Knowl Inf Syst (2016) 47:99–129
DOI 10.1007/s10115-015-0835-6

REGULAR PAPER

Efficient discovery of contrast subspaces for object explanation and characterization

Lei Duan[1] · Guanting Tang[2] · Jian Pei[2] · James Bailey[3] · Guozhu Dong[4] · Vinh Nguyen[3] · Akiko Campbell[5] · Changjie Tang[1]

Received: 30 November 2014 / Accepted: 7 April 2015 / Published online: 26 April 2015

Abstract We tackle the novel problem of mining contrast subspaces. Given a set of multidimensional objects in two classes C_+ and C_- and a query object o, we want to find the top-k subspaces that maximize the ratio of likelihood of o in C_+ against that in C_-. Such subspaces are very useful for characterizing an object and explaining how it differs between two classes. We demonstrate that this problem has important applications, and, at the same time, is very challenging, being MAX SNP-hard. We present CSMiner, a mining method that uses kernel density estimation in conjunction with various pruning techniques.

✉ Lei Duan
leiduan@scu.edu.cn

Guanting Tang
gta9@cs.sfu.ca

Jian Pei
jpei@cs.sfu.ca

James Bailey
baileyj@unimelb.edu.au

Guozhu Dong
guozhu.dong@wright.edu

Vinh Nguyen
vinh.nguyen@unimelb.edu.au

Akiko Campbell
acampbell@pac.bluecross.ca

Changjie Tang
cjtang@scu.edu.cn

1 School of Computer Science, Sichuan University, Chengdu, Sichuan, China

2 School of Computing Science, Simon Fraser University, Burnaby, BC, Canada

3 Department of Computing and Information Systems, The University of Melbourne, Melbourne, VIC, Australia

4 Department of Computer Science and Engineering, Wright State University, Dayton, OH, USA

5 Pacific Blue Cross, Burnaby, BC, Canada

We experimentally investigate the performance of CSMiner on a range of data sets, evaluating its efficiency, effectiveness, and stability and demonstrating it is substantially faster than a baseline method.

Keywords Contrast subspace · Kernel density estimation · Likelihood contrast

1 Introduction

Imagine you are a medical doctor facing a patient with symptoms that include being overweight, shortness of breath, and tiredness. You want to check the patient against two specific possible diseases: coronary artery disease and adiposity. Note that clogged arteries are among the top-5 most commonly misdiagnosed diseases. You have available a set of reference samples for both diseases. Then, you may naturally ask "In what aspects is this patient most similar to cases of coronary artery disease and, at the same time, dissimilar to adiposity?"

The above motivational scenario cannot be addressed well using existing data mining methods and thus suggests a novel data mining problem. In a multidimensional data set of two classes, given a query object and a target class, we want to find the subspaces where the query object is most likely to belong to the target class versus the other class. We call such subspaces *contrast subspaces*, since they contrast the likelihood of the query object in the target class against the other class. Mining contrast subspaces is an interesting problem with important applications. As another example, when an analyst in an insurance company is investigating a suspicious claim, she may want to compare this suspicious case against samples of frauds and normal claims. A useful question to ask is "In what aspects is this suspicious case most similar to fraudulent cases and different from normal claims?". In other words, finding the contrast subspaces for the suspicious claim is informative for the analyst and serves as a useful input for deeper exploration.

While there are many existing studies on outlier detection and contrast mining, they focus on collective patterns that are shared by many cases of the target class. The contrast subspace mining problem addressed here is different. It focuses on one query object and finds the customized contrast subspaces. This critical difference makes the problem formulation, the suitable applications and the mining methods rather different. We will review related work and explain the differences in more detail in Sect. 2.

Challenges: To tackle the problem of mining contrast subspaces, we need to address several technical challenges. First, we need to have a simple yet informative contrast measure to quantify the similarity between the query object and the target class and the difference between the query object and the other class.

Second, the problem of mining contrast subspaces is computationally challenging. Exhaustive search, which enumerates every non-empty subspace and computes the contrast measure, is very costly on data sets with a non-trivial dimensionality.

Third, one might attempt a brute-force method to tackle the contrast mining problem. One major obstacle preventing effective pruning is that the contrast measure does not have any monotonicity with respect to the subspace–superspace relationship.

Our contributions: Besides introducing the new problem of mining contrast subspaces, we make several contributions in this paper.

– We use the ratio of the likelihood of the query object in the target class against that in the other class as the contrast measure. This is essentially the Bayes factor on the query object and comes with a well-recognized explanation [15].

– We show that the problem of contrast subspace mining is MAX SNP-hard and thus does not allow polynomial time approximation methods unless P = NP. Therefore, the only hope is to develop heuristics that may work well in practice.
– We develop pruning techniques based on bounds of likelihood and contrast ratio. Our experimental results on real data sets clearly verify the effectiveness, stability, and efficiency of our method.

Organization: The rest of the paper is organized as follows. We review related work in Sect. 2. In Sect. 3, we formalize the problem and analyze it theoretically. We present a heuristic method in Sect. 4 and evaluate our method empirically using real data sets in Sect. 5. We conclude the paper in Sect. 6.

2 Related work

Our study is related to the existing work on contrast mining, subspace outlier detection, and typicality queries. We review the related work briefly here.

Contrast mining discovers patterns and models that manifest drastic differences between data sets. Dong and Bailey [9] presented a comprehensive review of contrast mining, together with a range of real-life applications. Some of the best known types of contrast patterns include emerging patterns [10], contrast sets [3], and subgroups [25]. Although their definitions vary, the mining methods share many similarities [19].

Contrast pattern mining identifies patterns by considering all objects of all classes in the complete pattern space. Orthogonally, contrast subspace mining focuses on one object and identifies subspaces where a query object demonstrates the strongest overall similarity to one class against the other. These two mining problems are fundamentally different. To the best of our knowledge, the contrast subspace mining problem has not been systematically explored in the data mining literature.[1]

Subspace outlier detection discovers objects that significantly deviate from the majority in some subspaces. Data sets from real life often have very high dimensionalities. Due to the curse of dimensionality, measurements designed to calculate the differences between an object and the other objects, such as distance and probability density, become meaningless in the full space [4].

Given a multidimensional database, subspace outlier detection aims to identify a set of subspaces, where the outlier objects drastically deviate from the majority. It is different from our study. In contrast subspace mining, the query object may or may not be an outlier. We are trying to find the top-k subspaces, in which a query object is the most typical in the current class and is very unlikely to occur in other classes. Some recent studies find subspaces that may contain substantial outliers. Böhm et al [5] and Keller et al [16] proposed statistical approaches *HiCS* and *CMI* to selecting subspaces for a multidimensional database, where there may exist outliers with high deviations. Both *HiCS* and *CMI* differ from our method. Technically, they choose subspaces for all outliers in a given database, while our method chooses the most contrasting subspaces for a query object. In *HiCS* and *CMI*, *contrast* refers to the differences between the assumptions on whether the subspaces are mutually independent or not. In our work, *contrast* is defined as the differences of the likelihoods that a query belongs to the given class or not.

[1] While [8] presented a contrast-pattern length based algorithm to detection global outliers, their problem setting is different from ours.

Kriegel et al [18] introduced SOD, a method to detect outliers in axis-parallel subspaces. For each outlier detected, the method selects a hyperplane, where the outlier deviates significantly from the neighbors of the outlier in the full space as references. SOD also differs from our work. First, SOD is still an outlier detection method, and the hyperplane is a byproduct of the detection process. Our method does not detect outliers at all. Second, the input data are different. Our work requires the input data to have class labels, while SOD does not have this requirement.

Our method uses probability density to estimate the likelihood of a query object belonging to different classes. There exist density-based outlier detection methods, such as [1,6,13,17]. Our method is different from those, since we do not target outlier objects and instead aim to analyze any type of object.

Hua et al [14] introduced a novel top-k *typicality query*, which ranks objects according to their typicality in a data set or a class of objects. Although both [14] and our work use density estimation methods to calculate the typicality/likelihood of a query object with respect to a set of data objects, typicality queries [14] do not consider subspaces. [14] aimed to find the most typical data objects according to the query object; in contrast, we find the most contrasting subspaces for a query object.

Cai et al [7] proposed a method that adopted concepts from human cognition, to answer the top-k typicality queries. The typicality of an object with respect to a set of data objects was calculated based on the similarity and support of the object with respect to the set of data objects. Again, the problem setting and the method differ from our work.

We tackled the problem of contrast subspace mining in [11], a preliminary version of this paper. Compared to that work, in this paper, we present a complete complexity analysis, provide a more detailed description of the key steps in our method and perform more extensive empirical evaluations, including using different bandwidths and kernel.

3 Problem formulation and analysis

In this section, we first formulate the problem. Then, we recall the basics of kernel density estimation for estimating the probability density of objects. Last, we investigate the complexity of the problem.

3.1 Problem definition

Let $D = \{D_1, \ldots, D_d\}$ be a d-dimensional space, where the domain of D_i is $\mathbb{R}$, the set of real numbers. A *subspace* $S \subseteq D$ ($S \neq \emptyset$) is a subset of D. We also call D the *full space*.

Consider an object o in space D. We denote by $o.D_i$ the value of o in dimension D_i ($1 \leq i \leq d$). For a subspace $S = \{D_{i_1}, \ldots, D_{i_l}\} \subseteq D$, the *projection* of o in S is $o^S = (o.D_{i_1}, \ldots, o.D_{i_l})$. For a set of objects $O = \{o_j \mid 1 \leq j \leq n\}$, the *projection* of O in S is $O^S = \{o_j^S \mid o_j \in O, 1 \leq j \leq n\}$.

Given a set of objects O, we assume a latent distribution $\mathcal{Z}$ that generates the objects in O. For a query object q, denote by $L_D(q \mid \mathcal{Z})$ the likelihood of q being generated by $\mathcal{Z}$ in full space D. The posterior probability of q given O, denoted by $L_D(q \mid O)$, can be estimated by $L_D(q \mid \mathcal{Z})$. For a non-empty subspace S ($S \subseteq D$, $S \neq \emptyset$), denote by $\mathcal{Z}^S$ the projection of $\mathcal{Z}$ in S. The *subspace likelihood* of object q with respect to $\mathcal{Z}$ in S, denoted by $L_S(q \mid \mathcal{Z})$, can be used to estimate the posterior probability of object q given O in S, denoted by $L_S(q \mid O)$.

In this paper, we assume that the objects in O belong to two classes, C_+ and C_-, exclusively in full space D. Thus, $O = O_+ \cup O_-$ and $O_+ \cap O_- = \emptyset$, where O_+ and O_- are the subsets of

objects of O belonging to C_+ and C_-, respectively. Given a query object q, we are interested in how likely q belongs to C_+ and does not belong to C_-. To measure these two factors comprehensively, we define the *likelihood contrast* as $LC(q) = \frac{L_D(q|O_+)}{L_D(q|O_-)}$.

Likelihood contrast is essentially the Bayes factor[2] of object q as the observation. In other words, we can regard O_+ and O_- as representing two models, and we need to choose one of them based on query object q. Consequently, the ratio of likelihoods indicates the plausibility of model represented by O_+ against that by O_-. Jeffreys [15] gave a scale for interpretation of Bayes factor. When $LC(q)$ is in the ranges of < 1, 1–3, 3–10, 10–30, 30–100, and over 100, respectively, the strength of the evidence is negative, barely worth mentioning, substantial, strong, very strong, and decisive.

We can extend likelihood contrast to subspaces. For a non-empty subspace $S \subseteq D$, we define the likelihood contrast in the subspace as $LC_S(q) = \frac{L_S(q|O_+)}{L_S(q|O_-)}$. To avoid triviality in subspaces where $L_S(q \mid O_+)$ is very small, we introduce a minimum likelihood threshold $\delta > 0$ and consider only the subspaces S where $L_S(q \mid O_+) \geq \delta$. The number of likelihood contrast subspaces will be reduced with larger δ.

Now, we formally define the problem. Given a multidimensional data set O in full space D, a query object q, a minimum likelihood threshold $\delta > 0$ and a parameter $k > 0$, the *problem of mining contrast subspaces* is to find the top-k subspaces S ordered by the subspace likelihood contrast $LC_S(q)$ subject to $L_S(q \mid O_+) \geq \delta$.

3.2 Kernel density estimation

We can use kernel density estimation to estimate the likelihood $L_S(q \mid O)$. Given a set of objects O, we denote by $\hat{f}_S(q, O)$ the density of a query object q in subspace S. Following [22], the general formula for multivariate kernel density estimation with kernel K and bandwidth parameter h_S in subspace S is defined as follows

$$\hat{f}_S(q, O) = \hat{f}_S(q^S, O) = \frac{1}{|O|h_S^{|S|}} \sum_{o \in O} K\left\{\frac{1}{h_S}(q - o)\right\} \tag{1}$$

Choosing K to be a radially symmetric unimodal[3] probability density function, in this paper, we adopt the Gaussian kernel

$$K(x) = \frac{1}{(2\pi)^{|S|/2}} e^{-\frac{1}{2}x^T x} \tag{2}$$

which is natural and widely used in density estimation.

This then leads to

$$\hat{f}_S(q, O) = \hat{f}_S(q^S, O) = \frac{1}{|O|(\sqrt{2\pi}h_S)^{|S|}} \sum_{o \in O} \exp\left(\frac{-dist_S(q, o)^2}{2{h_S}^2}\right)$$

where $dist_S(q, o)^2 = \sum_{D_i \in S}(q.D_i - o.D_i)^2$.

Silverman [22] suggested that the optimal bandwidth value for smoothing normally distributed data with unit variance is $h_{S\_opt} = A(K)|O|^{-1/(|S|+4)}$, where $A(K) = \{4/(|S| + 2)\}^{1/(|S|+4)}$ for the Gaussian kernel.

[2] Generally, given a set of observations Q, the plausibility of two models M_1 and M_2 can be assessed by the Bayes factor $K = \frac{Pr(Q|M_1)}{Pr(Q|M_2)}$.

[3] If it is not unimodal, then there could be multiple peaks at different distances from the query, which is counter to intuition. Similarly, we have no basis for preferring any direction over another, so symmetry is natural.

As the kernel is radially symmetric and the data are not normalized in subspaces, we can use a single scale parameter σ_S in subspace S and set $h_S = \sigma_S \cdot h_{S\_opt}$. As [22] suggested, a reasonable choice for σ_S is the root of the average marginal variance in S.

Using kernel density estimation, we can estimate $L_S(q \mid O)$ as

$$L_S(q \mid O) = \hat{f}_S(q, O) = \frac{1}{|O|(\sqrt{2\pi}h_S)^{|S|}} \sum_{o \in O} \exp\left(\frac{-dist_S(q,o)^2}{2{h_S}^2}\right) \tag{3}$$

Correspondingly, the likelihood contrast of object q in subspace S is given by

$$LC_S(q, O_+, O_-) = \frac{\hat{f}_S(q, O_+)}{\hat{f}_S(q, O_-)} = \frac{|O_-|}{|O_+|} \cdot \left(\frac{h_{S_-}}{h_{S_+}}\right)^{|S|} \cdot \frac{\sum_{o \in O_+} \exp\left(\frac{-dist_S(q,o)^2}{2{h_{S_+}}^2}\right)}{\sum_{o \in O_-} \exp\left(\frac{-dist_S(q,o)^2}{2{h_{S_-}}^2}\right)} \tag{4}$$

We often omit O_+ and O_- and write $LC_S(q)$ if O_+ and O_- are clear from context.

3.3 Complexity analysis

Before developing any algorithms to tackle the contrast subspace mining problem, let us first investigate its complexity. We will show that the contrast subspace mining problem is MAX SNP-hard by constructing a linear reduction (L-reduction for short) from the emerging pattern mining problem [10], which was been shown to be MAX SNP-hard [23]. The L-reduction linearly preserves approximability features of the original problem after the transformation, thus the name "linear reduction".

To make the discussion self-contained, a brief description of the emerging pattern mining problem is given as follows. Let $D' = \{D'_1, D'_2, \ldots, D'_d\}$ denote a set of d items. A transaction o'_i is represented by a binary vector of length d whose element $o'_{ij} = 1$ if item D'_j is present, and 0 otherwise. A pattern S' is a subset of items in D'. A transaction o'_i satisfies S' if $o'_{ij} = 1,\ \forall D'_j \in S'$. A transaction database O' is a set of transactions. Let $\mathrm{Sat}_{O'}(S')$ denote the set of transactions in O' satisfying S'.

Definition 1 (*Emerging pattern mining* (*EP*)) Given two transaction databases O'_+ and O'_-, find the pattern S' such that the cost function $c_{\boldsymbol{EP}}(S') = |\mathrm{Sat}_{O'_+}(S')|$ is maximized subject to the feasibility condition $|Sat_{O'_-}(S')| = 0$.

We consider the following simplified version of the contrast subspace mining problem, where the bandwidth parameters h_{S_+} and h_{S_-} for all subspaces are set to the same value h.

Definition 2 (*Contrast subspace mining* (*CS*)) Given $\{q, O_+, O_-\}$ where q is the query and O_+ and O_- are the two classes, find the subspace S maximizing the cost function

$$c_{\boldsymbol{CS}}(S, q) = \sum_{o \in O_+} \exp\left(\frac{-dist_S(q,o)^2}{2h^2}\right) \Big/ \sum_{o \in O_-} \exp\left(\frac{-dist_S(q,o)^2}{2h^2}\right)$$

(which is equivalent to the likelihood contrast, up to a constant multiplicative factor $\frac{|O_-|}{|O_+|}$).

In addition, we define the *complete contrast subspace mining problem* as follows:

Definition 3 (*Complete contrast subspace mining* (*Complete-CS*)) Given $\{O_+, O_-\}$ find the subspace S such that the cost function

$$c(S) = \max_{o_i \in O_+} c_{\boldsymbol{CS}}(S, q = o_i)$$

is maximized.

It can be seen that Complete-**CS** can be solved by solving at most $|O_+|$ **CS** sub-problems corresponding to unique data points in O_+. We will now prove that Complete-**CS** is MAX SNP-hard, via the following reduction from the emerging pattern mining problem.

Reduction 1 *The **EP** $\rightarrow$ Complete-**CS** reduction:*

- *For each item D'_i, set up a corresponding dimension D_i.*
- *For each transaction $o'_i \in O'_+$, insert 2 copies of o'_i into O_+.*
- *For each transaction $o'_i \in O'_-$, insert $2|O'_+|$ identical data points o'_i into O_-.*
- *Insert 1 item (a numeric vector) with all 1's into O_-.*
- *Let h be an arbitrary user-specified bandwidth parameter, replace each occurrence of the 0 value in $O = O_+ \cup O_-$ with a unique value in the set $\{2\gamma h, 3\gamma h, 4\gamma h \ldots\}$ where γ is some fixed large constant.*
- *Replace each occurrence of the value 1 in O with γh where γ is the same as the one used above.*

This transformation can be done in $\mathcal{O}(|O_+||O_-|)$ time. An example illustrating the transformation is given in Table 1.

Theorem 1 *The reduction **EP** $\rightarrow$ Complete-**CS** defined above is an L-reduction, denoted by **EP** $\rightarrow_L$ Complete-**CS**.*

For completeness, the formal definition of the L-reduction [20] is given as follows:

Definition 4 (*L-Reduction*) Let $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$ be two optimization problems. We say that $\mathbf{\Pi}_1$ L-reduces to $\mathbf{\Pi}_2$ if there are two polynomial time algorithms f, g and constants $\alpha, \beta > 0$ such that, for any instance I of $\mathbf{\Pi}_1$, $f(I)$ forms an instance of $\mathbf{\Pi}_2$ and

- (c1) $OPT(f(I)) \leq \alpha OPT(I)$ where $OPT(.)$ denotes the optimal value of the respective optimization problem.

Table 1 An example transformation from a transaction database to a numeric data set according to the **EP** $\rightarrow$ Complete-**CS** reduction

Database	Transactions	O_+	O_-
O'_+	[0, 1, 1, 0]	$[2\gamma h, 1\gamma h, 1\gamma h, 3\gamma h]$	
		$[4\gamma h, 1\gamma h, 1\gamma h, 5\gamma h]$	
	[0, 1, 0, 0]	$[6\gamma h, 1\gamma h, 7\gamma h, 8\gamma h]$	
		$[9\gamma h, 1\gamma h, 10\gamma h, 11\gamma h]$	
O'_-	[1, 1, 0, 0]		$[1\gamma h, 1\gamma h, 12\gamma h, 13\gamma h]$
			$[1\gamma h, 1\gamma h, 14\gamma h, 15\gamma h]$
			$[1\gamma h, 1\gamma h, 16\gamma h, 17\gamma h]$
			$[1\gamma h, 1\gamma h, 18\gamma h, 19\gamma h]$
	[0, 0, 0, 1]		$[20\gamma h, 21\gamma h, 22\gamma h, 1\gamma h]$
			$[23\gamma h, 24\gamma h, 25\gamma h, 1\gamma h]$
			$[26\gamma h, 27\gamma h, 28\gamma h, 1\gamma h]$
			$[29\gamma h, 30\gamma h, 31\gamma h, 1\gamma h]$
			$[1\gamma h, 1\gamma h, 1\gamma h, 1\gamma h]$

– (c2) Given any solution s of $f(I)$, algorithm g produces a solution $g(s)$ of I satisfying $|c_{\mathbf{\Pi}_1}(g(s)) - OPT(I)| \leq \beta |c_{\mathbf{\Pi}_2}(s) - OPT(f(I))|$, where $c_{\mathbf{\Pi}_i}(.)$ denotes the cost function of the corresponding optimization problem.

Proof First, we note that for any bandwidth value h, we can set γ to a large value such that $\exp\left(\frac{-dist_S(q,o)^2}{2h^2}\right)$ can be arbitrarily close to 0 for all $q \in O$ such that $q^S \neq o^S$. The cost function for **CS** can be computed as

$$c_{\mathbf{CS}}(S,q) = \frac{\sum_{o \in O_+} \exp\left(\frac{-dist_S(q,o)^2}{2h^2}\right)}{\sum_{o \in O_-} \exp\left(\frac{-dist_S(q,o)^2}{2h^2}\right)} = \frac{|O_+^{S,q}| + \epsilon_+(S,q)}{|O_-^{S,q}| + \epsilon_-(S,q)} \tag{5}$$

where $O^{S,q}$ denotes the set of data points in O having values identical to q in the subspace S, and

$$\epsilon_+(S,q) = \sum_{o \in O_+ \setminus O_+^{S,q}} \exp\left(\frac{-dist_S(q,o)^2}{2h^2}\right),$$

$$\epsilon_-(S,q) = \sum_{o \in O_- \setminus O_-^{S,q}} \exp\left(\frac{-dist_S(q,o)^2}{2h^2}\right).$$

Let $M > 1$ be the maximum integer value such that $M\gamma h$ is a value occurring in O (e.g., $M = 31$ in the example in Table 1). Then $|S|\gamma^2 h^2 < dist_S(q,o)^2 < M^2|S|\gamma^2 h^2$ for all $o \in O_+ \cup O_-$. Thus

$$(|O_+| - |O_+^{S,q}|)\exp\left(-|S|\gamma^2 M^2\right) < \epsilon_+(S,q) < (|O_+| - |O_+^{S,q}|)\exp\left(-|S|\gamma^2\right) \ll 1$$

and similarly

$$(|O_-| - |O_-^{S,q}|)\exp\left(-|S|\gamma^2 M^2\right) < \epsilon_-(S,q) < (|O_-| - |O_-^{S,q}|)\exp\left(-|S|\gamma^2\right) \ll 1$$

Note that $\lim_{\gamma \to \infty} \epsilon_+(S,q) = 0$ and $\lim_{\gamma \to \infty} \epsilon_-(S,q) = 0$. Now, it can be seen that:

– If a pattern S' is an emerging pattern, then by construction *at least one* object $q \in O_+$ must have $|O_+^{S,q}| \geq 2$ and $|O_-^{S,q}| = 1$. This is because S' only appears in O'_+, and for each transaction $o'_i \in O'_+$, we have inserted 2 copies of o'_i into O_+. On the other hand, S' does not appear in O'_- and the only object having values identical to q in the subspace S is the object containing all γh's. Therefore,

$$c_{\mathbf{CS}}(S,q) = \frac{|O_+^{S,q}| + \epsilon_+(S,q)}{|O_-^{S,q}| + \epsilon_-(S,q)} \geq \frac{2 + \epsilon_+(S,q)}{1 + \epsilon_-(S,q)} > 1 \tag{6}$$

– If a pattern S' is *not* an emerging pattern, then by construction *all* objects $q \in O_+$ must have $|O_-^{S,q}| \geq |O_+^{S,q}| + 1 > |O_+^{S,q}|$. Therefore,

$$c_{\mathbf{CS}}(S,q) = \frac{|O_+^{S,q}| + \epsilon_+(S,q)}{|O_-^{S,q}| + \epsilon_-(S,q)} < 1 \tag{7}$$

With these observations, we are ready to prove the main complexity result. We need to verify that the reduction **EP** $\rightarrow$ Complete-**CS** satisfies the two conditions (c1) and (c2) of the L-reduction.

– (c1) For any instance I of **EP**, if S' is the most frequent emerging pattern with $c_{\boldsymbol{EP}}(S') = |\mathrm{Sat}_{O'_+}(S')|$ and $|\mathrm{Sat}_{O'_-}(S')| = 0$, then the corresponding optimal S solution for Complete-**CS** must have a cost value of

$$c(S) = \frac{2|\mathrm{Sat}_{O'_+}(S')| + \epsilon_+(S,q)}{1 + \epsilon_-(S,q)} \simeq 2|\mathrm{Sat}_{O'_+}(S')| = 2c_{\boldsymbol{EP}}(S') \tag{8}$$

where q is any data point in O_+ corresponding to the transaction containing pattern S'. This is because for each transaction o'_i containing S' in O'_+, we have inserted 2 copies of o'_i into O_+. The '1' in the denominator is due to the object containing all γh in O_-. Thus condition 1 is satisfied with $\alpha = 2$ when γ is sufficiently large.

– (c2) For any solution S of Complete-**CS**, if $c(S) = \lambda \geq 2$, then the corresponding pattern S' constructed from S will be an emerging pattern. Further, let $[\lambda]$ be the nearest integer to λ. Then $[\lambda]$ must be even, and $[\lambda]/2$ will be the cost of the corresponding **EP** problem. Let λ^* denote the optimal cost of Complete-**CS**, then

$$\left|\frac{[\lambda]}{2} - \frac{[\lambda^*]}{2}\right| = \frac{1}{2}|[\lambda] - [\lambda^*]| \simeq \frac{1}{2}|\lambda - \lambda^*| \leq |\lambda - \lambda^*| \tag{9}$$

Thus condition 2 is satisfied with $\beta = 1$.

□

Since **EP** $\rightarrow_L$ Complete-**CS**, if there exists a polynomial time approximation algorithm for Complete-**CS** with performance guarantee $1 - \epsilon$, then there exists a polynomial time approximation algorithm for **EP** with performance guarantee $1 - \alpha\beta\epsilon$. Since **EP** is MAX SNP-hard, it follows that Complete-**CS** must also be MAX SNP-hard.

Last, we draw the connection between Complete-**CS** and **CS**.

Theorem 2 *If there exists a polynomial time approximation scheme (PTAS) for **CS**, then there must also be a PTAS for Complete-**CS**.*

Proof This is straightforward, as Complete-**CS** can be solved by a series of $|O_+|$ **CS** problems. □

Unless P = NP, there exists no PTAS for Complete-**CS**, implying no PTAS for **CS**.

The above theoretical result indicates that the problem of mining contrast subspaces is even hard to approximate—it is impossible (unless P = NP) to design a good approximation algorithm. In the rest of the paper, we turn to practical heuristic methods.

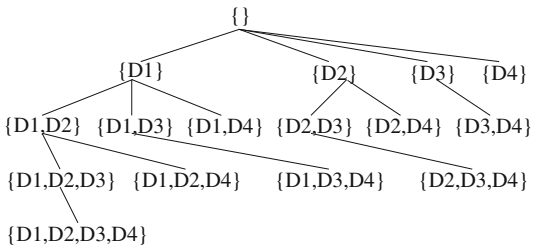
4 Mining methods

In this section, we first describe a baseline method that examines every possible non-empty subspace. Then, we present the design of our method CSMiner (for Contrast Subspace Miner) that employs smarter strategies for search.

4.1 A baseline method

A baseline naive method enumerates all possible non-empty spaces S and calculates the exact values of both $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$, since both $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$ are not monotonic with respect to the subspace–superspace relationship. Then, it returns the top-k

Fig. 1 A set enumeration tree



Algorithm 1 The baseline algorithm

Input: q: query object, O_+: objects belonging to C_+, O_-: objects belonging to C_-, δ: likelihood threshold, k: positive integer
Output: k subspaces with the highest likelihood contrast
1: let Ans be the current top-k list of subspaces, initialize Ans as k null subspaces associated with likelihood contrast 0
2: traverse the subspace set enumeration tree in a depth-first search manner
3: **for** each subspace S **do**
4: compute σ_{S+}, σ_{S-}, h_{opt};
5: compute $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$ using Equation 3;
6: **if** $L_S(q \mid O_+) \geq \delta$ and $\exists S' \in Ans$ s.t. $\frac{L_S(q|O_+)}{L_S(q|O_-)} > LC_{S'}(q)$ **then**
7: insert S into Ans and remove S' from Ans;
8: **end if**
9: **end for**
10: **return** Ans;

subspaces S with the largest $LC_S(q)$ values. To ensure the completeness and efficiency of subspace enumeration, the baseline method traverses the set enumeration tree [21] of subspaces in a depth-first manner. A set enumeration tree takes a total order on a set, the set of dimensions in our problem, and enumerates all possible subsets in the lexicographical order. Figure 1 shows a set enumeration tree that enumerates all subspaces of $D = \{D_1, D_2, D_3, D_4\}$.

Using Eqs. 3 and 4, the baseline algorithm, shown in Algorithm 1, computes the likelihood contrast for every subspace where $L_S(q \mid O_+) \geq \delta$, and returns the top-$k$ subspaces. The time complexity is $\mathcal{O}(2^{|D|} \cdot (|O_+| + |O_-|))$.

4.2 The framework of CSMiner

$L_S(q \mid O_+)$ is not monotonic in subspaces. To prune subspaces using the minimum likelihood threshold δ, we develop an upper bound of $L_S(q \mid O_+)$. We sort all the dimensions in their standard deviation descending order. Let $\mathcal{S}$ be the set of descendants of S in the subspace set enumeration tree using the standard deviation descending order. Define

$$L_S^*(q \mid O_+) = \frac{1}{|O_+|(\sqrt{2\pi}\sigma'_{min}h'_{opt\_min})^\tau} \sum_{o \in O_+} \exp\left(\frac{-dist_S(q,o)^2}{2(\sigma_S h'_{opt\_max})^2}\right) \tag{10}$$

where $\sigma'_{min} = \min\{\sigma_{S'} \mid S' \in \mathcal{S}\}$, $h'_{opt\_min} = \min\{h_{S'\_opt} \mid S' \in \mathcal{S}\}$, $h'_{opt\_max} = \max\{h_{S'\_opt} \mid S' \in \mathcal{S}\}$, and

$$\tau = \begin{cases} |S| & \text{if } \sqrt{2\pi}\sigma'_{min}h'_{opt\_min} \geq 1 \\ \max\{|S'| \mid S' \in \mathcal{S}\} & \text{if } \sqrt{2\pi}\sigma'_{min}h'_{opt\_min} < 1 \end{cases}$$

We have the following result.

Theorem 3 (Monotonic density bound) *For a query object q, a set of objects O, and subspaces S_1, S_2 such that S_1 is an ancestor of S_2 in the subspace set enumeration tree in which dimensions in full space D are sorted by their standard deviation descending order, it is true that $L^*_{S_1}(q \mid O) \geq L_{S_2}(q \mid O)$.*

Proof Let $\mathcal{S}$ be the set of descendants of S_1 in the subspace set enumeration tree using the standard deviation descending order in O. We define $\sigma'_{min} = \min\{\sigma_{S'} \mid S' \in \mathcal{S}\}$, $h'_{opt\_min} = \min\{h_{S'\_opt} \mid S' \in \mathcal{S}\}$, $h'_{opt\_max} = \max\{h_{S'\_opt} \mid S' \in \mathcal{S}\}$, and

$$\tau = \begin{cases} |S_1| & \text{if } \sqrt{2\pi}\sigma'_{min}h'_{opt\_min} \geq 1 \\ \max\{|S'| \mid S' \in \mathcal{S}\} & \text{if } \sqrt{2\pi}\sigma'_{min}h'_{opt\_min} < 1 \end{cases}$$

(Note that the computing of σ'_{min}, h'_{opt_min}, and h'_{opt_max} has linear complexity. As introduced in Sect. 3.2, $\sigma_{S'}$ is the root of the average marginal variance in S' and $h_{S'\_opt}$ depends on the values of $|O|$ and $|S'|$. Let $S'' \in \mathcal{S}$ such that for any subspace $S' \in \mathcal{S}$, $S' \subseteq S''$. Recall that the dimensions in the set enumeration tree are sorted by the standard deviation descending order, and then σ'_{min} can be obtained by checking dimensions in $S''\backslash S_1$ one by one in the standard deviation ascending order. Moreover, h'_{opt_min} (h'_{opt_max}) can be obtained by comparing $h_{S'\_opt}$ with different values of $|S'| \in [|S_1|+1, |S''|]$.) As $S_2 \in \mathcal{S}$, we have $1 \leq |S_1| < |S_2| \leq \max\{|S'| \mid S' \in \mathcal{S}\}$, and $\sigma_{S_1} \geq \sigma_{S_2} \geq \sigma'_{min}$. Then, $\sigma_{S_2}h_{S_2\_opt} \geq \sigma'_{min}h'_{opt\_min}$. Thus,

$$(\sqrt{2\pi}\sigma_{S_2}h_{S_2\_opt})^{|S_2|} > (\sqrt{2\pi}\sigma'_{min}h'_{opt\_min})^{\tau}$$

Moreover, for $o \in O$, $dist_{S_1}(q,o) \leq dist_{S_2}(q,o)$. Correspondingly,

$$\frac{-dist_{S_2}(q,o)^2}{2(\sigma_{S_2}h_{S_2\_opt})^2} \leq \frac{-dist_{S_1}(q,o)^2}{2(\sigma_{S_1}h'_{opt\_max})^2}$$

By Eq. 3,

$$\begin{aligned} L_{S_2}(q \mid O) &= \frac{1}{|O|(\sqrt{2\pi}\sigma_{S_2}h_{S_2\_opt})^{|S_2|}} \sum_{o\in O} \exp\left(\frac{-dist_{S_2}(q,o)^2}{2(\sigma_{S_2}h_{S_2\_opt})^2}\right) \\ &\leq \frac{1}{|O|(\sqrt{2\pi}\sigma'_{min}h'_{opt\_min})^{\tau}} \sum_{o\in O} \exp\left(\frac{-dist_{S_1}(q,o)^2}{2(\sigma_{S_1}h'_{opt\_max})^2}\right) \\ &= L^*_{S_1}(q \mid O) \end{aligned}$$

□

Using Theorem 3, in addition to $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$, we also compute $L^*_S(q \mid O_+)$ for each subspace S. We are now in a position to state a pruning rule based on this theorem.

Pruning Rule 1 *Given a minimum likelihood threshold δ, if $L^*_S(q \mid O_+) < \delta$ in a subspace S, all superspaces of S can be pruned.*

Note that by using depth-first search, the distance between two objects in a super-space can be computed incrementally from the distance among the objects in a subspace. Given two objects q and o, let subspace $S' = S \cup \{D_i\}$. We have $dist_{S'}(q,o)^2 = dist_S(q,o)^2 + (q.D_i - o.D_i)^2$.

Algorithm 2 shows the pseudo-code of the framework of CSMiner. Similar to the baseline method (Algorithm 1), CSMiner conducts a depth-first search on the subspace set enumeration

Algorithm 2 CSMiner(q, O_+, O_-, δ, k)

Input: q: query object, O_+: objects belonging to C_+, O_-: objects belonging to C_-, δ: likelihood threshold, k: positive integer
Output: k subspaces with the highest likelihood contrast
1: let Ans be the current top-k list of subspaces, initialize Ans as k null subspaces associated with likelihood contrast 0
2: traverse the subspace set enumeration tree in a depth-first search manner
3: **for** each subspace S **do**
4: compute σ_{S+}, σ_{S-}, σ'_{min}, h_{opt}, h'_{opt_min}, and h'_{opt_max};
5: compute $L^*_S(q \mid O_+)$ using Equation 10;
6: **if** $L^*_S(q \mid O_+) < \delta$ **then**
7: prune all descendants of S and go to Step 2; // Pruning Rule 1
8: **else**
9: compute $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$ using Equation 3;
10: **if** $L_S(q \mid O_+) \geq \delta$ and $\exists S' \in Ans$ s.t. $\frac{L_S(q|O_+)}{L_S(q|O_-)} > LC_{S'}(q)$ **then**
11: insert S into Ans and remove S' from Ans;
12: **end if**
13: **end if**
14: **end for**
15: **return** Ans;

tree. For a candidate subspace S, CSMiner calculates $L^*_S(q \mid O_+)$ using Eq. 10. If $L^*_S(q \mid O_+)$ is less than the minimum likelihood threshold, all superspaces of S can be pruned by Theorem 3. Due to the hardness of the problem shown in Sect. 3.3 and the heuristic nature of this method, the time complexity of CSMiner is $\mathcal{O}(2^{|D|} \cdot (|O_+| + |O_-|))$, the same as the exhaustive baseline method. However, as shown by our empirical study, CSMiner is substantially faster than the baseline method.

As stated in Algorithm 2, CSMiner starts with reading q, O_+ and O_-. For a candidate subspace S, CSMiner stores σ_{S+}, σ_{S-}, σ'_{min}, h_{opt}, h'_{opt_min}, and h'_{opt_max} to compute $L^*_S(q \mid O_+)$, and $LC_S(q)$. As CSMiner traverses the subspace set enumeration tree in a depth-first manner and finds top-k subspaces with the highest likelihood contrast, CSMiner only stores the likelihood contrast information of k candidate subspaces. The space complexity of CSMiner is $\mathcal{O}(|O_+| + |O_-| + k)$. Observe that $k \leq 2^{|D|}$ (D representing the full space).

4.3 A bounding-pruning-refining method

For a query object q and a set of objects O, the likelihood $L_S(q \mid O)$, computed by Eq. 3, is the sum of density contributions of objects in O to q in subspace S. In Gaussian kernel estimation, given object $o \in O$, the contribution from o to $L_S(q \mid O)$ is $\frac{1}{|O|(\sqrt{2\pi}h_S)^{|S|}} \exp\left(\frac{-dist_S(q,o)^2}{2{h_S}^2}\right)$. We observe that the contribution of o decays exponentially as the distance between q and o increases, and $L_S(q \mid O)$ can be bounded.

For a query object q and a set of objects O, the *ϵ-neighborhood* ($\epsilon > 0$) of q in subspace S is $N^\epsilon_S(q \mid O) = \{o \in O \mid dist_S(q, o) \leq \epsilon\}$. We can divide $L_S(q \mid O)$ into two parts, that is, $L_S(q \mid O) = L_{N^\epsilon_S}(q \mid O) + L^{rest}_S(q \mid O)$. The first part is contributed by the objects in the ϵ-neighborhood, that is,

$$L_{N^\epsilon_S}(q \mid O) = \frac{1}{|O|(\sqrt{2\pi}h_S)^{|S|}} \sum_{o \in N^\epsilon_S(q|O)} \exp\left(\frac{-dist_S(q,o)^2}{2{h_S}^2}\right),$$

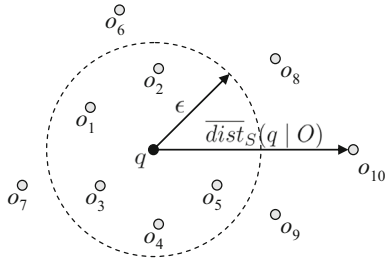


Fig. 2 An example of an ϵ-neighborhood in a 2-dimensional subspace (within the *dashed circle*)

and the second part is by the objects outside the ϵ-neighborhood, that is,

$$L_S^{rest}(q \mid O) = \frac{1}{|O|(\sqrt{2\pi}h_S)^{|S|}} \sum_{o \in O \setminus N_S^{\epsilon}(q|O)} \exp\left(\frac{-dist_S(q,o)^2}{2{h_S}^2}\right).$$

Let $\overline{dist}_S(q \mid O)$ be the maximum distance between q and all objects in O in subspace S. We have,

$$\exp\left(\frac{-\overline{dist}_S(q \mid O)^2}{2{h_S}^2}\right) \leq \frac{|O|(\sqrt{2\pi}h_S)^{|S|}}{|O \setminus N_S^{\epsilon}(q \mid O)|} L_S^{rest}(q \mid O) \leq \exp\left(\frac{-\epsilon^2}{2{h_S}^2}\right)$$

Example 1 Figure 2 illustrates an example of a ϵ-neighborhood of object q with respect to object set O in a 2-dimensional subspace S. From Fig. 2, we can see that $N_S^{\epsilon}(q \mid O) = \{o_1, o_2, o_3, o_4, o_5\}$, and $\overline{dist}_S(q \mid O) = dist_S(q, o_{10})$.

Using the above, an upper bound of $L_S^*(q \mid O_+)$ using ϵ-neighborhood ($N_S^{\epsilon}(q \mid O_+) = \{o \in O_+ \mid dist_S(q,o) \leq \epsilon\}$), denoted by $L_S^{*\epsilon}(q \mid O_+)$, is

$$L_S^{*\epsilon}(q \mid O_+) = \frac{\sum_{o \in N_S^{\epsilon}(q|O_+)} \exp\left(\frac{-dist_S(q,o)^2}{2(\sigma_S h'_{opt\_max})^2}\right) + |O_+ \setminus N_S^{\epsilon}(q \mid O_+)| \exp\left(\frac{-\epsilon^2}{2(\sigma_S h'_{opt\_max})^2}\right)}{|O_+|(\sqrt{2\pi}\sigma'_{min} h'_{opt\_min})^{\tau}} \quad (11)$$

where the meanings of σ'_{min}, h'_{opt_min}, h'_{opt_max}, and τ are the same as those in Eq. 10.

Pruning Rule 2 *Given a minimum likelihood threshold δ, if $L_S^{*\epsilon}(q \mid O_+) < \delta$ in a subspace S, all superspaces of S can be pruned.*

Moreover, using the ϵ-neighborhood, we have the following upper and lower bounds of $L_S(q \mid O)$.

Theorem 4 (Bounds) *For a query object q, a set of objects O and $\epsilon \geq 0$,*

$$LL_S^{\epsilon}(q \mid O) \leq L_S(q \mid O) \leq UL_S^{\epsilon}(q \mid O)$$

where

$$LL_S^{\epsilon}(q \mid O) = \frac{\sum_{o \in N_S^{\epsilon}(q|O)} \exp\left(\frac{-dist_S(q,o)^2}{2{h_S}^2}\right) + |O \setminus N_S^{\epsilon}(q \mid O)| \exp\left(\frac{-\overline{dist}_S(q|O)^2}{2{h_S}^2}\right)}{|O|(\sqrt{2\pi}h_S)^{|S|}}$$

and

$$UL_S^{\epsilon}(q \mid O) = \frac{\sum_{o \in N_S^{\epsilon}(q|O)} \exp\left(\frac{-dist_S(q,o)^2}{2h_S{}^2}\right) + |O \backslash N_S^{\epsilon}(q \mid O)| \exp\left(\frac{-\epsilon^2}{2h_S{}^2}\right)}{|O|(\sqrt{2\pi}h_S)^{|S|}}$$

Proof For any object $o \in O \backslash N_S^{\epsilon}(q \mid O)$, $\epsilon^2 \leq dist_S(q,o)^2 \leq \overline{dist}_S(q \mid O)^2$. Then,

$$\exp\left(\frac{-\epsilon^2}{2h_S{}^2}\right) \geq \exp\left(\frac{-dist_S(q,o)^2}{2h_S{}^2}\right) \geq \exp\left(\frac{-\overline{dist}_S(q \mid O)^2}{2h_S{}^2}\right)$$

Thus,

$$|O \backslash N_S^{\epsilon}(q \mid O)| e^{\frac{-\epsilon^2}{2h_S{}^2}} \geq |O \backslash N_S^{\epsilon}(q \mid O)| e^{\frac{-dist_S(q,o)^2}{2h_S{}^2}} \geq |O \backslash N_S^{\epsilon}(q \mid O)| e^{\frac{-\overline{dist}_S(q|O)^2}{2h_S{}^2}}$$

Correspondingly,

$$LL_S^{\epsilon}(q \mid O) \leq L_S(q \mid O) \leq UL_S^{\epsilon}(q \mid O)$$

□

We obtain an upper bound of $LC_S(q)$ based on Theorem 4 and Eq. 4.

Corollary 1 (Likelihood contrast upper bound) *For a query object q, a set of objects O_+, a set of objects O_-, and $\epsilon \geq 0$, $LC_S(q) \leq \frac{UL_S^{\epsilon}(q|O_+)}{LL_S^{\epsilon}(q|O_-)}$.*

Proof By Theorem 4, we have $L_S(q \mid O_+) \leq UL_S^{\epsilon}(q \mid O_+)$ and $L_S(q \mid O_-) \geq LL_S^{\epsilon}(q \mid O_-)$. Then,

$$LC_S(q) = \frac{L_S(q \mid O_+)}{L_S(q \mid O_-)} \leq \frac{UL_S^{\epsilon}(q \mid O_+)}{L_S(q \mid O_-)} \leq \frac{UL_S^{\epsilon}(q \mid O_+)}{LL_S^{\epsilon}(q \mid O_-)}$$

Using Corollary 1, we have the following.

Pruning Rule 3 *For a subspace S, if there are at least k subspaces whose likelihood contrasts are greater than $\frac{UL_S^{\epsilon}(q|O_+)}{LL_S^{\epsilon}(q|O_-)}$, then S cannot be a top-k subspace of the largest likelihood contrast.*

We implement the bounding-pruning-refining method in CSMiner to compute bounds of likelihood and contrast ratio. We call this version CSMiner-BPR. For a candidate subspace S, CSMiner-BPR calculates $UL_S^{\epsilon}(q \mid O_+)$, $LL_S^{\epsilon}(q \mid O_-)$, and $L_S^{*\epsilon}(q \mid O_+)$ using the ϵ-neighborhood. If $UL_S^{\epsilon}(q \mid O_+)$ is less than the minimum likelihood threshold (δ), CSMiner-BPR checks whether it is true that $L_S^{*\epsilon}(q \mid O_+) < \delta$ (Pruning Rule 2) or $L_S^{*}(q \mid O_+) < \delta$ (Pruning Rule 1). Otherwise, CSMiner-BPR checks whether the likelihood contrasts of the current top-k subspaces are larger than the upper bound of $LC_S(q)$ $\left(= \frac{UL_S^{\epsilon}(q|O_+)}{LL_S^{\epsilon}(q|O_-)}\right)$. If not, CSMiner-BPR refines $L_S^{*}(q \mid O_+)$, $L_S(q \mid O_+)$, and $L_S(q \mid O_-)$ by involving objects that are out of the ϵ-neighborhood. S will be added into the current top-k list if $L_S^{*}(q \mid O_+) \geq \delta$ and the ratio of $L_S(q \mid O_+)$ to $L_S(q \mid O_-)$ is larger than one of the current top-k ones. Note that the computational cost for $L_S^{*}(q \mid O_+)$ can be high, especially, when the size of O_+ is large. Thus for efficiency, we employ a tradeoff between Pruning Rule 1 and Pruning Rule 3. Specifically, when we are searching a subspace S, once we can determine that S cannot be a top-k contrast subspace, then we terminate the search of S immediately. Therefore, CSMiner-BPR accelerates CSMiner by avoiding the cost for computing the likelihood contributions

of objects outside the ϵ-neighborhood to q when $L_S^{*\epsilon}(q \mid O_+) < \delta$ (Pruning Rule 2) or there are at least k subspaces whose likelihood contrasts are greater than $\frac{UL_S^{\epsilon}(q|O_+)}{LL_S^{\epsilon}(q|O_-)}$ (Pruning Rule 3).

Computing ϵ-neighborhood is critical in CSMiner-BPR. The distance between objects increases when dimensionality increases. Thus, the value of ϵ should not be fixed. The standard deviation expresses the variability of a set of data. For subspace S, we set $\epsilon = \sqrt{\alpha \cdot \sum_{D_i \in S}(\sigma^2_{D_i{}^+} + \sigma^2_{D_i{}^-})}$ ($\alpha \geq 0$), where $\sigma^2_{D_i{}^+}$ and $\sigma^2_{D_i{}^-}$ are the marginal variances of O_+ and O_-, respectively, on dimension D_i ($D_i \in S$), and α is a system defined parameter. Our experiments show that α can be set in the range of 0.8–1.2 and is not sensitive. Algorithm 3 provides the pseudo-code of CSMiner-BPR. Theorem 5 guarantees that no matter how the neighborhood distance (ϵ) is varied, and the mining result of CSMiner-BPR is unchanged.

Theorem 5 *Given data set O, query object q, minimum likelihood threshold δ and parameter k, for any neighborhood distances ϵ_1 and ϵ_2, $CS^{\epsilon_1}(q \mid O) = CS^{\epsilon_2}(q \mid O)$, where $CS^{\epsilon_1}(q \mid O)$ ($CS^{\epsilon_2}(q \mid O)$) is the set of contrast subspaces discovered by CSMiner-BPR using ϵ_1 (ϵ_2).*

Proof We prove by contradiction.

Assume that subspace $S \in CS^{\epsilon_1}(q \mid O)$ but $S \notin CS^{\epsilon_2}(q \mid O)$. As $S \in CS^{\epsilon_1}(q \mid O)$, we have $(\star) L_S(q \mid O_+) \geq \delta$. On the other hand, $S' \notin CS^{\epsilon_2}(q \mid O)$ means that (i) $L_S^{*\epsilon_2}(q \mid O_+) < \delta$, or (ii) $\exists S' \in CS^{\epsilon_2}(q \mid O)$ such that $S' \notin CS^{\epsilon_1}(q \mid O)$ and $\frac{UL_S^{\epsilon_1}(q|O_+)}{LL_S^{\epsilon_1}(q|O_-)} < LC_{S'}(q)$. For case (i), as $L_S(q \mid O_+) \leq L_S^*(q \mid O_+) \leq L_S^{*\epsilon_2}(q \mid O_+)$, we have $L_S(q \mid O_+) < \delta$, contradicting $(\star)$. For case (ii), as $LC_S(q) \leq \frac{UL_S^{\epsilon_1}(q|O_+)}{LL_S^{\epsilon_1}(q|O_-)}$, we have $LC_S(q) < LC_{S'}(q)$, contradicting $S' \notin CS^{\epsilon_1}(q \mid O)$.

Corollary 2 *Given data set O, query object q, minimum likelihood threshold δ, and parameter k, the mining result of CSMiner-BPR, no matter what the value of parameter α is, the output is the same as that of CSMiner.*

Proof For subspace S, suppose ϵ, computed by parameter α, is greater than $\overline{dist}_S(q \mid O)$. We have $N_S^{\epsilon}(q \mid O) = O$. Correspondingly, $UL_S^{\epsilon}(q \mid O_+) = L_S(q \mid O_+)$, $LL_S^{\epsilon}(q \mid O_-) = L_S(q \mid O_-)$, and $L_S^{*\epsilon}(q \mid O_+) = L_S^*(q \mid O_+)$. Then the execution flow of CSMiner-BPR (Algorithm 3) is the same as that of CSMiner (Algorithm 2). Furthermore, by Theorem 5, the mining result of CSMiner-BPR is unchanged no matter what the value of neighborhood distance is.

5 Empirical evaluation

In this section, we report a systematic empirical study using real data sets to verify the effectiveness and efficiency of CSMiner (CSMiner-BPR). In general, we study how sensitive are our methods to the running parameters, such as δ, k, and α, in terms of discovered contrast subspaces and running time; and how sensitive are our methods to different bandwidth values and kernel function, in terms of the similarity of mining results. All experiments were conducted on a PC computer with an Intel Core i7-3770 3.40 GHz CPU, and 8 GB main memory, running Windows 7 operating system. All algorithms were implemented in Java and compiled by JDK 7. We set $\delta = 0.001$, $k = 10$, and $\alpha = 0.8$ as defaults in our experiments.

Algorithm 3 CSMiner-BPR(q, O_+, O_-, δ, k, α)

Input: q: a query object, O_+: the set of objects belonging to C_+, O_-: the set of objects belonging to C_-, δ: a likelihood threshold, k: positive integer, α: neighborhood parameter
Output: k subspaces with the highest likelihood contrast

let Ans be the current top-k list of subspaces, initialize Ans as k null subspaces associated with likelihood contrast 0
for each subspace S in the subspace set enumeration tree, searched in the depth-first manner **do**
  compute ϵ, σ_{S+}, σ_{S-}, σ'_{min}, h_{opt}, h'_{opt_min}, and h'_{opt_max};
  $N_S^\epsilon(q \mid O_+) \leftarrow \emptyset$; $N_S^\epsilon(q \mid O_-) \leftarrow \emptyset$; $\overline{dist}_S(q \mid O_-) \leftarrow 0$;
  for each object $o \in O_+ \cup O_-$ **do**
    $dist_S(q,o)^2 \leftarrow dist_{S^p}(q,o)^2 + (q.D' - o.D')^2$; // $S^p (= S \backslash \{D'\})$ is the parent of S.
    if $o \in O_+$ and $dist_S(q,o) < \epsilon$ **then**
      $N_S^\epsilon(q \mid O_+) \leftarrow N_S^\epsilon(q \mid O_+) \cup \{o\}$;
    end if
    if $o \in O_-$ **then**
      if $dist_S(q,o) < \epsilon$ **then**
        $N_S^\epsilon(q \mid O_-) \leftarrow N_S^\epsilon(q \mid O_-) \cup \{o\}$;
      end if
      if $\overline{dist}_S(q \mid O_-) < dist_S(q,o)$ **then**
        $\overline{dist}_S(q \mid O_-) \leftarrow dist_S(q,o)$;
      end if
    end if
  end for
  compute $UL_S^\epsilon(q \mid O_+)$, $LL_S^\epsilon(q \mid O_-)$ and $L_S^{*\epsilon}(q \mid O_+)$; // bounding
  if $UL_S^\epsilon(q \mid O_+) < \delta$ **then**
    if $L_S^{*\epsilon}(q \mid O_+) < \delta$ **then**
      prune all descendants of S and go to Step 2; // Pruning Rule 2
    end if
    compute $L_S^*(q \mid O_+)$;
    if $L_S^*(q \mid O_+) < \delta$ **then**
      prune all descendants of S and go to Step 2; // Pruning Rule 1
    end if
  else
    if $\exists S' \in Ans$ s.t. $\frac{UL_S^\epsilon(q|O_+)}{LL_S^\epsilon(q|O_-)} \geq LC_{S'}(q)$ **then**
      compute $L_S^*(q \mid O_+)$ using Equation 10; // refining
      if $L_S^*(q \mid O_+) < \delta$ **then**
        prune all descendants of S and go to Step 2; // Pruning Rule 1
      else
        compute $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$ using Equation 3; // refining
        if $L_S(q \mid O_+) \geq \delta$ and $\exists S' \in Ans$ s.t. $\frac{L_S(q|O_+)}{L_S(q|O_-)} > LC_{S'}(q)$ **then**
          insert S into Ans and remove S' from Ans;
        end if
      end if
    end if
  end if
end for
return Ans;

5.1 Effectiveness

We use 6 real data sets from the UCI machine learning repository [2]. We remove non-numerical attributes and all instances containing missing values. Table 2 shows the data characteristics.

As shown in Table 2, BCW, Glass, PID, and Wine are typical small data sets that contain hundreds of objects with around 10 numerical attributes. The objects in BCW, Glass, and PID are divided into 2 classes, respectively, while the objects in Wine are divided into 3 classes. Compared with BCW, Glass, PID, and Wine, CMSC and Waveform contain more numerical attributes. We note that CMSC is an unbalanced data set, in which the number of objects in the

two classes are 46 and 494, respectively. Among all selected data sets, Waveform containing 5000 objects is the largest one with the highest dimensionality.

For each data set, we take each record as a query object q, and all records except q belonging to the same class as q forming the set O_1, and records belonging to the other classes forming the set O_2. Using CSMiner, we compute for each record (1) the *inlying contrast subspace* taking O_1 as O_+ and O_2 as O_-, and (2) the *outlying contrast subspace* taking O_2 as O_+ and O_1 as O_-. In this experiment, we only compute the top-1 subspace. For clarity, we denote the likelihood contrasts of inlying contrast subspace by $LC_S^{in}(q)$ and those of outlying contrast subspace by $LC_S^{out}(q)$. The minimum likelihood threshold (δ) is set to 0.001.

Tables 3, 4, 5, 6, 7, and 8 list the joint distributions of $LC_S^{in}(q)$ and $LC_S^{out}(q)$ in each data set. Consider that the query object has the same class label as objects in O_1 in the original data set. Thus, it is expected that, for most objects, $LC_S^{in}(q)$ are larger than $LC_S^{out}(q)$. However, interestingly a good portion of objects have strong outlying contrast subspaces. For example, in CMSC, more than 40 % of the objects have outlying contrast subspaces satisfying $LC_S^{out}(q) \geq 10^3$. Moreover, we can see that, except PID, a non-trivial number of objects in each data set have both strong inlying and outlying contrast subspaces (e.g., $LC_S^{in}(q) \geq 10^4$ and $LC_S^{out}(q) \geq 10^2$).

Figures 3 and 4 show the distributions of dimensionality of top-1 inlying and outlying contrast subspaces with different minimum likelihood thresholds (δ), respectively. The dimensionality distribution is an interesting feature characterizing a data set. For example, in most cases the contrast subspaces tend to have low dimensionality. However, in CMSC and Wine, the inlying contrast subspaces tend to have high dimensionality. Moreover, we can see

Table 2 Data set characteristics

Data set	# Objects	# Attributes	# Classes
Breast cancer Wisconsin (BCW)	683	9	2
Climate model simulation crashes (CMSC)	540	18	2
Glass identification (Glass)	214	9	2
Pima Indians diabetes (PID)	768	8	2
Waveform	5000	21	3
Wine	178	13	3

Table 3 Distribution of $LC_S(q)$ in BCW ($\delta = 0.001, k = 1$)

$LC_S^{in}(q)$	$LC_S^{out}(q)$					Total
	<1	[1,3)	[3,10)	[10, 10^2)	$\geq 10^2$	
$<10^4$	0	3	0	7	23	33
[10^4, 10^5)	7	4	2	4	7	24
[10^5, 10^6)	21	21	5	8	9	64
[10^6, 10^7)	184	33	5	4	9	235
$\geq 10^7$	121	31	74	66	35	327
Total	333	92	86	89	83	683

Table 4 Distribution of $LC_S(q)$ in CMSC ($\delta = 0.001, k = 1$)

$LC_S^{in}(q)$	$LC_S^{out}(q)$					Total
	$[10, 10^2)$	$[10^2, 10^3)$	$[10^3, 10^4)$	$[10^4, 10^5)$	$\geq 10^5$	
$<10^3$	1	11	12	2	0	26
$[10^3, 10^4)$	6	35	47	6	6	100
$[10^4, 10^5)$	10	46	44	8	2	110
$[10^5, 10^6)$	11	40	32	8	2	93
$\geq 10^6$	39	110	50	11	1	211
Total	67	242	185	35	11	540

Table 5 Distribution of $LC_S(q)$ in Glass ($\delta = 0.001, k = 1$)

$LC_S^{in}(q)$	$LC_S^{out}(q)$					Total
	<1	[1,3)	[3,10)	$[10, 10^2)$	$\geq 10^2$	
$<10^2$	0	0	0	1	7	8
$[10^2, 10^3)$	2	8	4	4	7	25
$[10^3, 10^4)$	28	91	6	4	5	134
$[10^4, 10^5)$	1	4	0	0	3	8
$\geq 10^5$	0	1	0	30	8	39
Total	31	104	10	39	30	214

Table 6 Distribution of $LC_S(q)$ in PID ($\delta = 0.001, k = 1$)

$LC_S^{in}(q)$	$LC_S^{out}(q)$					Total
	<1	[1,3)	[3,10)	[10, 30)	≥ 30	
<1	0	0	1	0	0	1
[1, 3)	2	241	62	8	2	315
[3, 10)	36	328	31	3	0	398
[10, 30)	23	23	2	0	0	48
≥ 30	3	3	0	0	0	6
Total	64	595	96	11	2	768

Table 7 Distribution of $LC_S(q)$ in Waveform ($\delta = 0.001, k = 1$)

$LC_S^{in}(q)$	$LC_S^{out}(q)$					Total
	[1, 3)	[3,10)	$[10, 10^2)$	$[10^2, 10^3)$	$\geq 10^3$	
<10	0	24	34	8	2	68
$[10, 10^2)$	204	676	772	190	71	1913
$[10^2, 10^3)$	471	1049	981	228	56	2785
$[10^3, 10^4)$	53	103	67	4	4	231
$\geq 10^4$	0	2	1	0	0	3
Total	728	1854	1855	430	133	5000

Table 8 Distribution of $LC_S(q)$ in Wine ($\delta = 0.001, k = 1$)

$LC_S^{in}(q)$	$LC_S^{out}(q)$					Total
	<1	[1,3)	[3,10)	$[10, 10^2)$	$\geq 10^2$	
$<10^3$	0	13	8	7	5	33
$[10^3, 10^4)$	1	18	11	4	0	34
$[10^4, 10^5)$	2	23	12	5	2	44
$[10^5, 10^6)$	3	7	5	1	0	16
$\geq 10^6$	7	20	16	4	4	51
Total	13	81	52	21	11	178

that with the decrease of δ, the number of subspaces with higher dimensionality is typically increased.

5.2 Efficiency

To the best of our knowledge, there is no previous method tackling the exact same mining problem. Therefore, we evaluate the efficiency of CSMiner and its variations. Specifically, we implemented the baseline method (Algorithm 1). To evaluate the efficiency of our pruning techniques for contrast subspace mining, we also implemented CSMiner (Algorithm 2) and CSMiner-BPR (Algorithm 3) using the bounding-pruning-refining method.

We report the results on the Waveform data set only, since it is the largest one with the highest dimensionality. We randomly select 100 records from Waveform as query objects and report the average runtime. The results on the other data sets follow similar trends.

Figure 5 shows the runtime with respect to the minimum likelihood threshold δ. A logarithmic scale has been used for the runtime to better demonstrate the difference in the behavior between CSMiner and the baseline. The baseline performs exhaustive subspace search, and thus its runtime is unchanged across different δ values. For CSMiner and CSMiner-BPR, as δ decreases, their runtime increase exponentially. However, the heuristic pruning techniques implemented in CSMiner and CSMiner-BPR accelerate the search substantially in practice. Moreover, CSMiner-BPR is slightly more efficient than CSMiner.

Figure 6 shows the runtime with respect to the data set size, which is measured by the number of objects. Again, the runtime is plotted using the logarithmic scale. We can see that our pruning techniques can achieve a roughly linear runtime in practice. Both CSMiner and CSMiner-BPR are considerably faster than the baseline method, and CSMiner-BPR is slightly more efficient than CSMiner.

Figure 7 shows the runtime with respect to the dimensionality of the data set. The runtime is also plotted using the logarithmic scale. As dimensionality increases, more candidate subspaces are generated. Correspondingly, the runtime increases exponentially. However, our heuristic pruning techniques implemented in CSMiner and CSMiner-BPR speed up the search in practice. Moreover, CSMiner-BPR is faster than CSMiner.

As stated in Sect. 4.3, CSMiner-BPR employs a user-defined parameter α to define the ϵ-neighborhood. Table 9 lists the average runtime of CSMiner-BPR for a query object with respect to α on each real data set. The runtime of CSMiner-BPR is not sensitive to α in general.

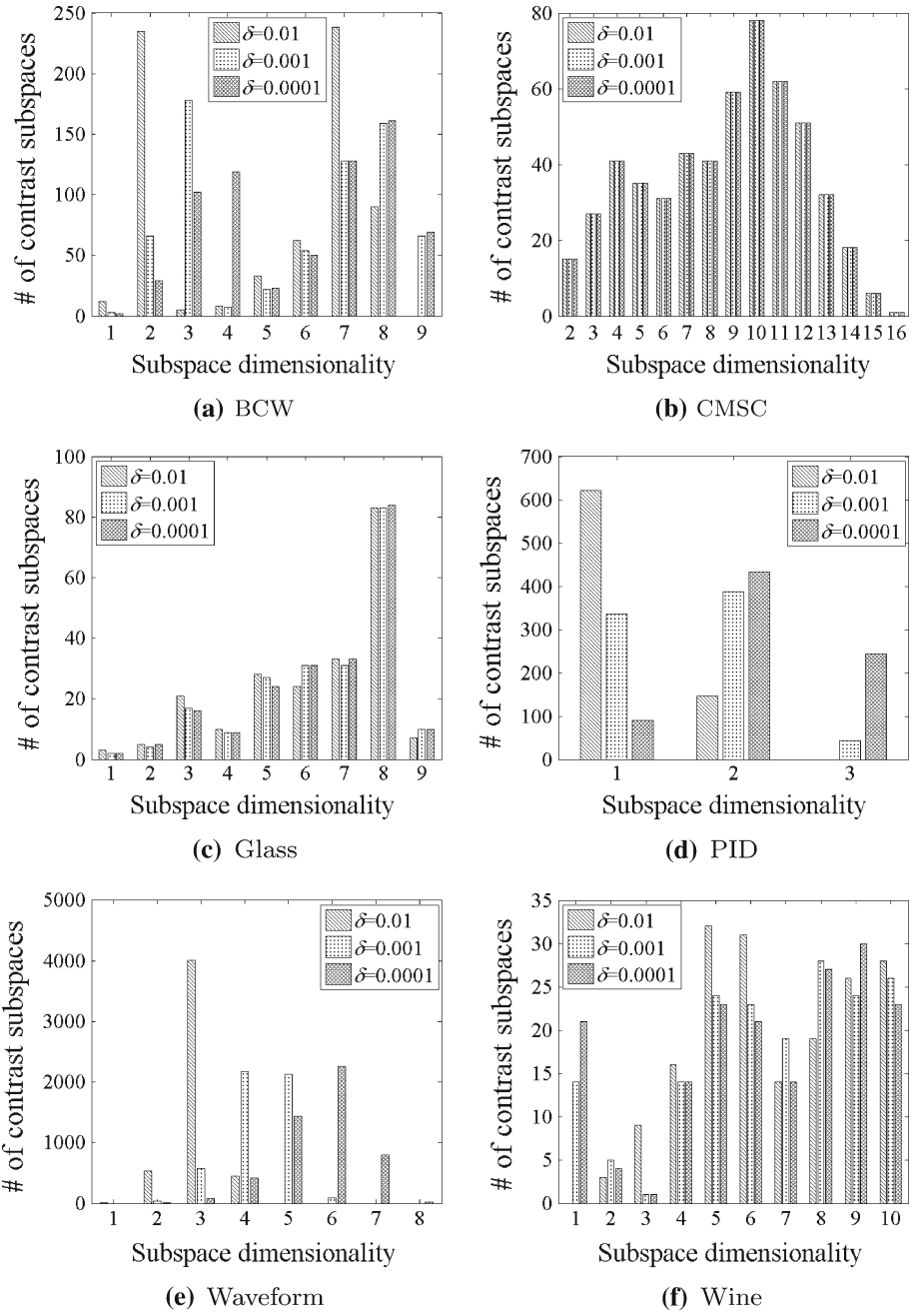


Fig. 3 Dimensionality distributions of top inlying contrast subspaces ($k = 1$)

Experimentally, the shortest runtime of CSMiner-BPR (bold values in Table 9) happens when α is in [0.6, 1.0].

Figure 8 illustrates the relative runtime of CSMiner-BPR with respect to k on each real data set, showing that CSMiner-BPR is linearly scalable with respect to k. Note that we show

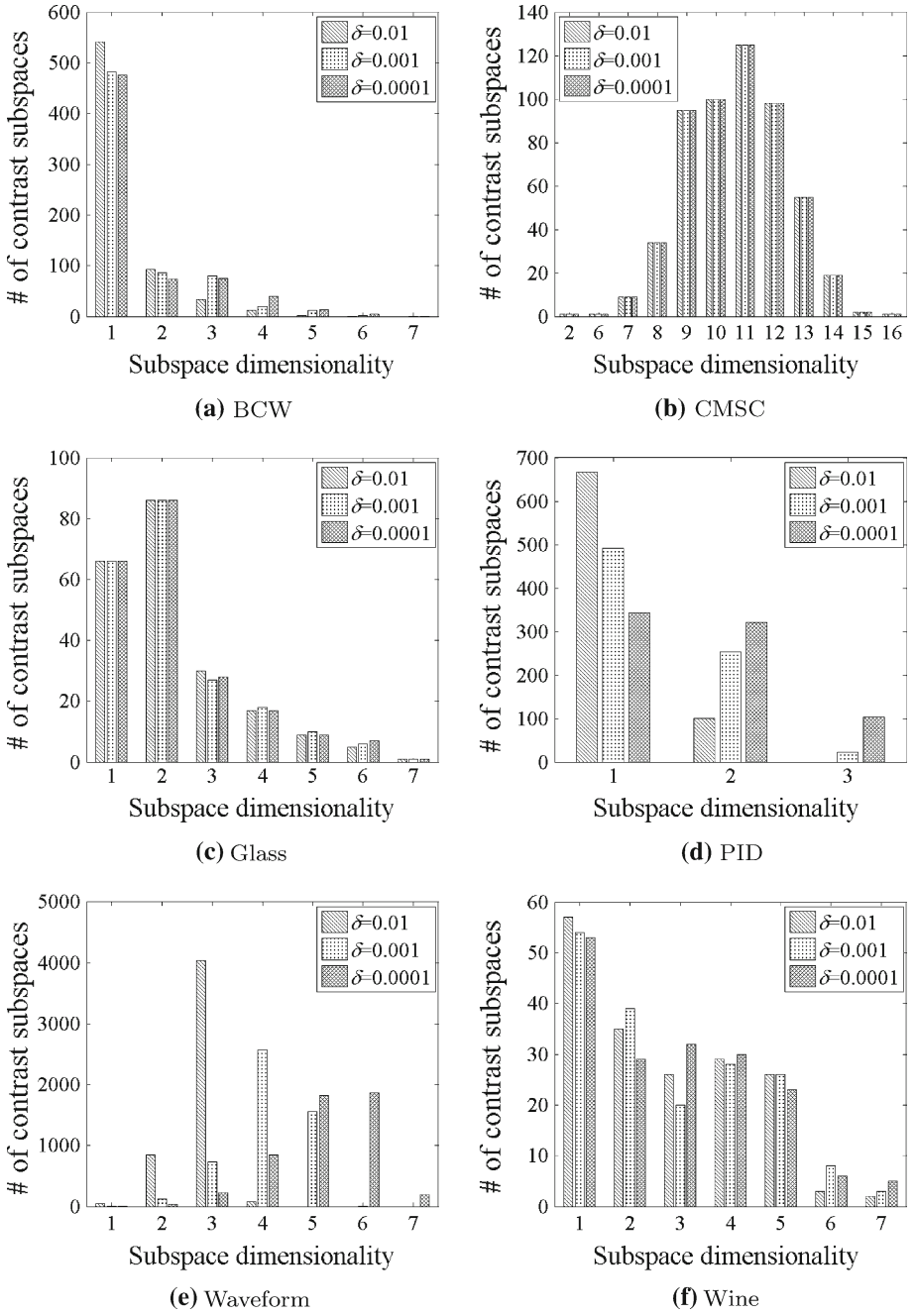


Fig. 4 Dimensionality distributions of top outlying contrast subspaces ($k = 1$)

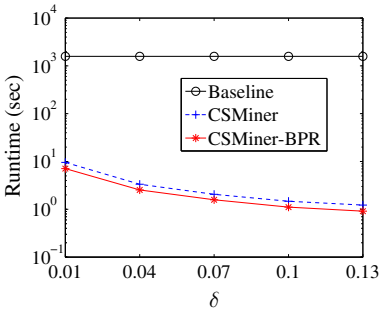


Fig. 5 Scalability test w.r.t δ ($k = 10, \alpha = 0.8$)

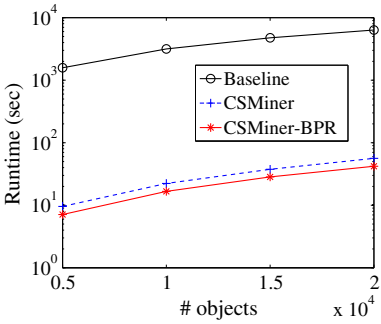


Fig. 6 Scalability test w.r.t data set size ($k = 10, \delta = 0.01, \alpha = 0.8$)

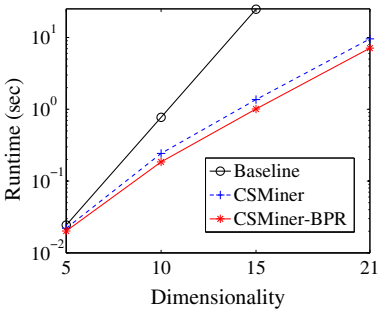


Fig. 7 Scalability test w.r.t dimensionality ($k = 10, \delta = 0.01, \alpha = 0.8$)

relative performance in Fig. 8 so that the scalability of CSMiner-BPR with respect to k on different data sets can be compared in one figure. The absolute performance of CSMiner-BPR with $k = 10$, $\delta = 0.01$ and $\alpha = 0.8$ can be found in Table 9.

5.3 Sensitivity to the bandwidth

To test the sensitivity of the top-k contrast subspaces with respect to the bandwidth value, we begin by defining the similarity measure for two lists of top-k contrast subspaces.

For any two subspaces S_1 and S_2, we measure the similarity between S_1 and S_2 by the Jaccard similarity coefficient, denoted by $Jaccard(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$.

Given a positive integer r, let $\mathbb{P}^r$ be the set of all permutations of the set $\{i \mid 1 \leq i \leq r\}$. Correspondingly, $|\mathbb{P}^r| = r!$. For permutation $P \in \mathbb{P}^r$, we denote the j-th ($1 \leq j \leq r$) element in P by $P[j]$. For example, by writing each permutation as a tuple, we have

Table 9 Average runtime of CSMiner-BPR w.r.t α ($k = 10$, $\delta = 0.01$)

Data set	Average runtime (ms)				
	$\alpha = 0.6$	$\alpha = 0.8$	$\alpha = 1.0$	$\alpha = 1.2$	$\alpha = 1.4$
BCW	20.97	20.14	17.76	16.32	**15.59**
CMSC	**11446.2**	11643.5	12915.1	14125.0	15210.2
Glass	16.13	15.83	**15.62**	15.69	15.76
PID	4.21	**4.17**	4.23	4.25	4.29
Waveform	**6807.1**	7102.3	7506.7	7874.7	8183.7
Wine	18.51	**18.16**	18.42	18.69	19.12

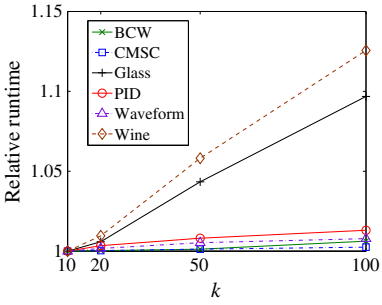


Fig. 8 Relative runtime of CSMiner-BPR w.r.t $k(\delta = 0.01, \alpha = 0.8)$

$\mathbb{P}^3 = \{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}$. Suppose $P = (2, 3, 1)$, then $P[2] = 3$.

To the best of our knowledge, there is no previous work on measuring the similarity between two ranked lists of subspaces. Given two ranked lists of top-k contrast subspaces ℓ_1 and ℓ_2, without loss of generality, we follow the definition of *average overlap* [24] (also named as *average accuracy* [26], or *intersection metric* [12]), which derives the similarity measure by averaging the overlaps of two ranked lists at each rank, to measure the similarity between ℓ_1 and ℓ_2. In addition, in consideration of the fact that each subspace in a list is a set of dimensions, we introduce the Jaccard similarity coefficient into the overlap calculation. Specifically, let $\ell_1[i]$ be the element (subspace) at rank i ($1 \leq i \leq k$) in list ℓ_1. The *agreement* of lists ℓ_1 and ℓ_2 at rank r ($1 \leq r \leq k$), $Agr(\ell_1, \ell_2, r)$, is

$$Agr(\ell_1, \ell_2, r) = \frac{1}{r} \max \left\{ \sum_{i=1}^{r} Jaccard(\ell_1[P_1[i]], \ell_2[P_2[i]]) \mid P_1, P_2 \in \mathbb{P}^r \right\}$$

Then, the similarity between ℓ_1 and ℓ_2, denoted by $Sim(\ell_1, \ell_2)$, is

$$Sim(\ell_1, \ell_2) = \frac{1}{k} \sum_{r=1}^{k} Agr(\ell_1, \ell_2, r) \tag{12}$$

Clearly, $0 \leq Sim(\ell_1, \ell_2) \leq 1$. The larger the value of $Sim(\ell_1, \ell_2)$, the more similar ℓ_1 and ℓ_2 are.

Given a set of objects O, and a query object q, to find top-k contrast subspaces for q with respect to O by CSMiner (Algorithm 2), as discussed in Sect. 3.2, we first fix the bandwidth value $h_S = \sigma_S \cdot h_{S\_opt}$ and use the Gaussian kernel function to estimate the subspace

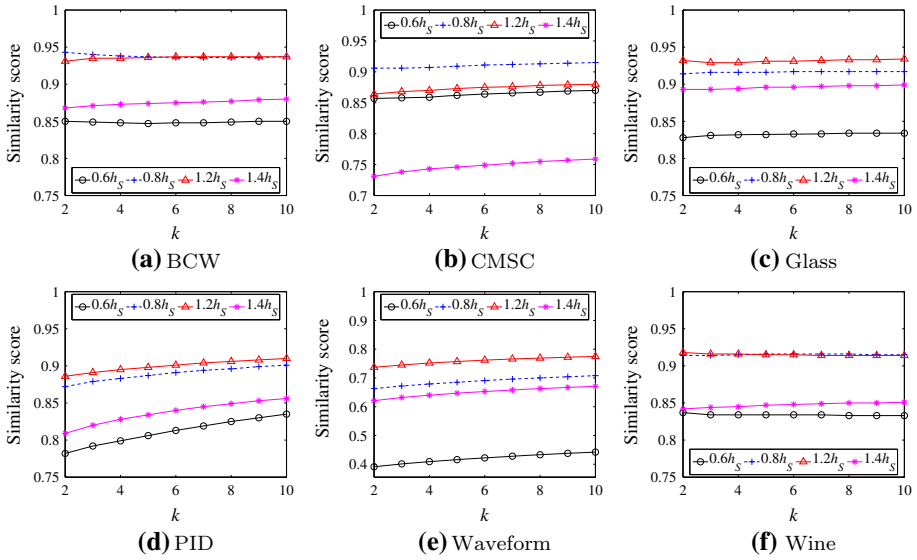


Fig. 9 The similarity scores of inlying contrast subspaces using different bandwidth values with respect to k ($\delta = 0.001$)

likelihood of q with respect to O in subspace S. We then vary the bandwidth value from $0.6h_S$ to $1.4h_S$ for density estimation in S. Let ℓ_{h_S} be the top-k contrast subspaces computed using the default bandwidth value h_S and $\ell_{\widetilde{h_S}}$ be the top-k contrast subspaces computed using other bandwidth values. For each object $q \in O$, we discover top inlying contrast subspaces and top outlying contrast subspaces of q by CSMiner using different bandwidth values. Figure 9 illustrates the average value of $Sim(\ell_{h_S}, \ell_{\widetilde{h_S}})$ of inlying contrast subspaces with respect to k, and Fig. 10 illustrates the average value of $Sim(\ell_{h_S}, \ell_{\widetilde{h_S}})$ of outlying contrast subspaces with respect to k. From the results, we can see that the contrast subspaces computed using different bandwidth values are similar in most data sets. As expected, using a bandwidth whose value is closer to h causes less difference. Moreover, we observe that with increasing k, the value of $Sim(\ell_{h_S}, \ell_{\widetilde{h_S}})$ slightly increases.

5.4 Comparison with Epanechnikov Kernel

Besides Gaussian kernel (Eq. 2), another possible kernel for multivariate kernel density estimation is the multivariate Epanechnikov kernel

$$K_e(x) = \begin{cases} \frac{1}{2} {c_d}^{-1}(d+2)(1-x^T x) & \text{if } x^T x < 1 \\ 0 & \text{otherwise} \end{cases}$$

where c_d is the volume of the unit d-dimensional sphere and can be expressed by making use of the Gamma function. It is,

$$c_d = \frac{\pi^{d/2}}{\Gamma(1+d/2)} = \begin{cases} \pi^{d/2}/(d/2)! & \text{if } d \geq 0 \text{is even} \\ \pi^{\lfloor d/2 \rfloor} 2^{\lceil d/2 \rceil}/d!! & \text{if } d \geq 0 \text{is odd} \end{cases}$$

where $d!!$ is the double factorial.

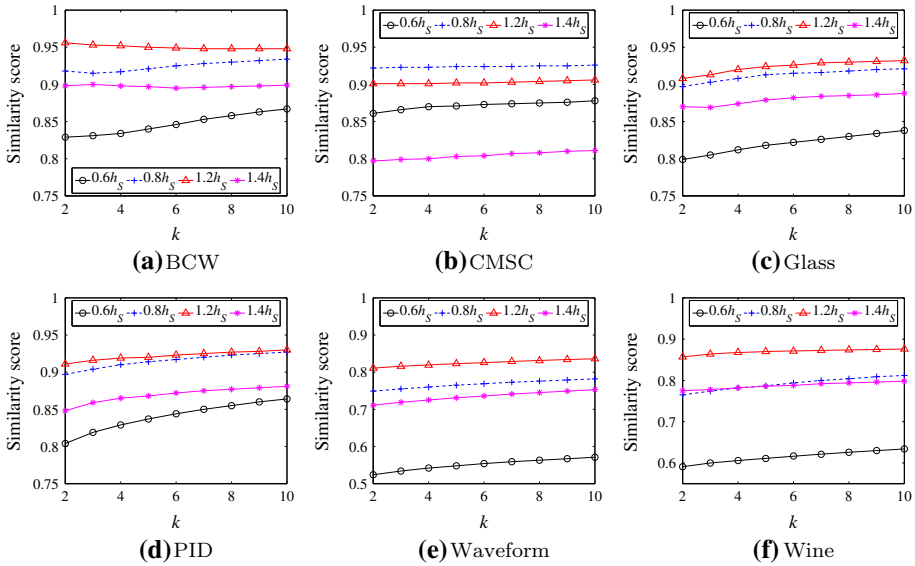


Fig. 10 The similarity scores of outlying contrast subspaces using different bandwidth values with respect to k ($\delta = 0.001$)

Plugging $K_e(x)$ into Eq. 1, the density of a query object q with respect to a set of objects O in subspace S can be estimated as

$$\hat{f}_S(q, O) = \frac{1}{|O| h_S^{|S|}} \sum_{o \in O \wedge \frac{dist_S(q,o)^2}{h_S^2} < 1} \left(\frac{1}{2} {c_{|S|}}^{-1} (|S| + 2) \left(1 - \frac{dist_S(q, o)^2}{h_S^2} \right) \right) \quad (13)$$

where h_S is the bandwidth for subspace S.

Similar to calculating the bandwidth using Gaussian kernel in Sect. 3.2, we calculate h_S as follows.

$$h_S = \sigma_S \cdot h_{S\_opt}$$

As Silverman [22] suggested, σ_S is a single scale parameter that equals to the root of the average marginal variance in S, and h_{S_opt} is the optimal bandwidth value which equals to $A(K)|O|^{-1/(|S|+4)}$, where $A(K) = \{8{c_{|S|}}^{-1}(|S| + 4)(2\sqrt{\pi})^{|S|}\}^{1/(|S|+4)}$ for the Epanechnikov kernel.

We implemented CSMiner (Algorithm 2) using the Epanechnikov kernel for contrast subspace mining as follows. Given a subspace S, let $\mathcal{S}$ be the set of descendants of S in the subspace set enumeration tree using the standard deviation descending order. Then, $L_S(q \mid O_+)$ and $L_S(q \mid O_-)$ can be computed by Eq. 13, and $L_S^*(q \mid O_+) =$

$$\frac{1}{|O|(\sigma'_{min} h'_{opt\_min})^\tau} \sum_{o \in O \wedge \frac{dist_S(q,o)^2}{h_S^{max\,2}} < 1} \left(\frac{1}{2} {c_S^{min}}^{-1} (d_S^{max} + 2) \left(1 - \frac{dist_S(q, o)^2}{h_S^{max\,2}} \right) \right)$$

where $h_S^{max} = (\sigma_S h'_{opt\_max})$, $c_S^{min} = \min\{c_d \mid |S| < d \leq d_S^{max}\}$, $d_S^{max} = \max\{|S'| \mid S' \in \mathcal{S}\}$, and the meaning of σ'_{min}, h'_{opt_min}, h'_{opt_max}, τ are the same as those in Eq. 10.

Technically, the Epanechnikov kernel could also be implemented using the CSMiner-BPR approach (Algorithm 3). However, the performance improvement by the bounding-pruning-refining method would be less significant. The reason lies in the fact that on the one hand, different from using the Gaussian kernel that each object o has a nonzero likelihood contribution to the query object q, the contribution of o satisfying $\frac{dist_S(q,o)^2}{h_S^2} \geq 1$ is 0 (by the definition) to q when uses the Epanechnikov kernel. On the other hand, computing the neighborhood requires additional computational overhead.

Note that when using the Epanechnikov kernel, $\hat{f}_S(q, O_-) = 0$ if for any object $o \in O_-$, $\frac{dist_S(q,o)^2}{h_S^2} \geq 1$. Correspondingly, $LC_S(q) = \frac{\hat{f}_S(q,O_+)}{\hat{f}_S(q,O_-)} = +\infty$. Given data set O (composed by O_+ and O_-), we denote by $O_E^{+\infty}$ the set of objects whose maximum likelihood contrast, computed using the Epanechnikov kernel, is infinity. That is, $O_E^{+\infty} = \{o \in O \mid \exists S \text{ s.t. } LC_S(o) = +\infty\}$.

Let ℓ_G be the top-k contrast subspaces computed using the Gaussian kernel, and ℓ_E be the top-k contrast subspaces computed using the Epanechnikov kernel. For each object $q \in O$, we discover the top-10 inlying contrast subspaces and the top-10 outlying contrast subspaces of q using the Gaussian kernel and the Epanechnikov kernel, respectively, and compute $Sim(\ell_G, \ell_E)$ in each data set. For subspaces whose likelihood contrast values are infinity ($LC_S(q) = +\infty$), we sort them by $\hat{f}_S(q, O_+)$ in descending order. Tables 10 and 11 list the minimum, maximum, and average values of $Sim(\ell_G, \ell_E)$, as well as the ratio of $|O_E^{+\infty}|$ to $|O|$.

Table 10 Similarity between top-10 inlying contrast subspaces using different kernel functions in data set O ($\delta = 0.001$)

Data set O	$Sim(\ell_G, \ell_E)$			$\frac{\lvert O_E^{+\infty}\rvert}{\lvert O\rvert}$
	Min	Max	Avg	
BCW	0.168	0.980	0.539	590/683 = 0.864
CMSC	0.066	0.826	0.391	540/540 = 1.0
Glass	0.242	0.984	0.814	76/214 = 0.355
PID	0.620	1.0	0.924	1/768 = 0.001
Waveform	0.189	0.981	0.690	2532/5000 = 0.506
Wine	0.159	0.993	0.670	145/178 = 0.815

Table 11 Similarity between top-10 outlying contrast subspaces using different kernel functions in data set O ($\delta = 0.001$)

Data set O	$Sim(\ell_G, \ell_E)$			$\frac{\lvert O_E^{+\infty}\rvert}{\lvert O\rvert}$
	Min	Max	Avg	
BCW	0.239	1.0	0.916	67/683 = 0.098
CMSC	0.174	0.926	0.614	540/540 = 1.0
Glass	0.358	1.0	0.906	16/214 = 0.075
PID	0.655	1.0	0.938	1/768 = 0.001
Waveform	0.364	0.998	0.820	894/5000 = 0.179
Wine	0.209	1.0	0.804	40/178 = 0.225

Table 12 Similarity between top-10 inlying contrast subspaces using different kernel functions in data set $O \backslash O_E^{+\infty}$ ($\delta = 0.001$)

Data set $O \backslash O_E^{+\infty}$	$Sim(\ell_G, \ell_E)$			$\lvert O \backslash O_E^{+\infty} \rvert$
	Min	Max	Avg	
BCW	0.643	0.980	0.922	93
Glass	0.720	0.984	0.929	138
PID	0.620	1.0	0.924	767
Waveform	0.324	0.981	0.754	2468
Wine	0.527	0.988	0.904	33

Table 13 Similarity between top-10 outlying contrast subspaces using different kernel functions in data set $O \backslash O_E^{+\infty}$ ($\delta = 0.001$)

Data set $O \backslash O_E^{+\infty}$	$Sim(\ell_G, \ell_E)$			$\lvert O \backslash O_E^{+\infty} \rvert$
	Min	Max	Avg	
BCW	0.561	1.0	0.934	616
Glass	0.629	1.0	0.925	198
PID	0.655	1.0	0.938	767
Waveform	0.437	0.998	0.836	4106
Wine	0.482	1.0	0.863	138

From the results, shown in Tables 10 and 11, we can see that the value of $Sim(\ell_G, \ell_E)$ is related to $\frac{|O_E^{+\infty}|}{|O|}$. Specifically, the smaller the value of $\frac{|O_E^{+\infty}|}{|O|}$ the more similar ℓ_G and ℓ_E are. For example, when mining inlying contrast subspaces (Table 10), the values of $\frac{|O_E^{+\infty}|}{|O|}$ in BCW, CMSC, Waveform, and Wine are larger than 0.5, which is larger than the values of $\frac{|O_E^{+\infty}|}{|O|}$ in PID and Glass, while the values of $Sim(\ell_G, \ell_E)$ are lower in BCW, CMSC, Waveform, and Wine than those values in PID and Glass. When mining outlying contrast subspaces (Table 11), we note that the values of $\frac{|O_E^{+\infty}|}{|O|}$ are <0.1 in BCW, Glass, and PID, while the values of $Sim(\ell_G, \ell_E)$ in these data sets are over 0.9.

Furthermore, we compute $Sim(\ell_G, \ell_E)$ in $O \backslash O_E^{+\infty}$ for each data set except CMSC, because for CMSC, $O \backslash O_E^{+\infty} = \emptyset$. From the results shown in Table 12 (inlying contrast subspace mining) and Table 13 (outlying contrast subspace mining), we can see that ℓ_G is more similar to ℓ_E without considering the objects whose maximum likelihood contrast is infinity.

6 Conclusions

In this paper, we studied the novel and interesting problem of mining contrast subspaces to discover the aspects in which a query object is most similar to a class and dissimilar to another class. We demonstrated theoretically that the problem is very challenging and is MAX SNP-hard. We presented a heuristic method based on pruning rules and upper and lower bounds of likelihood and likelihood contrast. Our experiments on real data sets clearly show that our method improves contrast subspace mining substantially compared to the baseline method.

As future work, we intend to investigate the use of contrast subspaces for improving the accuracy of supervised learning methods. It is also interesting to consider using contrast

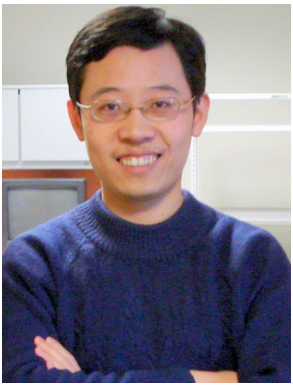
subspaces to characterize a given data set. Moreover, we will explore parallel computation approaches to improving the efficiency of CSMiner, and extend CSMiner for complex data sets involving both nominal and numerical values.

Acknowledgments The authors are grateful to the editor and the anonymous reviewers for their constructive comments, which help to improve this paper. Lei Duan's research was supported in part by National Natural Science Foundation of China (Grant No. 61103042), China Postdoctoral Science Foundation (Grant No. 2014M552371), and SRFDP 20100181120029. Jian Pei's and Guanting Tang's research was supported in part by an NSERC Discovery grant, a BCIC NRAS Team Project. James Bailey's work was supported by an ARC Future Fellowship (FT110100112). Work by Lei Duan and Guozhu Dong at Simon Fraser University was supported in part by an Ebco/Eppich visiting professorship. All opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

1. Aggarwal CC, Yu PS (2001) Outlier detection for high dimensional data. ACM Sigmod Rec 30:37–46
2. Bache K, Lichman M (2013) UCI machine learning repository
3. Bay SD, Pazzani MJ (2001) Detecting group differences: mining contrast sets. Data Min Knowl Discov 5(3):213–246
4. Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) When is "nearest neighbor" meaningful? In: Proc. of the 7th Int'l Conf on Database Theory, pp 217–235
5. Böhm K, Keller F, Müller E, Nguyen HV, Vreeken J (2013) CMI: An information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In: Proc. of the 13th SIAM Int'l Conf on Data Min, pp 198–206
6. Breunig MM, Kriegel HP, Ng RT, Sander J (2000) LOF: Identifying density-based local outliers. In: Proc. of the 2000 ACM SIGMOD Int'l Conf on Manag of data, pp 93–104
7. Cai Y, Zhao HK, Han H, Lau RYK, Leung HF, Min H (2012) Answering typicality query based on automatically prototype construction. In: Proc. of the 2012 IEEE/WIC/ACM Int'l Joint Conf Web Intell Intell Agent Technol, 01:362–366
8. Chen L, Dong G (2006) Masquerader detection using OCLEP: one class classification using length statistics of emerging patterns. In: Proc. of Int'l workshop on information Processing over Evolving Networks (WINPEN), p 5
9. Dong G, Bailey J (eds) (2013) Contrast data mining: concepts, algorithms, and applications. CRC Press, Boca Raton
10. Dong G, Li J (1999) Efficient mining of emerging patterns: discovering trends and differences. In: Proc. of the 5th ACM SIGKDD Int'l Conf on Knowledge Discovery and Data Mining, pp 43–52
11. Duan L, Tang G, Pei J, Bailey J, Dong G, Campbell A, Tang C (2014) Mining contrast subspaces. In: Proc. of the 18th Pacific-Asia Conf on Knowledge Discovery and Data Mining, pp 249–260
12. Fagin R, Kumar R, Sivakumar D (2003) Comparing top k lists. In: Proc. of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp 28–36
13. He Z, Xu X, Huang ZJ, Deng S (2005) FP-outlier: frequent pattern based outlier detection. Comput Sci Inf Syst 2(1):103–118
14. Hua M, Pei J, Fu AW, Lin X, Leung HF (2009) Top-k typicality queries and efficient query answering methods on large databases. VLDB J 18(3):809–835
15. Jeffreys H (1961) The theory of probability, 3rd edn. Oxford
16. Keller F, Müller E, Böhm K (2012) HiCS: high contrast subspaces for density-based outlier ranking. In: Proc. of the IEEE 28th Int'l Conf on Data Engineering, pp 1037–1048
17. Kriegel HP, Schubert M, Zimek A (2008) Angle-based outlier detection in high-dimensional data. In: Proc. of the 14th ACM SIGKDD Int'l Conf on Knowledge Discovery and Data Mining, pp 444–452
18. Kriegel HP, Kröger P, Schubert E, Zimek A (2009) Outlier detection in axis-parallel subspaces of high dimensional data. In: Proc. of the 13th Pacific-Asia Conf on Knowledge Discovery and Data Mining, pp 831–838
19. Novak PK, Lavrac N, Webb GI (2009) Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern and subgroup mining. J Mach Learn Res 10:377–403

20. Papadimitriou CH, Yannakakis M (1991) Optimization, approximation, and complexity classes. J Comput Syst Sci 43(3):425–440
21. Rymon R (1992) Search through systematic set enumeration. In: Proc. of the 3rd Int'l Conf on Principles of Knowledge Representation and Reasoning, pp 539–550
22. Silverman BW (1986) Density estimation for statistics and data analysis. Chapman and Hall/CRC, London
23. Wang L, Zhao H, Dong G, Li J (2005) On the complexity of finding emerging patterns. Theor Comput Sci 335(1):15–27
24. Webber W, Moffat A, Zobel J (2010) A similarity measure for indefinite rankings. ACM Trans Inf Syst 28(4):20:1–20:38
25. Wrobel S (1997) An algorithm for multi-relational discovery of subgroups. In: Proc. of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery, pp 78–87
26. Wu S, Crestani F (2003) Methods for ranking information retrieval systems without relevance judgments. In: Proc. of the 2003 ACM Symposium on Applied Computing. ACM, New York, NY, USA, pp 811–816



Lei Duan is currently an Associate Professor in the School of Computer Science at Sichuan University. He received his B.Sc. and Ph.D. degrees both in Computer Science from Sichuan University in 2003 and 2008, respectively. He was a visiting Ph.D. student in the Department of Computer Science and Engineering at Wright State University from 2007 to 2008, and was a visiting scholar in the School of Computing Science at Simon Fraser University from 2012 to 2013. His research interests include data mining, knowledge management, evolutionary computation, bioinformatics and health-informatics.

Guanting Tang received her B.Sc. degree in Computer Science from Shanghai Ocean University, China, in 2007 and her M.Sc. degree in Computer Science from Georgia Southwestern State University, USA, in 2008. She is currently a Ph.D. candidate at the School of Computing Science at Simon Fraser University, Canada. Her research interests include data mining, text mining, machine learning, information retrieval and natural language processing.

Jian Pei is a Professor of Computing Science and Statistics at Simon Fraser University, Canada. His expertise is on developing effective and efficient data analysis techniques for novel data intensive applications. Particularly, he is currently interested in and actively working on developing various techniques of data mining, web search, information retrieval, data warehousing, online analytical processing, and database systems, as well as their applications in social networks, health-informatics, and business intelligence. Since 2000, he has published one textbook, two monographs and over 200 research papers in refereed journals and conferences, and has served in the organization committees and the program committees of over 200 international conferences and workshops. He is the editor-in-chief of the *IEEE Transactions on Knowledge and Data Engineering*, and an associate editor or editorial board member of several major academic journals in his fields. He is a fellow of IEEE and a senior member and a distinguished speaker of ACM.

James Bailey is a Professor and Australian Research Council (ARC) Future Fellow in the Department of Computing and Information Systems at the University of Melbourne. He has an extensive track record in databases and data mining and has been chief investigator on multiple ARC discovery grants. He has been the recipient of five best paper awards and is an active member of the knowledge discovery community. He is an Associate Editor for the journals *IEEE Transactions on Knowledge and Data Engineering*, *Knowledge and Information Systems* and *Social Network Analysis and Mining*. He regularly serves as a Senior PC member for top conferences in data mining and he will be the co-general chair for ACM CIKM 2015 conference.

Guozhu Dong is currently a full Professor at Wright State University. He received his Ph.D. in Computer Science from the University of Southern California in 1988. His main research interests are data science, data mining, and bioinformatics. He has published over 150 articles and two books, and he holds 4 US patents. He was a recipient of the Best Paper Award from the 2005 IEEE ICDM and the 2014 PAKDD. He is a senior member of IEEE and ACM. He has served on over a hundred program committees of international conferences, including serving as program committee chairs.

Vinh Nguyen is currently a Research Fellow at the Department of Computing and Information Systems, the University of Melbourne. He has made significant contribution to promoting information theoretic approaches for data mining, in particular clustering and feature selection, with more than 500 citations since 2010. His current research focuses on big data analytics in diverse domains, including bioinformatics, transportation and social networks.

Akiko Campbell is currently at Pacific Blue Cross, British Columbia's leading benefits provider for over 75 years. As the Director of Innovation Centre, Akiko leads company's Research and Development activities, ranging from application development to data analytics. Through collaboration with Simon Fraser University, Akiko's priority in research is computational health informatics for developing business intelligence for the company as well as its customers.

Changjie Tang is a Professor, the Director of the Institute of Database and Knowledge Engineering in the School of Computer Science at Sichuan University, and is a Vice Director of China Computer Federation Technical Committee on Databases. His research interests include database theory, data mining and knowledge discovery, data cube and OLAP, and information security.