\max(x_i,x_j)\big)\big(v-\max(y_i,y_j)\big)\right)^2\\
&\le \sum_{j=1}^{N}\big(v-\max(x_i,x_j)\big)^2\sum_{j=1}^{N}\big(v-\max(y_i,y_j)\big)^2. \qquad (4)
\end{aligned}$$

Taking the square root for each side of Eq. (4) and summing up over all $i\in[1,N]$, we arrive at the final result. □

Lemma 4. *It holds that*

$$\begin{aligned}
T_1 &= \sum_{i=1}^{N}(v-x_{s(i)})(v-y_{s(i)})\\
&\quad + 2\sum_{i=1}^{N}(v-x_{s(i)})\sum_{j=i+1}^{N}\big(v-\max(y_{s(i)},y_{s(j)})\big).
\end{aligned}$$

Proof. We have:

$$\begin{aligned}
T_1 &= \sum_{i=1}^{N}\sum_{j=1}^{N}\big(v-\max(x_i,x_j)\big)\big(v-\max(y_i,y_j)\big)\\
&= \sum_{i=1}^{N}\sum_{j=1}^{N}\big(v-\max(x_{s(i)},x_{s(j)})\big)\big(v-\max(y_{s(i)},y_{s(j)})\big)\\
&= \sum_{i=1}^{N}(v-x_{s(i)})(v-y_{s(i)})\\
&\quad + 2\sum_{i=1}^{N}\sum_{j=i+1}^{N}\big(v-\max(x_{s(i)},x_{s(j)})\big)\big(v-\max(y_{s(i)},y_{s(j)})\big)\\
&= \sum_{i=1}^{N}(v-x_{s(i)})(v-y_{s(i)})\\
&\quad + 2\sum_{i=1}^{N}(v-x_{s(i)})\sum_{j=i+1}^{N}\big(v-\max(y_{s(i)},y_{s(j)})\big). \qquad \square
\end{aligned}$$

Theorem 4 (Sorted-based bound).

$$T_1 \le \sum_{i=1}^{N}(v-x_{s(i)})(v-y_{s(i)}) + 2\sum_{i=1}^{N}(v-x_{s(i)})\sum_{j=i+1}^{N}(v-y_{s(j)}).$$

Proof. The proof is derived directly from Lemma 4 and the fact that $v-\max(y_{s(i)},y_{s(j)}) \le v-y_{s(j)}$. □

The statistics required for the Cauchy–Schwarz bound, for instance $\sum_{j=1}^{N}(v-\max(x_i,x_j))^2$ for $1\le i\le N$, can be pre-computed for each dimension in $O(N\log N)$ time. This is from our observation that

$$\begin{aligned}
&\sum_{j=1}^{N}\big(v-\max(x_i,x_j)\big)^2\\
&= \sum_{x_j\ge x_i}(v-x_j)^2 + \sum_{x_j<x_i}(v-x_i)^2\\
&= \sum_{x_j\ge x_i}(v-x_j)^2 + (v-x_i)^2\cdot\big|\{x_j : x_j<x_i\}\big|.
\end{aligned}$$

Please cite this article in press as: H.-V. Nguyen et al., A Near-Linear Time Subspace Search Scheme for Unsupervised Selection of Correlated Features, Big Data Research (2014), http://dx.doi.org/10.1016/j.bdr.2014.07.004

Algorithm 2: Computing the statistics of the Cauchy–Schwarz bound on X.

```
1 {(x_{s(1)}, org_pos_1), ..., (x_{s(N)}, org_pos_N)} ← Sort {x_1, ..., x_N} in descending order;
2 sum = 0;
3 for i = 1 → N do
4     ret(org_pos_i) = sum + (v − x_{s(i)})^2 · (N − i + 1);
5     sum = sum + (v − x_{s(i)})^2;
6 Return {ret(1), ..., ret(N)};
```

That is, for each dimension, we first sort its data in descending order. Then, we loop through the data *once* in that order and pre-compute the required statistics. We illustrate our point by giving in Algorithm 2 a sample pseudocode, which computes the statistics of the Cauchy–Schwarz bound on X. We note that org_pos_i stands for the original position (before sorting) of $x_{s(i)}$. A corresponding numerical example is described below.

Example 2. Consider three data points $P_1 = (1, -1)$, $P_2 = (-1, 1)$, and $P_3 = (0, 0)$ (i.e., $dom(X) = [-1, 1]$). To compute the statistics for X, we sort X in descending order and obtain $\{1, 0, -1\}$. Then, we compute $\sum_{j=1}^{3}(1 - \max(x_i, x_j))^2$ $(1 \le i \le 3)$ by looping through the sorted list once and obtain: 0 for P_1, 5 for P_2, and 2 for P_3. Similarly, for $\sum_{j=1}^{3}(1 - \max(y_i, y_j))^2$ $(1 \le i \le 3)$, we obtain: 5 for P_1, 0 for P_2, and 2 for P_3. We calculate the Cauchy–Schwarz bound by looping through the stored statistics once, and achieve: $\sqrt{0 \cdot 5} + \sqrt{5 \cdot 0} + \sqrt{2 \cdot 2} = 2$.

The statistics required to *exactly* compute the second term T_2 of $Corr(X, Y)$, which is $\sum_{j=1}^{N}(v - \max(x_i, x_j))$ for $1 \le i \le N$, can be pre-computed similarly. The statistics of the third term T_3, which is $\sum_{i=1}^{N}\sum_{j=1}^{N}(v - \max(x_i, x_j))$, is also computed during this phase by incrementally summing up the statistics of T_2 (per dimension).

During the pairwise correlation computation, we maintain the top K values seen so far. For a new pair of dimensions (X, Y), we first compute the bounds. This computation is in $O(N)$ (see Algorithm 2). Similarly, the exact value of the second term T_2 is computed. Similarly to Algorithm 2, we obtain the sorted-based bound in Theorem 4 in $O(N)$ time. The details are as follows. We loop through the data sorted w.r.t. X. For each point $(x_{s(i)}, y_{s(i)})$, we compute $(v - x_{s(i)})(v - y_{s(i)})$ and $(v - x_{s(i)})\sum_{j=i+1}^{N}(v - y_{s(j)})$. We do so by taking into account that

$$\sum_{j=i+1}^{N}(v - y_{s(j)}) = \sum_{j=1}^{N}(v - y_{s(j)}) - \sum_{j=1}^{i}(v - y_{s(j)}).$$

The sorted-based bound can also be computed w.r.t. Y. So in fact, we have two versions of this bound, one for X and one for Y. The exact value of T_3 is computed in just $O(1)$ time using its pre-computed statistics, which are $\sum_{i=1}^{N}\sum_{j=1}^{N}(v - \max(x_i, x_j))$ and $\sum_{i=1}^{N}\sum_{j=1}^{N}(v - \max(y_i, y_j))$.

If any upper bound of $Corr(X, Y)$ is less than the Kth largest value so far, we can safely stop computing its actual value. Otherwise, we compute T_1, and hence $Corr(X, Y)$ (and update the top K correlation values), using Lemma 4. That is, for each $x_{s(i)}$, we search for $y_{s(i)}$ in the list of values of Y sorted in descending order. For each value $y > y_{s(i)}$ encountered, we add $2(v - x_{s(i)})(v - y)$ to T_1. Once $y_{s(i)}$ is found, the search stops. Suppose that the position found is p, and the list has e elements. We add $2(e - p + 1)(v - x_{s(i)})(v - y_{s(i)})$ to T_1. We remove $y_{s(i)}$ from the list and proceed to $x_{s(i+1)}$. This helps us to avoid scanning the whole list and, hence, reduces the runtime. We note that $\sum_{i=1}^{N}(v - x_{s(i)})(v - y_{s(i)})$ is already computed during the sorted-based bound computation.

By means of pruning rules and efficient computation heuristics, *MultiPruning* is able to achieve efficiency in practice. However, as there is no theoretical guarantee on the effectiveness of the pruning rules, the worst-case complexity of *MultiPruning* is still $O(D^2N^2)$. This motivates us to introduce *Sketching*, which trades accuracy for further improvement in scalability.

7.2. Sketching

To better address the scalability issue (i.e., quadratic in N), we propose *Sketching* as an alternative solution. First, we see that T_3 is computed in only $O(1)$ time using its pre-computed statistics. Thus, our main intuition is to convert the terms T_1 and T_2 to forms similar to that of T_3. We observe that T_1 and T_2 can be perceived as dot products of vectors. In particular, T_1 is the product of vectors

$$\big(v - \max(x_1, x_1), \ldots, v - \max(x_1, x_N), \ldots, v - \max(x_N, x_1), \ldots, v - \max(x_N, x_N)\big)$$

and

$$\big(v - \max(y_1, y_1), \ldots, v - \max(y_1, y_N), \ldots, v - \max(y_N, y_1), \ldots, v - \max(y_N, y_N)\big).$$

Likewise, T_2 is the product of vectors

$$\left(\sum_{j=1}^{N}\big(v - \max(x_1, x_j)\big), \ldots, \sum_{j=1}^{N}\big(v - \max(x_N, x_j)\big)\right)$$

and

$$\left(\sum_{j=1}^{N}\big(v - \max(y_1, y_j)\big), \ldots, \sum_{j=1}^{N}\big(v - \max(y_N, y_j)\big)\right).$$

Vector products in turn can be efficiently estimated by AMS Sketch [6]. AMS Sketch provides rigorous theoretical bounds for its estimation and can outperform other sketching schemes [39]. However, to our knowledge, we are first to use this theory to efficiently compute pairwise correlations of continuous random variables.

Our general idea is to use AMS Sketch to derive unbiased estimators of T_1 and T_2 that have forms similar to T_3. The estimators are unbiased since their expected values equal to their respective true values. We will prove that the estimators are close to the true values of T_1 and T_2, respectively. Overall, *Sketching* reduces the time complexity of computing $Corr(X, Y)$ to $O(N \log N)$.

Sketching approximates $Corr(X, Y)$ through unbiased estimators by projecting X and Y onto random 4-wise independent vectors. Let $u, w \in \{\pm 1\}^N$ be two such vectors which are independent to each other. We estimate T_1 as follows:

Theorem 5. *Let Z be a random variable that equals to*

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\big(v - \max(x_i, x_j)\big)u_i w_j \sum_{i=1}^{N}\sum_{j=1}^{N}\big(v - \max(y_i, y_j)\big)u_i w_j$$

then $E(Z) = T_1$ and $Var(Z) \le 8[E(Z)]^2$.

Likewise, we estimate T_2 as:

Theorem 6. *Let W be a random variable that equals to*

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\big(v - \max(x_i, x_j)\big)u_i \sum_{i=1}^{N}\sum_{j=1}^{N}\big(v - \max(y_i, y_j)\big)u_i$$

then $E(W) = T_2$ and $Var(W) \le 2[E(W)]^2$.

Please cite this article in press as: H.-V. Nguyen et al., A Near-Linear Time Subspace Search Scheme for Unsupervised Selection of Correlated Features, Big Data Research (2014), http://dx.doi.org/10.1016/j.bdr.2014.07.004

We derive Theorems 5 and 6 based on [6]. These theorems allow us to approximate T_1 and T_2 by estimators having forms similar to that of T_3. Hence, $Corr(X, Y)$ can be approximated in $O(N \log N)$ time by pre-computing the statistics required in a way similar to *MultiPruning*. Please note that, we also need to ensure estimators to concentrate closely enough around their respective mean. To accomplish this, we apply Chebychev's inequality. The variance of Z is upper-bounded by $8[E(Z)]^2$. By averaging over s_1 different values of u and w, the variance is reduced to at most $\frac{8[E(Z)]^2}{s_1}$. Using Chebychev's inequality, we have

$$P\big(|Z - E(Z)| > \varepsilon E(Z)\big) \leq \frac{8}{s_1 \varepsilon^2}.$$

If we repeat the averaging $s_2 = O(1/\delta)$ times and take the median of these averages, the relative error of Z w.r.t. $E(Z)$ is at most ε with probability at least $1 - \delta$, as proven in [6].

Similarly, by averaging over s_1 different values of u, the variance of W is reduced to at most $\frac{2[E(W)]^2}{s_1}$. Applying Chebychev's inequality, we have

$$P\big(|W - E(W)| > \varepsilon E(W)\big) \leq \frac{2}{s_1 \varepsilon^2}.$$

We again boost the estimation accuracy by repeating the averaging $s_2 = O(1/\delta)$ times.

Sorting all dimensions costs $O(DN \log N)$. For each random vector and each dimension, it costs the same amount of time as that of T_3 to pre-compute the statistics, which is $O(N)$ (see Section 7.1). For all vectors and all dimensions, the total cost of pre-computing statistics is $O(s_1 s_2 DN)$. Since $s_1 s_2$ must be large enough to guarantee estimation accuracy, the cost of pre-computing statistics dominates that of data sorting. For each pair of dimensions, the cost to calculate its (estimated) correlation is $O(s_1 s_2)$. Thus, computing the correlations for all dimension pairs and maintaining the top values cost $O(s_1 s_2 D^2 + D^2 \log K)$, with $O(s_1 s_2 D^2)$ dominating. Therefore, the total time complexity of *Sketching* is $O(s_1 s_2 DN + s_1 s_2 D^2)$. In our experiments, $D < N$, i.e., the time complexity becomes $O(s_1 s_2 DN)$, a considerable improvement from $O(D^2 N^2)$. We note that the factor $s_1 s_2$ does not contribute much to the overall runtime, and in practice *Sketching* scales linearly in both N and D.

8. Scalable mining of $\mathcal{L}_k$

Based on the set of 2-dimensional subspaces found in $\mathcal{L}_2$, denoted as $\mathcal{S}_2$, we now explain how to mine subspaces in higher-level layers. According to our notion, a subspace has a high correlation if its member dimensions are all pairwise correlated. We now point out that subspaces fulfilling our notion likely have a high total correlation. We also formally prove that our new notion of correlated subspaces is more general than that of APR. That is, given the same correlation measure, all subspaces found by APR are also discovered by our mining scheme. Further, we will demonstrate empirically later on that, with our notion, 4S produces better subspaces than APR. First, let us consider a subspace S with all pairs $\{X_i, X_j\} \in \mathcal{S}_2$. W.l.o.g., assume that $S = \{X_1, \ldots, X_d\}$.

Lemma 5. *The total correlation is lower-bounded by*

$$T(X_1, \ldots, X_d) \geq \sum_{i=2}^{d} H(X_i) - H(X_i|X_{i-1}).$$

Proof. As conditioning reduces entropy [35], we have: $H(X_i|X_{i-1}) \geq H(X_i|X_1, \ldots, X_{i-1})$. Using Definition 1, we arrive at the result. □

By definition, every pair $\{X_{i-1}, X_i\} \in \mathcal{S}_2$ has a high correlation. Following Definition 2, this means that $P(X_{i-1}, X_i)$ and $P(X_{i-1})P(X_i)$ deviate from each other. Thus, the joint density function $p(X_{i-1}, X_i)$ of X_{i-1} and X_i deviates from the product of their marginal density functions, which is $p(X_{i-1})p(X_i)$ [42]. Consequently, $H(X_i) - H(X_i|X_{i-1})$, which equals to the Kullback–Leibler divergence of $p(X_{i-1}, X_i)$ and $p(X_{i-1})p(X_i)$, is high. Based on Lemma 5, we conclude that: $T(X_1, \ldots, X_d)$ is high. Lemma 5 also holds for any permutation of $X_1, \ldots, X_d$. Hence, under any permutation of the dimensions of S, S has a high total correlation. This also means: The difference between the joint density function of S and the product of its marginal density functions is high w.r.t. the Kullback–Leibler divergence. Hence, subspaces fulfilling our notion likely are mutually correlated, not just pairwise correlated. Since many other correlation measures define mutual correlation based on the difference between the joint distribution and the product of marginal distributions [36], our subspaces are also likely mutually correlated under such correlation measures.

We now prove that our new notion of correlated subspaces is more general than that of APR:

Theorem 7. *Let S be a subspace detected by APR using Corr as correlation measure and given MAX_NUM $\leq K$, then all of its pairs $\{X_i, X_j\} \in \mathcal{S}_2$.*

Proof. We use induction:
Let $S = \{X_1, \ldots, X_d\}$ be a subspace mined by APR.

Basis: When $d = 2$, since *MAX_NUM* $\leq K$, we have that $S \in \mathcal{S}_2$.

Hypothesis: Suppose that Theorem 7 holds for $d = n \geq 2$.

Inference: We prove that Theorem 7 also holds for $d = n + 1$, i.e., we prove $\forall X_i \neq X_j \in S$: $\{X_i, X_j\} \in \mathcal{S}_2$. This is straightforward. For $X_i \neq X_j$, there exists an n-dimensional subspace $U \subset S$ such that $X_i, X_j \in U$ and U is included by APR in the output (cf., monotonicity property). Hence, $\{X_i, X_j\} \in \mathcal{S}_2$ according to the hypothesis. □

Theorem 7 also holds for other correlation measures, e.g., the ones in [9,20,33], with $\mathcal{S}_2$ being formed according to the measure used. It implies that, given the same correlation measure and *MAX_NUM* $\leq K$, all subspaces included in the final output of APR are also discovered by our mining scheme. This is because any two of their dimensions are pairwise correlated, i.e., they form cliques in the correlation graph. This shows that our mining scheme is more general than APR and, hence, can discover subspaces missed by APR. Note that a subspace satisfying the pairwise condition is not necessarily included in the final output of APR. Also, the monotonicity restriction imposed by APR is only to reduce the runtime [33], and does not guarantee the quality of subspaces. Our empirical study also confirms this.

Having formally analyzed the theoretical properties of our notion of correlated subspaces, we now map the problem of mining subspaces in higher-level layers to maximal clique mining in the correlation graph. Consider an undirected correlation graph $\mathcal{G}_\mathcal{D}$ with nodes being the dimensions. An edge exists connecting two dimensions X_i and X_j iff $\{X_i, X_j\} \in \mathcal{S}_2$. A subspace of our interest then forms a clique in $\mathcal{G}_\mathcal{D}$. To avoid redundancy, we propose to mine only maximal cliques, i.e., subspaces are not completely contained in each other. We regard maximal cliques of $\mathcal{G}_\mathcal{D}$ as the result of this step.

Given D dimensions, the worst-case complexity to find all maximal cliques is $O(3^{D/3})$. To ensure the practicality of 4S, we rely on a recent finding [5]. It states that the properties of a data set (e.g., distances between data points) are preserved after dimensionality reduction as long as the number of dimensions kept is $O(\log N)$. As a result, we set $K \leq D \log N$, i.e., $O(D \log N)$. Hence, the expected maximal degree of each node in $\mathcal{G}_\mathcal{D}$ is $O(\log N)$, i.e., each dimension can be part of subspaces (maximal cliques) with

Please cite this article in press as: H.-V. Nguyen et al., A Near-Linear Time Subspace Search Scheme for Unsupervised Selection of Correlated Features, Big Data Research (2014), http://dx.doi.org/10.1016/j.bdr.2014.07.004

expected maximal dimensionality $O(\log N)$. This implies that the expected degeneracy of $\mathcal{G}_\mathcal{D}$ is $O(\log N)$. Following [13], we obtain the following result:

Theorem 8. *The expected time complexity of mining maximal cliques is* $O(DN^{1/3}\log N)$. *The expected number of maximal cliques is* $O((D - \log N)N^{1/3})$.

Therefore, using our strategy, we can efficiently and directly mine high dimensional subspaces with reduced knowledge loss. Further, we achieve this without traversing the subspace lattice in a levelwise manner. Note that our scheme is different from approaches imposing the maximal dimensionality of subspaces. This is because the maximal dimensionality is *implicitly* embedded in 4S (by setting K), rather than explicitly. Further, 4S is not constrained by the $O(\log N)$ bound in practice. This is due to our MDL-based merge of subspaces described next, which reconstructs high dimensional correlated subspaces from fragments.

9. Subspace merge

We denote the set of dimensions, each belonging to at least one maximal clique, as $\{X_{r(j)}\}_{j=1}^{l}$. Also, $\{C_i\}_{i=1}^{m}$ is the set of maximal cliques. Due to the pairwise restriction of our subspace notion, subspaces (maximal cliques) obtained by mining $\mathcal{G}_\mathcal{D}$ may be projections of the same higher-dimensional correlated subspaces (see Fig. 3). To reconstruct such subspaces and to remove redundancy in the output, we merge subspaces into groups such that the new set of subspaces *guarantees completeness and minimizes redundancy*. To accomplish this, we first construct a binary matrix $\mathcal{B}$ with l rows and m columns. The rows are dimensions, and the columns are cliques. $\mathcal{B}_{ij} = 1$ iff X_i is in C_j, and 0 otherwise.

Example 3. The binary data set $\mathcal{B}$ of the example in Fig. 3 is as follows:

		Columns		
		S_1	S_2	S_3
Rows	X_1	1	1	0
	X_2	1	0	0
	X_3	1	1	0
	X_4	1	1	0
	X_5	0	1	0
	X_7	0	0	1
	X_8	0	0	1

The columns of $\mathcal{B}$ correspond to the three subspaces detected. The cells are colored according to the color of its respective subspace.

We transform the subspace merge to grouping similar columns of $\mathcal{B}$, each final group constituting one subspace. We aim at achieving the task without having to define any distance function among the dimensions of $\mathcal{B}$. Thus, we apply the merge algorithm proposed in [25] which uses the Minimum Description Length (MDL) principle.

Given a set of models $\mathcal{M}$, MDL identifies the best model $M \in \mathcal{M}$ as the one that minimizes

$$L(\mathcal{B}, M) = L(M) + L(\mathcal{B} \mid M).$$

Here, $L(M)$ is the length in bits of the description of the model M, and $L(\mathcal{B} \mid M)$ is the length of the description of the data $\mathcal{B}$ encoded by M. That is, MDL helps select a model that yields the best balance between goodness of fit and model complexity.

In our problem, each model is a grouping of maximal cliques $\{C_i\}_{i=1}^{m}$. Each candidate grouping $G = \{A_1, \ldots, A_k\}$ under consideration is a partitioning of $\{C_i\}_{i=1}^{m}$, i.e., it must satisfy three properties: (a) $\bigcup_{i=1}^{k} A_i = \{C_i\}_{i=1}^{m}$, (b) for $i \neq j$: $A_i \cap A_j = \emptyset$, and (c) for every i: $A_i \neq \emptyset$.

Each $A_i \in G$ can be described by a *code table* CT_i. This table has an entry for each possible value a from $dom(A_i)$. The left-hand column of CT_i contains the value, and the right-hand column contains the corresponding code (the code assigned by MDL encoding). The frequency of $a \in CT_i$ is defined as its support relative to the number of rows l of the binary data set $\mathcal{B}$: $fr(A_i = a) = supp(A_i = a)/l$. The total encoding cost $L(\mathcal{B}, G)$ is given as:

Definition 3. Let a grouping $G = \{A_1, \ldots, A_k\}$ be given. We have: $L(\mathcal{B}, G) = L(G) + L(\mathcal{B} \mid G)$, where

- $L(\mathcal{B} \mid G) = l \sum_{i=1}^{k} H(A_i)$,
- $L(G) = \log B_n + \sum_{i=1}^{k} L(CT_i)$,
- $L(CT_i) = \sum_{a \in dom(A_i)} |A_i| + \log\log l - \log fr(A_i = a)$

with $fr(A_i = a) \neq 0$, and B_n being the Bell number.

Following Definition 3, we mine the grouping G that minimizes the total encoding cost. However, since the search space is $O(2^m)$ and unstructured, we utilize a heuristic algorithm. We start with each attribute forming its own group. Then, we progressively pick two groups whose merge leads to the largest reduction in the total encoding cost, and we merge them. This practice also ensures two most similar groups are merged at each step [25]. The algorithm terminates when either there are no more groups to merge, or when the current step does not reduce the total encoding cost any more. We have the following result:

Theorem 9. *The subspace merge guarantees completeness and minimizes redundancy.*

That is, our subspace merge ensures that its output subspaces contain all the subspaces produced by the second step (completeness). This stems from the fact that MDL guarantees a *lossless compression* [15]. Thus, the original set of subspaces is compressed while ensuring that no information loss occurs. Besides, our algorithm heuristically selects the grouping of subspaces that minimizes the overall compression cost. For instance, if a grouping contains two very similar subspaces (i.e., redundant ones), our algorithm would not pick it since the merge of two subspaces can result in a better grouping with a lower encoding cost. Hence, redundancy is minimized.

According to [25], the total time complexity of this step is $O(lm^3)$, which is $O((D - \log N)^3 lN)$. Nevertheless, the runtime in practice is much smaller because (a) the number of cliques is much smaller than the one stated in Theorem 8, (b) the number l of dimensions left is small compared to D, and (c) the subspace merge algorithm in [25] terminates early. Our experiments also point out that the runtime of this step is negligible compared to the first step. While APR can also apply this subspace merge, it does not achieve the same quality as 4S since it hardly ever reaches high dimensional subspaces.

10. Overall complexity analysis

Table 1 summarizes the time complexity reduction achieved by 4S for each step. The computation of $\mathcal{L}_2$ (using *Sketching*) costs $O(DN)$. The mining of $\mathcal{L}_k$ costs $O(DN^{1/3}\log N)$. The subspace merge costs $O((D - \log N)^3 lN)$. Thus, the worst-case complexity of 4S is $O((D - \log N)^3 lN)$. However, our experiments point out that the most time-consuming step is the computation of $\mathcal{L}_2$, which accounts for nearly 90% of the overall runtime. Hence, overall, we

Please cite this article in press as: H.-V. Nguyen et al., A Near-Linear Time Subspace Search Scheme for Unsupervised Selection of Correlated Features, Big Data Research (2014), http://dx.doi.org/10.1016/j.bdr.2014.07.004

Table 1
Overview of complexity reduction.

Step	Brute-force	4S
Computation of $\mathcal{L}_2$	$O(D^2N^2)$	$O(DN)$
Mining of $\mathcal{L}_k$	$O(3^{D/3})$	$O(DN^{1/3}\log N)$
Subspace merge	$O(2^{(D-\log N)N^{1/3}})$	$O((D-\log N)^3 lN)$
Overall complexity	$O(2^{(D-\log N)N^{1/3}})$	$O((D-\log N)^3 lN)$

for our part conclude that 4S has $O(DN)$ average-case complexity. Our experiments also confirm that 4S has near-linear scalability with both N and D.

Considering the high time complexity of APR where k is the highest layer reached (see Lemma 2), one can see that the speed-up 4S achieves is a significant improvement: It enables a near-linear time heuristic search to explore an exponential search space. Further, it eliminates the efficiency bottleneck of exploring $\mathcal{L}_k$. Nevertheless, we note that 4S does not push the envelop. That is, from the linear scalability of 4S, there is in fact still much room of improvement towards a truly scalable solution for big data. For instance, we could use parallelization: Parallelizing the computation in $\mathcal{L}_2$ is straightforward; parallelizing the search in $\mathcal{L}_k$ can be done by using special techniques, such as [41].

11. Experiments

We write 4S-M and 4S-S as 4S with *MultiPruning* and *Sketching*, respectively. We compare 4S with the following methods: FS as baseline in full-space; FB [23] using random subspaces for outlier mining; EC [9], CMI [33], and HICS [20] representing the APRstyle methods; FEM [12] representing the unsupervised feature selection approaches. For all of these methods, we try several parameter combinations in order to find the optimal parameter setting for each data set. Finally, we use the results of the parameter combination showing the best quality results. We note that we attempt to find the most relaxing parameters for EC, CMI, and HICS that can return good results within five days. As shown later, this practice, however, is not always possible due to the high complexity of APR methods.

For our methods, following Theorem 8, we set $K = D\log N$ to ensure a reasonable tradeoff between quality and efficiency. For 4S-S, we need to set s_1 and s_2. Regarding the former, we fix $s_1 = 10\,000$ as large values of s_1 yield high estimation accuracy [34]. For the latter, we fix $s_2 = 2$ following the observation that smaller values for s_2 generally result in better accuracy [11].

We test the quality of subspaces produced by all methods in: outlier detection with LOF [8], clustering with DBSCAN [14], and classification with the C4.5 decision tree. The first two areas are known to yield meaningful results when the subspaces selected have high correlations, i.e., include few or no irrelevant dimensions [9,20,23,29,33]. Hence, they are good choices for evaluating the quality of correlated subspaces. The third area is to show that correlated subspaces found by 4S are also useful for the supervised domain. For each method, LOF, DBSCAN, and C4.5 are applied on its detected subspaces and the results are combined (following [23] for LOF, [9] for DBSCAN, and [17] for C4.5), and judged using corresponding well-known performance metrics. Regarding parameter settings: We set *MinPts* of LOF to about $\min(N \cdot 0.005, 100)$. For DBSCAN, we set *MinPts* from 2 to 10, and ε from 0.01 to 0.04. For C4.5, we use the default parameter setting in WEKA.

For this quantitative assessment of subspaces, we use synthetic data and 6 real-world labeled data sets from the UCI Repository: the Gisette data about handwritten digits; HAR, PAMAP1, and PAMAP2 all sensor data sets with physical activity recordings; Mutant1 and Mutant2 containing biological data used for cancer prediction. Further, we use the facility management's database of our university (KIT) with energy indicators recorded from 2006 to 2011. More details are in Table 2. Note that each of them has more than 1 trillion subspaces. This features a challenging search space w.r.t. dimensionality for all methods.

Table 2
Characteristics of real-world data sets. Each of them has more than 1 trillion subspaces.

Data set	Size	Attributes	Classes
Gisette	13 500	5000	2
HAR	10 299	561	6
KIT	48 501	540	2
Mutant1	16 592	5408	2
Mutant2	31 159	5408	2
PAMAP1	1 686 000	42	15
PAMAP2	1 662 216	51	18

Besides, we also further our study on the performance of 4S by experimenting it with a real-world unlabeled data set on climate and energy consumption. Our objective here is to qualitatively assess the subspaces detected by 4S, e.g., if they make sense to our domain expert. Lastly, we investigate how well the subspace merge of 4S compresses the output subspaces, or in other words, how succinct the output of 4S is.

We assist future comparison, by providing data sets, parameters, and algorithms on our project website.[1]

11.1. Experiments on synthetic data

Quality on outlier detection. We have created 6 synthetic data sets of 10 000 records and 100 to 1000 dimensions. Each data set contains subspace clusters with dimensionality varying from 8 to 24 and we embed 20 outliers deviating from these clusters. Our performance metric is the Area Under the ROC Curve (AUC), as in [23,20,21]. From Table 3, one can see that 4S-M overall has the best AUC on all data sets. 4S-S in turn achieves the second-best performance. In fact, in most cases, 4S-M correctly discovers all embedded subspaces. Though 4S-S does not achieve that, its subspaces are close to the best ones, and it has better performance than other methods. We are better than FS, which focuses on the full-space where noisy dimensions likely hinder the detection of outliers. Our methods outperform FB, which highlights the utility of our correlated subspaces compared to random ones. Examining the subspaces found by APR-style methods (EC, CMI, and HICS), we see that they are either irrelevant, or they are low dimensional fragments of relevant subspaces. This explains their poor performance. FEM has low AUC since it only mines a single subspace and hence, misses other important correlated subspaces where outliers are present.

Quality on clustering. Synthetic data sets with 100 to 1000 dimensions are used again. Our performance metric is the F1 measure, as in [29,26]. Table 4 displays clustering results of all methods. One can see that 4S-M and 4S-S have the best performance on all data sets tested. This again highlights the quality of subspaces found by our methods.

From the outlier detection and clustering experiments, we can see that 4S-S is a good approximation of 4S-M.

APR using subspace merge. For illustration, we only present the outlier detection and clustering results on the synthetic data set with 10 000 records and 1000 dimensions. From Table 5, by applying the subspace merge, APR-style methods achieve better AUC and F1 values than without merge. Yet, our methods outperform all of them. This is because APR-style methods already face severe issue with reaching high dimensional subspaces. Thus, applying subspace merge in their case cannot bring much of improvement.

[1] http://www.ipd.kit.edu/~muellere/4S/.

Please cite this article in press as: H.-V. Nguyen et al., A Near-Linear Time Subspace Search Scheme for Unsupervised Selection of Correlated Features, Big Data Research (2014), http://dx.doi.org/10.1016/j.bdr.2014.07.004

Table 3
AUC on outlier mining for synthetic data sets. Highest values are in **bold**.

Data set	4S-M	4S-S	FS	FB	EC	CMI	HICS	FEM
D100	**1.00**	**1.00**	**1.00**	0.65	0.90	0.46	0.43	0.50
D200	**1.00**	**1.00**	0.99	0.50	0.85	0.47	0.46	0.48
D400	**0.99**	0.98	0.96	0.51	0.83	0.46	0.45	0.63
D600	**0.99**	0.98	0.77	0.54	0.76	0.42	0.29	0.54
D800	**0.99**	0.87	0.75	0.61	0.74	0.43	0.40	0.59
D1000	**0.99**	0.92	0.81	0.47	0.75	0.46	0.40	0.64

Table 4
F1 on clustering for synthetic data sets. Highest values are in **bold**.

Data set	4S-M	4S-S	FS	FB	EC	CMI	HICS	FEM
D100	**0.99**	**0.99**	0.72	0.95	0.67	0.50	0.80	0.76
D200	**0.89**	**0.89**	0.67	0.66	0.67	0.50	0.80	0.76
D400	**0.85**	0.83	0.67	0.81	0.67	0.80	0.77	0.75
D600	**0.96**	0.95	0.67	0.66	0.67	0.67	0.83	0.53
D800	**0.99**	0.93	0.67	0.67	0.67	0.67	0.83	0.74
D1000	**0.91**	0.88	0.67	0.67	0.83	0.67	0.74	0.75

Table 5
Comparison with APR using subspace merge on the synthetic data set with 10 000 records and 1000 dimensions. Highest values are in **bold**.

Task	4S-M	4S-S	EC	CMI	HICS
Outlier Mining (AUC)	**0.99**	0.92	0.76	0.49	0.44
Clustering (F1)	**0.91**	0.88	0.84	0.70	0.76

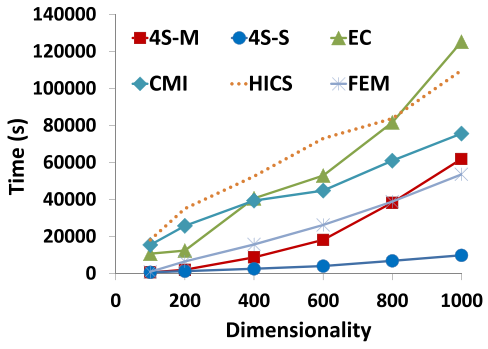


Fig. 4. Runtime vs. dimensionality on synthetic data.

Scalability. Since FS and FS do not spend time for finding subspaces, we only analyze the runtime of the remaining methods. To test scalability to dimensionality, we use data sets with 10 000 data points and dimensionality of 100 to 1000. Based on Fig. 4, we see that 4S-S has the best scalability. FEM scales better than 4S-M because it only searches for a single subspace. Overall, 4S-S has near-linear scalability to dimensionality, thanks to our efficient search scheme.

For scalability to data size, we use data sets with 100 dimensions and sizes of 10 000 to 100 000. From Fig. 5, we see that 4S-S scales linearly and is more efficient than 4S-M. This agrees with our theoretical analysis.

We also note that the runtime of the first step in our methods dominates the other two steps. For example, on the data set of 10 000 records and 1000 dimensions, 4S-S takes about 150 minutes for the first step and only 14 minutes for the remaining two steps. These costs in turn are negligible compared to the cost of performing clustering and outlier detection on the subspaces detected. For instance, LOF requires about 4 days to process this data set. From the results obtained, we can conclude that 4S-S achieves the efficiency goal while still ensuring high quality of subspaces found. From now onwards, we use 4S-S for the remaining experiments and write only 4S.

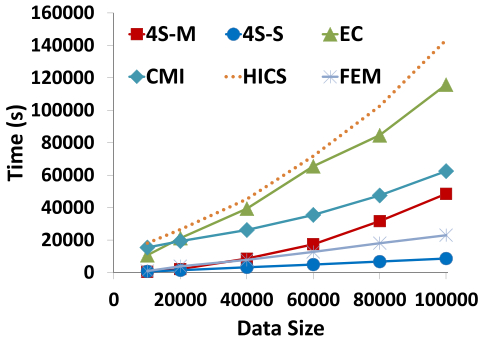


Fig. 5. Runtime vs. data size on synthetic data.

11.2. Experiments on real data

We apply all methods to two applications: outlier detection and classification. Clustering is skipped here since it conveys similar trends among the methods as with synthetic data.

Quality on outlier detection. As a standard procedure in outlier mining [23,20,21], the data sets used are converted to two-class ones, i.e., each contains only a class of normal objects and a class of outliers. This is done by either picking the smallest class or down-sampling one class to create the outlier class. The rest forms the normal class. From Table 6, 4S achieves the best results. Its superior performance compared to other methods, including APR-style methods techniques (EC, CMI, and HICS), stems from the fact that 4S better discovers correlated subspaces where outliers are visible. For example, on the KIT data set, 4S finds subspaces where several consumption indicators of different buildings of the same type (e.g., office buildings, laboratories) cluster very well with a few exceptions, possibly caused by errors in smart-meter readings, or rare events (e.g., university holidays when energy consumption is low or large-scale physics experiments when electricity consumption is extremely high). These subspaces however are not discovered by other methods.

On the PAMAP1 and PAMAP2 data sets, we can only compare 4S against FS, FB, and FEM. This is because other methods take excessively long time without completing. These data sets contain data collected by sensors attached to human bodies when they perform different activities, e.g., walking, running, ascending stairs. The best AUC of 4S on both data sets once again implies that 4S successfully discovers high quality subspaces, which in turn assist in the detection of outliers. For example, the subspaces found by 4S on PAMAP1 exhibit correlations among the hand, chest, and ankle of

Please cite this article in press as: H.-V. Nguyen et al., A Near-Linear Time Subspace Search Scheme for Unsupervised Selection of Correlated Features, Big Data Research (2014), http://dx.doi.org/10.1016/j.bdr.2014.07.004

Table 6
AUC on outlier mining for real-world data sets. Highest values are in **bold**. (*) means the result is unavailable due to excessive runtime.

Data set	4S	FS	FB	EC	CMI	HICS	FEM
Gisette	**0.77**	0.67	0.60	0.73	0.74	0.74	0.68
HAR	**0.67**	0.42	0.53	0.27	0.65	0.15	0.53
KIT	**0.73**	0.36	0.51	0.33	0.55	0.55	0.44
Mutant1	**0.62**	0.58	0.55	0.56	0.58	0.57	0.55
Mutant2	**0.64**	0.57	0.53	0.55	0.58	0.59	0.56
PAMAP1	**0.86**	0.54	0.47	*	*	*	0.48
PAMAP2	**0.87**	0.53	0.45	*	*	*	0.41

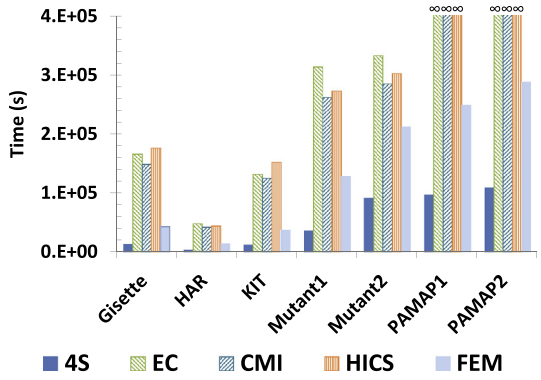


Fig. 6. Runtime (in seconds) of subspace search methods on real-world data sets. EC, CMI, and HICS did not finish within 5 days on the PAMAP data sets.

Table 7
Classification accuracy for real-world data sets. Highest values are in **bold**.

Data set	4S	Random Forest	FEM	CFS
Gisette	0.76	0.75	0.72	**0.84**
HAR	0.83	0.81	0.74	**0.85**
KIT	**0.97**	0.96	0.85	0.92
Mutant1	**0.99**	0.88	0.85	0.97
Mutant2	**0.99**	0.87	0.89	0.98
PAMAP1	**0.91**	0.71	0.69	0.87
PAMAP2	**0.93**	0.71	0.66	0.86

human subjects. There are of course different grouping patterns representing different types of activities. In any case, such correlations let records representing transient activities become outliers. This is intuitive because those activities are very random and do not feature any specific correlation among different parts of human bodies [37].

In Fig. 6, we show the wall-clock runtime (in seconds) for each subspace search method. We note that Apriori search techniques EC, CMI, and HICS did not finish within 5 days on the PAMAP data sets. The results show that 4S is much faster than all competitors.

Quality on classification. We here test 4S against the well-known Random Forest classifier [17], FEM for unsupervised feature selection, and CFS [16] for supervised feature selection. We skip other methods since previous experiments already show that 4S outperforms them. The classification accuracy (obtained by 10-fold cross validation) is in Table 7. Overall, 4S consistently yields better accuracy than Random Forest and FEM. It is comparable to CFS which has access to the class label. The results obtained show that the correlated subspaces found by 4S are also useful for data classification.

11.3. Discovering novel correlations on climate data

In this experiment, we evaluate whether 4S can be used to discover novel correlated subspaces in non-benchmark data. To this end, we apply 4S on a large real-world data set of climate and energy consumption measurements for an office building in Frankfurt, Germany [45]. After data pre-processing to handle missing values, our final data set contains 35 601 records and 251 dimensions. Example dimensions include room CO_2 concentration, indoor and outdoor temperature, temperature produced by heating systems, drinking water consumption, and electricity consumption by different devices, etc. Since this data set is unlabeled, we cannot calculate clustering/outlier detection/classification quality as above. Instead, we focus on detecting correlated subspaces, and investigate the discovered correlations. Our objective is thus to study how climate and energy consumption indicators interact with each other.

Overall, the results show that 4S detects many interesting high dimensional correlated subspaces—some were already known, others are novel. Below we report some of these subspaces whose intuitions can be straightforwardly perceived:

- wind speed, wind direction, outdoor temperature, outdoor CO_2 concentration, outdoor humidity
- amount of heating by wood, amount of heating by gas, outdoor temperature, temperature of room 401, temperature of room 402
- occupation of building, outdoor temperature, amount of drinking water consumption, electricity used for ventilator
- outdoor temperature, light intensity north (of the building), south, east, west, amount of solar heating produced
- air temperature supplied to the heating system, temperature of the heating boiler, and the amount of heating it produces (see Fig. 7)
- room temperature, amount of drinking water consumption, and room CO_2 concentration (see Fig. 8)

We make the observation that the correlations 4S detected range from linear ones (e.g., Fig. 8(c)) to non-linear functional (e.g., Fig. 7(b)), and non-linear non-functional (e.g., Fig. 8(b)). In terms of runtime, 4S only needs about 1.5 hours to explore the huge search space of this data set. In conclusion, the results suggest that 4S is a practical tool for scalable correlation analysis—an important feature which is crucial for understanding and mining large and high dimensional real-world data.

11.4. Succinctness of output

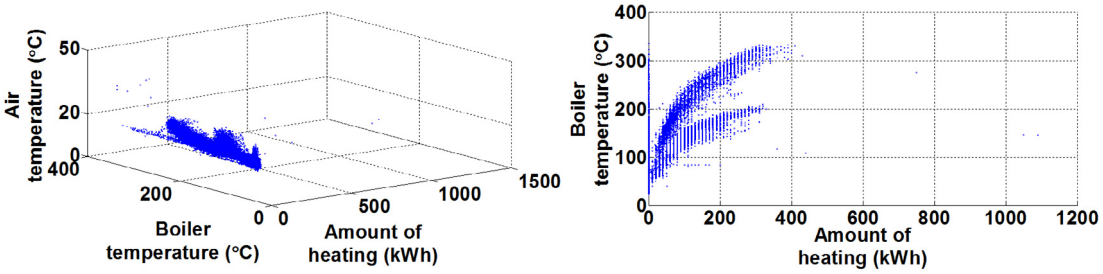
We study the benefits of our MDL subspace merge. Our performance metric is the reduction ratio, i.e., $\frac{m}{m'}$ where m and m' are the number of subspaces before and after merging. The results are in Fig. 9. We see that the merging phase achieves up to an order of magnitude reduction ratio. This verifies our claim that 4S produces a succinct set of overlapping correlated subspaces while still guaranteeing their quality (see for instance Section 11.2). By providing end users with a succinct set of correlated subspaces, in practice 4S facilitates manual inspection and post-analysis which benefit advanced applications based on the knowledge derived from the subspaces.

12. Related work

We categorize related literature into the following types:

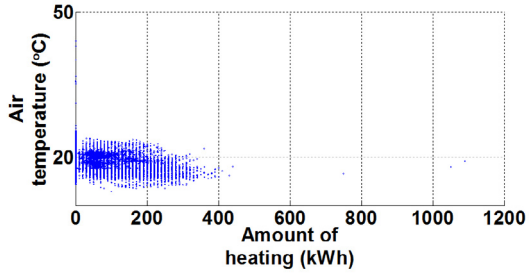
Feature selection. Related methods such as PCA and others [12, 22] select one subspace only. Since a dimension not relevant for one subspace may form a correlated subspace together with other dimensions, these simplified schemes likely miss important subspaces. 4S in turn is capable of mining multiple possibly overlapping subspaces.

Subspace search for specific tasks. There exist subspace search methods designed specifically for tasks such as outlier detection [2,21,30], clustering [3,46,26], and classification [47,16]. However,

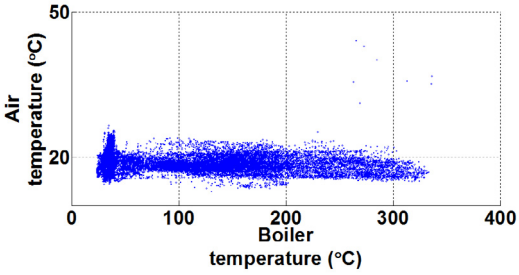


(a) 3-d space

(b) amount of heating vs. heating temperature

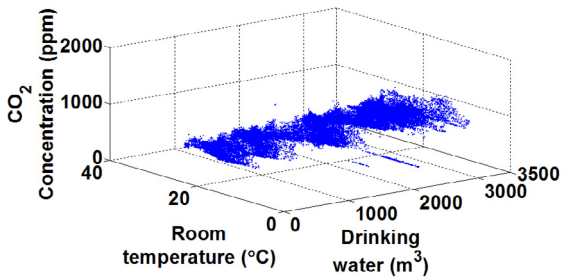


(c) amount of heating vs. air temperature

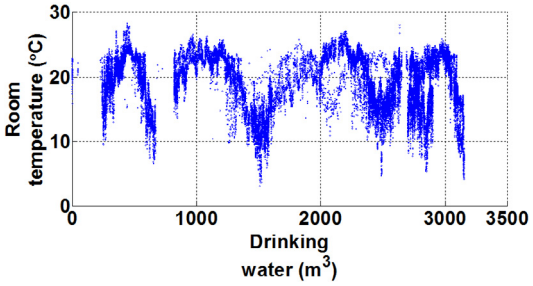


(d) heating temperature vs. air temperature

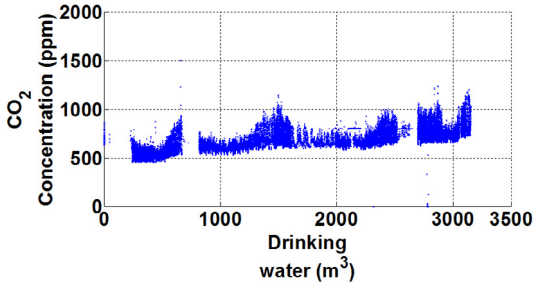
Fig. 7. Correlation among air temperature, heating temperature, amount of heating.



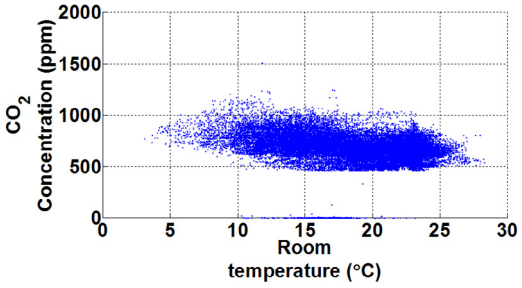
(a) 3-d space



(b) drinking water vs. room temperature



(c) drinking water vs. CO_2 concentration



(d) room temperature vs. CO_2 concentration

Fig. 8. Correlation among room temperature, drinking water consumption, and CO_2 concentration.

they are strongly coupled with the underlying tasks. For instance, supervised feature selection focuses on the correlation between each dimension and the class label, not the correlations among dimensions. As a result, they tend to have little effect on other tasks. 4S in turn is unsupervised, and further, not bound to any specific task. We have shown that its subspaces are useful for, e.g., outlier detection, clustering, and classification.

General subspace search. The methods in [9,19,20,33] are recent proposals to mine overlapping subspaces, abstracting from any concrete task. They explore the search space using an Apriori-style search. However, due to the monotonicity restriction, they detect only low dimensional subspaces. Such subspaces in turn likely are different projections of the same high dimensional correlated subspaces. This causes redundancy that is well-known for most subspace mining models [29,26]. Besides, they suffer severe scalability issue due to their expensive mining of correlated dimension pairs, and their levelwise search scheme which generates very many candidate subspaces. In contrast, 4S aims at a novel scalable search scheme that departs from these drawbacks. Experiments show that 4S yields good results with much less runtime.

Correlation measures. So far we use *Corr* as the correlation measure in 4S. Our method, however, is also applicable to other quadratic measures of (in)dependence [4,1,43] whose detailed investigation is reserved for future work. Recently, we also propose

Please cite this article in press as: H.-V. Nguyen et al., A Near-Linear Time Subspace Search Scheme for Unsupervised Selection of Correlated Features, Big Data Research (2014), http://dx.doi.org/10.1016/j.bdr.2014.07.004

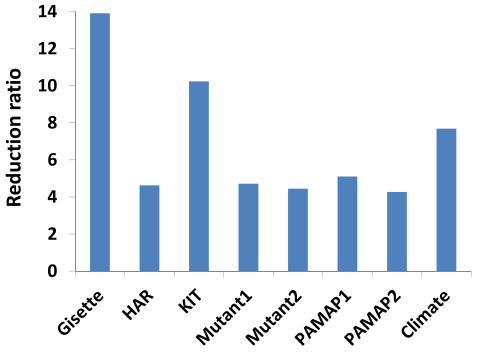


Fig. 9. Reduction ratio of our MDL merging phase. 4S achieves up to an order of magnitude reduction ratio.

MAC [32]—a multivariate correlation measure based on Shannon entropy. Studying MAC with our subspace search framework, again, is beyond the scope of this work.

13. Conclusions

Mining high dimensional correlated subspaces is a very challenging but important task for knowledge discovery in multi-dimensional data. We have introduced 4S, a new scalable subspace search scheme that addresses the issue. 4S works in three steps: scalable computation of $\mathcal{L}_2$, scalable mining of $\mathcal{L}_k$ ($k > 2$), and subspace merge to reconstruct fragmented subspaces and to reduce redundancy. Our experiments show that 4S scales to data sets of more than 1.5 million records and 5000 dimensions (i.e., more than 1 trillion subspaces). Not only being more efficient than existing methods, 4S also better detects high quality correlated subspaces that are useful for outlier mining, clustering, and classification. The superior performance of 4S compared to existing methods comes from (a) our new notion of correlated subspaces that has proved to be more general than existing notions and hence, allows to discover subspaces missed by such methods, (b) our scalable subspace search scheme that can discover high dimensional correlated subspaces, and (c) our subspace merge that can recover fragmented subspaces and remove redundancy.

Directions for future work include a systematic study our search scheme with different correlation measures, and the integration of the subspace merge into the correlation graph to perform an in-process removal of redundancy.

Acknowledgements

We thank Patricia Iglesias Sanchez for helpful discussion. Hoang Vu Nguyen is supported by the German Research Foundation (DFG) within GRK 1194. Emmanuel Müller is supported by the Young Investigator Group program of KIT as part of the German Excellence Initiative, and by a Post-Doctoral Fellowship of the Research Foundation – Flanders (FWO).

References

[1] S. Achard, Asymptotic properties of a dimension-robust quadratic dependence measure, C. R. Math. 346 (3) (2008) 213–216.
[2] C. Aggarwal, P. Yu, Outlier detection for high dimensional data, in: SIGMOD Conference, 2001, pp. 37–46.
[3] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, in: SIGMOD Conference, 1998, pp. 94–105.
[4] I.A. Ahmad, Q. Li, Testing independence by nonparametric kernel method, Stat. Probab. Lett. 34 (2) (1997) 201–210.
[5] N. Ailon, B. Chazelle, Faster dimension reduction, Commun. ACM 53 (2) (2010) 97–104.
[6] N. Alon, Y. Matias, M. Szegedy, The space complexity of approximating the frequency moments, in: STOC, 1996, pp. 20–29.
[7] K.S. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft, When is "nearest neighbor" meaningful?, in: ICDT, 1999, pp. 217–235.
[8] M. Breunig, H.-P. Kriegel, R. Ng, J. Sander, LOF: identifying density-based local outliers, in: SIGMOD Conference, 2000, pp. 93–104.
[9] C.H. Cheng, A.W.-C. Fu, Y. Zhang, Entropy-based subspace clustering for mining numerical data, in: KDD, 1999, pp. 84–93.
[10] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley–Interscience, New York, 2006.
[11] A. Dobra, M.N. Garofalakis, J. Gehrke, R. Rastogi, Processing complex aggregate queries over data streams, in: SIGMOD Conference, 2002, pp. 61–72.
[12] J.G. Dy, C.E. Brodley, Feature selection for unsupervised learning, J. Mach. Learn. Res. 5 (8) (2004) 909–921.
[13] D. Eppstein, M. Löffler, D. Strash, Listing all maximal cliques in sparse graphs in near-optimal time, in: ISAAC (1), 2010, pp. 403–414.
[14] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: KDD, 1996, pp. 226–231.
[15] P.D. Grünwald, The Minimum Description Length Principle, MIT Press, 2007.
[16] M.A. Hall, Correlation-based feature selection for discrete and numeric class machine learning, in: ICML, 2000, pp. 359–366.
[17] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 832–844.
[18] D. Jiang, J. Pei, Mining frequent cross-graph quasi-cliques, ACM Trans. Knowl. Discov. Data 2 (4) (2009).
[19] K. Kailing, H.-P. Kriegel, P. Kröger, S. Wanka, Ranking interesting subspaces for clustering high dimensional data, in: PKDD, 2003, pp. 241–252.
[20] F. Keller, E. Müller, K. Böhm, HiCS: High contrast subspaces for density-based outlier ranking, in: ICDE, 2012, pp. 1037–1048.
[21] H.-P. Kriegel, P. Kröger, E. Schubert, A. Zimek, Outlier detection in arbitrarily oriented subspaces, in: ICDM, 2012, pp. 379–388.
[22] M.H.C. Law, M.A.T. Figueiredo, A.K. Jain, Simultaneous feature selection and clustering using mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 26 (9) (2004) 909–921.
[23] A. Lazarevic, V. Kumar, Feature bagging for outlier detection, in: KDD, 2005, pp. 157–166.
[24] J.H. Macke, P. Berens, A.S. Ecker, A.S. Tolias, M. Bethge, Generating spike trains with specified correlation coefficients, Neural Comput. 21 (2) (2009) 397–423.
[25] M. Mampaey, J. Vreeken, Summarizing categorical data by clustering attributes, Data Min. Knowl. Discov. 26 (1) (2013) 130–173.
[26] E. Müller, I. Assent, S. Günnemann, R. Krieger, T. Seidl, Relevant subspace clustering: mining the most interesting non-redundant concepts in high dimensional data, in: ICDM, 2009, pp. 377–386.
[27] E. Müller, I. Assent, S. Günnemann, T. Seidl, Scalable density-based subspace clustering, in: CIKM, 2011, pp. 1077–1086.
[28] E. Müller, I. Assent, R. Krieger, S. Günnemann, T. Seidl, DensEst: density estimation for data mining in high dimensional spaces, in: SDM, 2009, pp. 175–186.
[29] E. Müller, S. Günnemann, I. Assent, T. Seidl, Evaluating clustering in subspace projections of high dimensional data, Proc. VLDB Endow. 2 (1) (2009) 1270–1281.
[30] E. Müller, M. Schiffer, T. Seidl, Statistical selection of relevant subspace projections for outlier ranking, in: ICDE, 2011, pp. 434–445.
[31] H.V. Nguyen, E. Müller, K. Böhm, 4S: scalable subspace search scheme overcoming traditional apriori processing, in: BigData Conference, 2013, pp. 359–367.
[32] H.V. Nguyen, E. Müller, J. Vreeken, P. Efros, K. Böhm, Multivariate maximal correlation analysis, in: ICML, 2014, pp. 775–783.
[33] H.V. Nguyen, E. Müller, J. Vreeken, F. Keller, K. Böhm, CMI: an information-theoretic contrast measure for enhancing subspace cluster and outlier detection, in: SDM, 2013, pp. 198–206.
[34] N. Pham, R. Pagh, A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data, in: KDD, 2012, pp. 877–885.
[35] M. Rao, Y. Chen, B.C. Vemuri, F. Wang, Cumulative residual entropy: a new measure of information, IEEE Trans. Inf. Theory 50 (6) (2004) 1220–1228.
[36] M. Rao, S. Seth, J.-W. Xu, Y. Chen, H. Tagare, J.C. Príncipe, A test of independence based on a generalized correlation function, Signal Process. 91 (1) (2011) 15–27.
[37] A. Reiss, D. Stricker, Towards global aerobic activity monitoring, in: PETRA, 2011, p. 12.
[38] D.N. Reshef, Y.A. Reshef, H.K. Finucane, S.R. Grossman, G. McVean, P.J. Turnbaugh, E.S. Lander, M. Mitzenmacher, P.C. Sabeti, Detecting novel associations in large data sets, Science 334 (6062) (2011) 1518–1524.
[39] F. Rusu, A. Dobra, Sketches for size of join estimation, ACM Trans. Database Syst. 33 (3) (2008).

Please cite this article in press as: H.-V. Nguyen et al., A Near-Linear Time Subspace Search Scheme for Unsupervised Selection of Correlated Features, Big Data Research (2014), http://dx.doi.org/10.1016/j.bdr.2014.07.004

[40] L. Schietgat, F. Costa, J. Ramon, L.D. Raedt, Effective feature construction by maximum common subgraph sampling, Mach. Learn. 83 (2) (2011) 137–161.

[41] M.C. Schmidt, N.F. Samatova, K. Thomas, B.-H. Park, A scalable, parallel algorithm for maximal clique enumeration, J. Parallel Distrib. Comput. 69 (4) (2009) 417–428.

[42] S. Seth, M. Rao, I. Park, J.C. Príncipe, A unified framework for quadratic measures of independence, IEEE Trans. Signal Process. 59 (8) (2011) 3624–3635.

[43] A.J. Smola, A. Gretton, L. Song, B. Schölkopf, A Hilbert space embedding for distributions, in: ALT, 2007, pp. 13–31.

[44] K. Tzoumas, A. Deshpande, C.S. Jensen, Lightweight graphical models for selectivity estimation without independence assumptions, Proc. VLDB Endow. 4 (11) (2011) 852–863.

[45] A. Wagner, T. Lützkendorf, K. Voss, G. Spars, A. Maas, S. Herkel, Performance analysis of commercial buildings—results and experiences from the German demonstration program 'energy optimized building (EnOB)', Energy Build. 68 (2014) 634–638.

[46] M.L. Yiu, N. Mamoulis, Iterative projected clustering by subspace mining, IEEE Trans. Knowl. Data Eng. 17 (2) (2005) 176–189.

[47] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (2004) 1205–1224.

Please cite this article in press as: H.-V. Nguyen et al., A Near-Linear Time Subspace Search Scheme for Unsupervised Selection of Correlated Features, Big Data Research (2014), http://dx.doi.org/10.1016/j.bdr.2014.07.004