in Proc. 4th IEEE Int. Conf. on Data Mining (ICDM 04), Brighton, UK, 2004

Subspace Selection for Clustering High-Dimensional Data

Christian Baumgartner, Claudia Plant
University for Health Sciences, Medical Informatics and Technology, Innsbruck, Austria
{christian.baumgartner,claudia.plant}@umit.at

Karin Kailing, Hans-Peter Kriegel, Peer Kröger
Institute for Computer Science, University of Munich, Germany
{kailing,kriegel,kroegerp}@dbs.ifi.lmu.de

Abstract

In high-dimensional feature spaces traditional clustering algorithms tend to break down in terms of efficiency and quality. Nevertheless, the data sets often contain clusters which are hidden in various subspaces of the original feature space. In this paper, we present a feature selection technique called SURFING (SUbspaces Relevant For clusterING) that finds all subspaces interesting for clustering and sorts them by relevance. The sorting is based on a quality criterion for the interestingness of a subspace using the k-nearest neighbor distances of the objects. As our method is more or less parameterless, it addresses the unsupervised notion of the data mining task "clustering" in a best possible way. A broad evaluation based on synthetic and real-world data sets demonstrates that SURFING is suitable to find all relevant subspaces in high dimensional, sparse data sets and produces better results than comparative methods.

1. Introduction

One of the primary data mining tasks is clustering which is intended to help a user discovering and understanding the natural structure or grouping in a data set. In particular, clustering aims at partitioning the data objects into distinct groups (clusters) while minimizing the intra-cluster similarity and maximizing the inter-cluster similarity. A lot of work has been done in the area of clustering (see e.g. [8] for an overview). However, many real-world data sets consist of very high dimensional feature spaces. In such high dimensional feature spaces, most of the common algorithms tend to break down in terms of efficiency and accuracy because usually many features are irrelevant and or correlated. In addition, different subgroups of features may be irrelevant or correlated according to varying subgroups of data objects. Thus, objects can often be clustered differently in varying subspaces. Usually, global dimensionality reduction techniques such as PCA cannot be applied to these data sets because they cannot account for local trends in the data.

To cope with these problems, the procedure of feature selection has to be combined with the clustering process more closely. In recent years, the task of subspace clustering was introduced to address these demands. In general, subspace clustering is the task of automatically detecting all clusters in all subspaces of the original feature space, either by directly computing the subspace clusters (e.g. in [3]) or by selecting interesting subspaces for clustering (e.g. in [9]).

In this paper, we propose an advanced feature selection method preserving the information of objects clustered differently in varying subspaces. Our method called SURFING (*SU*bspaces *R*elevant *F*or cluster*ING*) computes all relevant subspaces and ranks them according to the interestingness of the hierarchical clustering structure they exhibit.

The remainder of this paper is organized as follows. We discuss related work and point out our contributions in Section 2. A quality criterion for ranking the interestingness of subspaces is developed in Section 3. In Section 4 the algorithm SURFING is presented. An experimental evaluation of SURFING in the context of comparative subspace clustering methods is presented in Section 5. Section 6 concludes the paper.

2. Related Work

2.1. Subspace Clustering

The pioneering approach to subspace clustering is CLIQUE [3], using an *Apriori*-like method to navigate through the set of possible subspaces. The data space is

partitioned by an axis-parallel grid into equi-sized units of width ξ. Only units whose densities exceed a threshold τ are retained. A cluster is defined as a maximal set of connected dense units. The performance of CLIQUE heavily depend on the positioning of the grid. Objects that naturally belong to a cluster may be missed or objects that are naturally noise may be assigned to a cluster due to an unfavorable grid position.

Another recent approach called DOC [10] proposes a mathematical formulation for the notion of an optimal projected cluster, regarding the density of points in subspaces. DOC is not grid-based but as the density of subspaces is measured using hypercubes of fixed width w, it has similar problems like CLIQUE.

In [2] the method PROCLUS to compute projected clusters is presented. However, PROCLUS misses out the information of objects clustered differently in varying subspaces. The same holds for ORCLUS [1].

2.2. Feature Selection for Clustering

In [9] a method called RIS is proposed that ranks the subspaces according to their clustering structure. The ranking is based on a quality criterion using the density-based clustering notion of DBSCAN [7]. An *Apriori*-like navigation through the set of possible subspaces in a bottom-up way is performed to find all interesting subspaces. Aggregated information is accumulated for each subspace to rank its interestingness.

In [6] a quality criterion for subspaces based on the entropy of point-to-point distances is introduced. However, there is no algorithm presented to compute the interesting subspaces. The authors propose to use a forward search strategy which most likely will miss interesting subspaces, or an exhaustive search strategy which is obviously not efficient in higher dimensions.

2.3. Our Contributions

Recent density-based approaches to subspace clustering or subspace selection methods (RIS) use a global density threshold for the definition of clusters due to efficiency reasons. However, the application of a global density threshold to subspaces of different dimensionality and to all clusters in one subspace is rather unacceptable. The data space naturally increases exponentially with each dimension added to a subspace and clusters in the same subspace may exceed different density parameters or exhibit a nested hierarchical clustering structure. Therefore, for subspace clustering, it would be highly desirable to adapt the density threshold to the dimensionality of the subspaces or even better to rely on a hierarchical clustering notion that is independent from a globally fixed threshold.

In this paper, we introduce SURFING, a feature selection method for clustering which does not rely on a global density parameter. Our approach explores all subspaces exhibiting an *interesting* hierarchical clustering structure and ranks them according to a quality criterion. SURFING is more or less parameterless, i.e. it does not require the user to specify parameters that are hard to anticipate such as the number of clusters, the (average) dimensionality of subspace clusters, or a global density threshold. Thus, our algorithm addresses the *unsupervised* notion of the data mining task "clustering" in a best possible way.

3. Subspaces Relevant for Clustering

Let DB be a set of N feature vectors with dimensionality d, i.e. $DB \subseteq \mathbb{R}^d$. Let $\mathcal{A} = \{a_1, \ldots, a_d\}$ be the set of all attributes a_i of DB. Any subset $S \subset \mathcal{A}$, is called a *subspace*. T is a *superspace* of S if $S \subset T$. The projection of an object o onto a subspace $S \subseteq \mathcal{A}$ is denoted by o_S. We assume that $d : DB \times DB \to \mathbb{R}$ is a metric distance function.

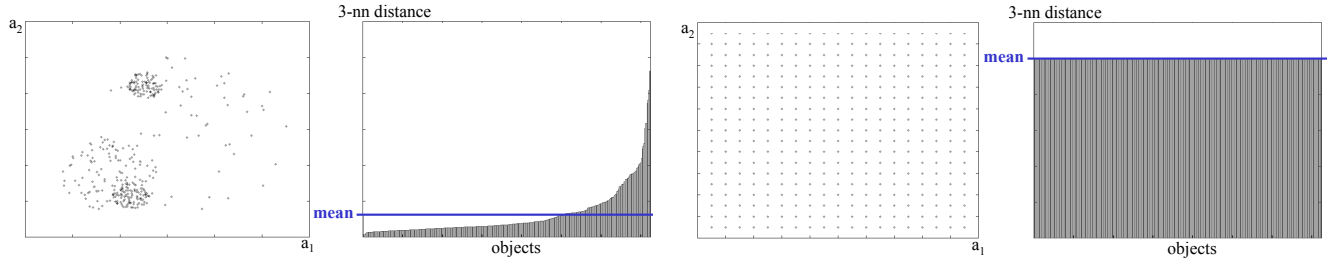
3.1. General Idea

The main idea of SURFING is to measure the "interestingness" of a subspace w.r.t. to its hierarchical clustering structure, independent from its dimensionality. Like most previous approaches to subspace clustering, we base our measurement on a density-based clustering notion. Since we do not want to rely on a global density parameter, we developed a quality criterion for relevant subspaces built on the k-nearest neighbor distances (k-nn-distances) of the objects in DB.

For a user-specified $k \in \mathbb{N}$ ($k \leq N$) and a subspace $S \subseteq \mathcal{A}$ let $NN_k^S(o)$ be the set of k-nearest neighbors of an object $o \in DB$ in a subspace S. The *k-nn-distance* of o in a subspace S, denoted by $nn\text{-}Dist_k^S(o)$, is the distance between o and its k-nearest neighbor, formally:

$$nn\text{-}Dist_k^S(o) = \max\{d(o_S, p_S) \,|\, p \in NN_k^S(o)\}.$$

The k-nn-distance of an object o indicates how densely the data space is populated around o in S. The smaller the value of $nn\text{-}Dist_k^S(o)$, the more dense the objects are packed around o, and *vice versa*. If a subspace contains a recognizable hierarchical clustering structure, i.e. clusters with different densities and noise objects, the k-nn-distances of objects should differ significantly. On the other hand, if all points are uniformly distributed, the k-nn-distances can be assumed



(a) Hierarchical clustering structure in a 2D subspace(left); according sorted 3-nn graph (right)

(b) Uniform distribution in a 2D subspace (left); according sorted 3-nn graph (right)

Figure 1: Usefulness of the k-nn distance to rate the interestingness of subspaces.

to be almost equal. Figure 1 illustrates these considerations using a sample 2D subspace $S = \{a_1, a_2\}$ and $k = 3$. Consequently, we are interested in subspaces where the k-nn-distances of the objects differ significantly from each other, because the hierarchical clustering structure in such subspaces will be considerably clearer than in subspaces where the k-nn-distances are rather similar to each other.

3.2. A Quality Criterion for Subspaces

As mentioned above we want to measure how much the k-nn-distances in S differ from each other. To achieve comparability between subspaces of different dimensionality, we scale all k-nn-distances in a subspace S into the range $[0, 1]$. Thus, we assume that $nn\text{-}Dist_k^S(o) \in [0, 1]$ for all $o \in DB$ throughout the rest of the paper.

Two well-known statistical measures for our purpose are the mean value μ_S of all k-nn-distances in subspace $S$and the variance. However, the variance is not appropriate for our purpose because it measures the squared differences of each k-nn-distance to μ_S and thus, high differences are weighted stronger than low differences. For our quality criterion we want to measure the non-weighted differences of each k-nn-distance to μ_S. Since the sum of the differences of all objects *above* μ_S is equal to the sum of the differences of all objects *below* μ_S, we only take half of the sum of all differences to the mean value, denoted by $diff_{\mu_S}$, which can be computed by

$$diff_{\mu_S} = \frac{1}{2} \sum_{o \in DB} |\mu_S - nn\text{-}Dist_k^S(o)|.$$

In fact, $diff_{\mu_S}$ is already a good measure for rating the interestingness of a subspace. We can further scale this value by μ_S times the number of objects having a smaller k-nn-distance in S than μ_S, i.e. the objects contained in the following set:

$$Below_S := \{o \in DB \mid nn\text{-}Dist_k^S(o) < \mu_S\}.$$

Obviously, if $Below_S$ is empty, the subspace contains uniformly distributed noise.

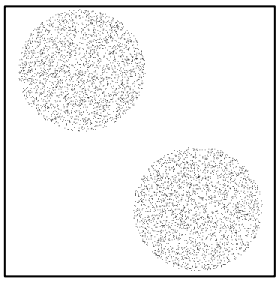
Definition 1 (quality of a subspace) *Let $S \subseteq \mathcal{A}$. The quality of S, denoted by $quality(S)$, is defined as follows:*

$$quality(S) = \begin{cases} 0 & if\, Below_S = \emptyset \\ \frac{diff_{\mu_S}}{|Below_S| \cdot \mu_S} & else. \end{cases}$$

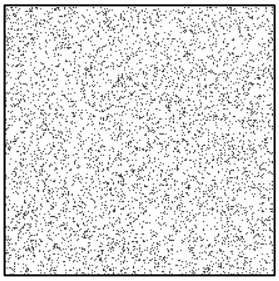
The quality values are in the range between 0 and 1. A subspace where all objects are uniformly distributed (e.g. as depicted in Figure 1(b)) has a quality value of approximately 0, indicating a less interesting clustering structure. On the other hand, the clearer the hierarchical clustering structure in a subspace S is, the higher is the value of $quality(S)$. For example, the sample 2D subspace in which the data is highly structured as depicted in Figure 1(a) will have a significantly higher quality value. Let us note that in the synthetic case where all objects in $Below_S$ have a k-nn-distance of 0 and all other objects have a k-nn-distance of $2 \cdot \mu_S$, the quality value $quality(S)$ is 1.

In almost all cases, we can detect the relevant subspaces with this quality criterion, but there are two artificial cases rarely found in natural data sets which nevertheless cannot be ignored.

First, there might be a subspace containing some clusters, each of the same density, and without noise (e.g. data set A in Figure 2). If the number of data objects in the clusters exceeds k, such subspaces cannot be distinguished from subspaces containing uniformly distributed data objects spread over the whole attribute range (e.g. data set B in Figure 2) because in both



data set A



data set B

% of inserted points	$quality(A)$	$quality(B)$
0	0.13	0.15
0.1	0.15	0.15
0.2	0.19	0.15
0.5	0.31	0.15
1	0.38	0.15
5	0.57	0.15
10	0.57	0.15

Figure 2: Benefit of inserted points.

cases, the k-nn-distances of the objects will marginally differ from the mean value.

Second, subspaces containing data of one Gaussian distribution spread over the whole attribute range are not really interesting. However, the k-nn-distances of the objects will scatter significantly around the mean value. Thus, such subspaces cannot be distinguished from subspaces containing two or more Gaussian clusters without noise.

To overcome these two artificial cases, we can virtually insert some randomly generated points before computing the quality value of a subspace. In cases of uniform or Gaussian distribution over the whole attribute range, the insertion of a few randomly generated additional objects does not significantly affect the quality value. The k-nn-distances of these objects are similar to the k-nn-distances of all the other data objects. However, if there are dense and empty areas in a subspace, the insertion of some additional points very likely increases the quality value, because these additional objects have large k-nn-distances compared to those of the other objects. The table in Figure 2 shows the quality value of the 2D data set A depicted in Figure 2 w.r.t. the percentage of virtually inserted random objects. Data set B in Figure 2 has no visible cluster structure and therefore the virtually inserted points do not affect the quality value. For example, 0.2 % additionally inserted points means that for $n = 5,000$ 10 random objects have been virtually inserted before calculating the quality value.

Thus, inserting randomly generated points is a proper strategy to distinguish (good) subspaces containing several uniformly distributed clusters of equal density or several Gaussian clusters without noise from (bad) subspaces containing only one uniform or Gaussian distribution. In fact, it empirically turned out that 1% of additional points is sufficient to achieve the desired results. Let us note that this strategy is only required, if the subspaces contain a clear clustering structure without noise. In most real-world data sets the subspaces do not show a clear cluster structure and often have much more than 10% noise. In addition, the number of noise objects is usually growing with increasing dimensionality. In such data sets, virtually inserting additional points is not required. Since our quality criterion is very sensible to areas of different density, it is suitable to detect relevant subspaces in data sets with high percentages of noise, e. g. in gene expression data sets or in synthetic data sets containing up to 90% noise.

4. Algorithm

The pseudocode of the algorithm SURFING is given in Figure 3. Since lower dimensional subspaces are more likely to contain an interesting clustering, SURFING generates all relevant subspaces in a bottom-up way, i.e. it starts with all 1-dimensional subspaces $\mathcal{S}_1$ and discards as many irrelevant subspaces as early as possible. Therefore, we need a criterion to decide whether it is interesting to generate and examine a certain subspace or not. Our above described quality measure can only be used to decide about the interestingness of an already given subspace. An important information we have gathered while proceeding to dimension l is the quality of all $(l-1)$-dimensional subspaces. We can use this information to compute a quality threshold which enables us to rate all l-dimensional candidate subspaces $\mathcal{S}_l$. We use the lowest quality value of any $(l-1)$-dimensional subspace as threshold. If the quality values of the $(l-1)$-dimensional subspaces do not differ enough (it empirically turned out that a difference of at least 1/3 is a reasonable reference difference), we take half of the best quality value instead. Using this quality threshold, we can divide all l-dimensional subspaces into three different categories:

Interesting subspace: the quality value increases or stays the same w.r.t. its $(l-1)$-dimensional subspaces.

Neutral subspaces: the quality decreases w.r.t. its $(l-1)$-dimensional subspaces, but lies above the threshold and thus might indicate a higher dimensional interesting subspace.

Irrelevant subspaces: the quality decreases w.r.t its

$(l-1)$-dimensional subspace below the threshold.
We use this classification to discard all irrelevant l-dimensional subspaces from further consideration. We know that these subspaces are not interesting itself and, as our quality value is comparable over different dimensions, we further know that no superspace of such a subspace will obtain a high quality value compared to interesting subspaces of dimensionality l. Even if through adding a "good" dimension, the quality value would slightly increase it will not be getting better than already existing ones.

However, before we discard an irrelevant subspace S of dimensionality l, we have to test whether its clustering structure exhibits one of the artificial cases mentioned in the previous section. For that purpose, if the quality of S is lower than the quality of a subspace containing an l-dimensional Gaussian distribution, we insert 1% random points and recompute the quality of S. Otherwise, the clustering structure of S cannot get better through the insertion of additional points. In case of a clean cluster structure without noise in S, the quality value improves significantly after the insertion. At least it will be better than the quality of the l-dimensional Gaussian distribution, and, in this case, S is not discarded.

If, due to the threshold, there are only irrelevant l-dimensional subspaces, we don't use the threshold, but keep all l-dimensional subspaces. In this case, the information we have so far, is not enough to decide about the interestingness.

Finally, the remaining l-dimensional subspaces in $\mathcal{S}_l$ are joined if they share any $(l-1)$-dimensions to generate the set of $(l+1)$-dimensional candidate subspaces $\mathcal{S}_{l+1}$. SURFING terminates if the resulting candidate set is empty.

SURFING needs only one input parameter k, the choice of which is rather simple. If k is too small, the k-nn-distances are not meaningful, since objects within dense regions might have similar k-nn-distance values as objects in sparse regions. If k is too high, the same phenomenon may occur. Obviously, k must somehow correspond to the minimum cluster size, i.e. the minimal number of objects regarded as a cluster.

algorithm SURFING(Database DB, Integer k)
// **1-dimensional subspaces**
  $\mathcal{S}_1 := \{\{a_1\}, \ldots, \{a_d\}\}$;
  compute quality of all subspaces $S \in \mathcal{S}_1$;
  $S_l := S \in \mathcal{S}_1$ with lowest quality;
  $S_h := S \in \mathcal{S}_1$ with highest quality;
  if $quality(S_l) > \frac{2}{3} \cdot quality(S_h)$ **then**
    $\tau := \frac{quality(S_h)}{2}$;
  else
    $\tau := quality(S_l)$;
    $\mathcal{S}_1 = \mathcal{S}_1 - \{S_l\}$;
  end if
// k-**dimensional-subspaces**
  $k := 2$;
  create $\mathcal{S}_2$ from $\mathcal{S}_1$;
  while not $\mathcal{S}_k = \emptyset$ **do**
    compute quality of all subspaces S in $\mathcal{S}_k$;
    $Interesting := \{S \in \mathcal{S}_k | quality(S) \uparrow\}$;
    $Neutral := \{S \in \mathcal{S}_k | quality(s) \downarrow \wedge quality(S) > \tau\}$;
    $Irrelevant := \{S \in \mathcal{S}_k | quality(S) \leq \tau\}$;
    $S_l := S \in \mathcal{S}_k$ with lowest quality;
    $S_h := S \in \mathcal{S}_k - Interesting$ with highest quality;
    if $quality(S_l) > \frac{2}{3} \cdot quality(S_h)$ **then**
      $\tau := \frac{quality(S_h)}{2}$;
    else
      $\tau := quality(s_l)$;
    end if
    if not all subspaces irrelevant **then**
      $\mathcal{S}_k := \mathcal{S}_k - Irrelevant$;
    end if
    create $\mathcal{S}_{k+1}$ from $\mathcal{S}_k$;
    $k := k + 1$;
  end while
end

Figure 3: Algorithm SURFING.

5. Evaluation

We tested SURFING on several synthetic and real-world data sets and evaluated its accuracy in comparison to CLIQUE, RIS and the subspace selection proposed in [6] (in the following called Entropy). All experiments were run on a PC with a 2.79 GHz CPU and 504 MB RAM. We combined SURFING, RIS and Entropy with the hierarchical clustering algorithm OPTICS [4] to compute the hierarchical clustering structure in the detected subspaces.

Synthetic Data. The synthetic data sets were generated by a self-implemented data generator. It permits to specify the number and dimensionality of subspace clusters, dimensionality of the feature space and density parameters for the whole data set as well as for each cluster. In a subspace that contains a cluster, the average density within that cluster is much larger than the density of noise. In addition, it is ensured that none of the synthetically generated data sets can be clustered in the full dimensional space.

Gene Expression Data. We tested SURFING on a real-world gene expression data set studying the yeast mitotic cell cycle [11]. We used only the data set of the CDC15 mutant and eliminated those genes from our test data set having missing attribute values. The re-

Table 1: Results on synthetic data sets.

data set	d	cluster dim.	N	# subspaces m	# subspaces %	time (s)
02	10	4	4936	107	10.45	351
03	10	4	18999	52	5.08	2069
04	10	4	27704	52	5.08	4401
05	15	2	4045	119	0.36	194
06	15	5	3802	391	1.19	807
07	15	3,5,7	4325	285	0.87	715
08	15	5	4057	197	0.60	391
09	15	7	3967	1046	3.19	3031
10	15	12	3907	4124	12.59	15321
11	10	5	3700	231	22.56	442
12	20	5	3700	572	0.05	1130
13	30	5	3700	1077	0.0001	2049
14	40	5	3700	1682	$1.5{\cdot}10^{-7}$	3145
15	50	5	3700	2387	$2.1{\cdot}10^{-10}$	4255
16	15	4,6,7,10	2671	912	2.8	4479

sulting data set contains around 4000 genes measured at 24 different time slots. The task is to find functionally related genes using cluster analysis.

Metabolome Data. In addition we tested SURFING on high-dimensional metabolic data provided from the newborn screening program in Bavaria, Germany. Our experimental data sets were generated from modern tandem mass spectrometry. In particular we focused on a dimensionality of 14 metabolites in order to mine single and promising combinations of key markers in the abnormal metabolism of phenylketonuria (PKU), a severe amino acid disorder. The resulting database contains 319 cases designated as PKU and 1322 control individuals expressed as 14 amino acids and intermediate metabolic products.The task is to extract a subset of metabolites that correspond well to the abnormal metabolism of PKU.

5.1. Efficiency

The runtimes of SURFING applied to the synthetic data sets are summarized in Table 1. In all experiments, we set $k = 10$.

For each subspace, SURFING needs $O(N^2)$ time to compute for each of the N points in DB the k-nn-distance, since there is no index structure which could support the partial k-nn-queries in arbitrary subspaces in logarithmic time. If SURFING analyzes m different subspaces the overall runtime complexity is $O(m \cdot N^2)$. Of course in the worst case m can be 2^d, but in practice we are only examining a very small percentage of all possible subspaces. Indeed, our experiments show, that the heuristic generation of subspace candidates used by SURFING ensures a small value for m (cf. Table 1). For most complex data sets, SURFING computes less than 5% of the total number of possible subspaces. In most cases, this ratio is even significantly less than 1%. For data set 10 in Table 1 where the cluster is hidden in a 12-dimensional subspace of a 15-dimensional feature space, SURFING only computes 12.5% of the possible subspaces. Finally, for both the real world data sets, SURFING computes even significantly less than 0.1% of the possible subspaces (not shown in Table 1). The worst ever observed percentage was around 20%. This empirically demonstrates that SURFING is a highly efficient solution for the complex subspace selection problem.

5.2. Effectivity

Results on Synthetic Data. We applied SURFING to several synthetic data sets (cf. Table 1). In all but one case, SURFING detected the correct subspaces containing the relevant clusters and ranked them first. Even for data set 16, SURFING was able to detect 4 out of 5 subspaces containing clusters, although the clustering structure of the subspaces containing clusters was rather weak, e.g. one of the 4-dimensional subspaces contained a cluster with only 20 objects having an average k-nn-distance of 2.5 (the average k-nn-distance for all objects in all dimensions was 15.0). SURFING only missed a 10-dimensional subspace which contained a cluster with 17 objects having an average k-nn-distance of 9.0.

Results on Gene Expression Data. We tested SURFING on the gene expression data set and retrieved a hierarchical clustering by applying OPTICS [4] to the top-ranked subspaces. We found many biologically interesting and significant clusters in several subspaces. The functional relationships of the genes in the resulting clusters were validated using the public Saccharomyces Genome Database[1]. Some excerpts from sample clusters in varying subspaces found by SURFING applied to the gene expression data are depicted in Table 2. Cluster 1 contains several cell cycle genes. In addition, the two gene products are part of a common protein complex. Cluster 2 contains the gene STE12, an important regulatory factor for the mitotic cell cycle [11] and the genes CDC27 and EMP47 which are most likely co-expressed with STE12. Cluster 3 consists of the genes CDC25 (starting point for mitosis), MYO3 and NUD1 (known for an active role dur-

1 http://www.yeastgenome.org/

Table 2: Results on gene expression data.

Gene Name	Function
Cluster 1 (subspace 90, 110, 130, 190)	
RPC40	builds complex with CDC60
CDC60	tRNA synthetase
FRS1	tRNA synthetase
DOM34	protein synthesis, mitotic cell cycle
CKA1	mitotic cell cycle control
MIP6	RNA binding activity, mitotic cell cycle
Cluster 2 (subspace 90, 110, 130, 190)	
STE12	transcription factor (cell cycle)
CDC27	possible STE12-site
EMP47	possible STE12-site
XBP1	transcription factor
Cluster 3 (subspace 90, 110, 130, 190)	
CDC25	starting control factor for mitosis
MYO3	control/regulation factor for mitosis
NUD1	control/regulation factor for mitosis
Cluster 4 (subspace 190, 270, 290)	
RPT6	protein catabolism; complex with RPN10
RPN10	protein catabolism; complex with RPT6
UBC1	protein catabolism; part of 26S protease
UBC4	protein catabolism; part of 26S protease
Cluster 5 (subspace 70, 90, 110, 130)	
SOF1	part of small ribosomal subunit
NAN1	part of small ribosomal subunit
RPS1A	structural constituent of ribosome
MIP6	RNA binding activity, mitotic cell cycle
Cluster 6 (subspace 70, 90, 110, 130)	
RIB1	participate in riboflavin biosynthesis
RIB4	participate in riboflavin biosynthesis
RIB5	participate in riboflavin biosynthesis

ing mitosis) and various other transcription factors required during the cell cycle. Cluster 4 contains several genes related to the protein catabolism. Cluster 5 contains several structural parts of the ribosomes and related genes. Let us note, that MPI6 is clustered differently in varying subspaces (cf. Cluster 1 and Cluster 5). Cluster 6 contains the genes that code for proteins participating in a common pathway.

Results on Metabolome Data. Applying SURFING to metabolic data, we identified 13 subspaces considering quality values > 0.8. In detail, we extracted 5 one-dimensional spaces (the metabolites ArgSuc, Phe, Glu, Cit and Arg), 6 two- dimensional spaces (e.g. Phe-ArgSuc, Phe-Glu) and 3 three-dimensional spaces (e.g. Phe-Glu-ArgSuc). Alterations of our best ranked single metabolites correspond well to the abnormal metabolism of PKU [5]. We compared SURFING findings with results using PCA. Only components with eigen value > 1 were extracted. Varimax rotation was applied. PCA findings showed 4 components (eigen values of components 1-4 are 4.039, 2.612, 1.137 and 1.033) that retain 63% of total variation. However, SURFING's best ranked single metabolites ArgSuc, Glu, Cit and Arg are not highly loaded (> 0.6) on one of four extracted components. Moreover, combinations of promising metabolites (higher dimensional subspaces) are not able to be considered in PCA. Particularly in abnormal metabolism, not only alterations of single metabolites but more interactions of several markers are often involved. As our results demonstrate, SURFING is more usable on metabolic data taking higher dimensional subspaces into account.

Table 3: Comparative tests on synthetic data.

data set	# clusters/ subspaces	correct clusters/subspaces found by CLIQUE	RIS	E	SURFING
06	2	1	2	0	2
07	3	1	2	0	2
08	3	1	3	0	3
16	5	0	3	0	4

Influence of Parameter k. We re-ran our experiments on the synthetic data sets with $k = 3, 5, 10, 15, 20$. We observed that if $k = 3$, SURFING did find the correct subspaces but did not rank the subspaces first (i.e. subspaces with a less clear hierarchical clustering structure got a higher quality value). In the range of $5 \leq k \leq 20$, SURFING produced similar results for all synthetic data sets. This indicates that SURFING is quite robust against the choice of k within this range.

Comparison with CLIQUE. The results of CLIQUE applied to the synthetic data sets confirmed the suggestions that its accuracy heavily depends on the choice of the input parameters which is a nontrivial task. In some cases, CLIQUE failed to detect the subspace clusters hidden in the data but computed some dubious clusters. In addition, CLIQUE is not able to detect clusters of different density. Applied to our data sets which exhibit several clusters with varying density (e.g. data set 16), CLIQUE was not able to detect all clusters correctly but could only detect (parts of) one cluster (cf. Table 3) — even though we used a broad parameter setting. A similar result can be reported when we applied CLIQUE to the gene expression data set. CLIQUE was not able to obtain any useful clusters for a broad range of parameter settings. In sum-

mary, SURFING does not only outperform CLIQUE by means of quality, but also saves the user from finding a suitable parameter setting.

Comparison with RIS. Using RIS causes similar problems as observed when using CLIQUE. The quality of the results computed by RIS also depends, with slightly less impact, on the input parameters. Like CLIQUE, in some cases RIS failed to detect the correct subspaces due to the utilization of a global density parameter (cf. Table 3). For example, applied to data set 16, RIS was able to compute the lower dimensional subspaces, but could not detect the higher dimensional one. The application of RIS to the gene expression data set is described in [9]. SURFING confirmed these results but found several other interesting subspaces with important clusters, e.g. clusters 5 and 6 in subspace 70, 90, 110, 130 (cf. Figure 2). Applying RIS to the metabolome data set the best ranked subspace contains 12 attributes which represent nearly the full feature space and are biologically not interpretable. The application of RIS to all data sets, was limited by the choice of the right parameter setting. Again, SURFING does not only outperform RIS by means of quality, but also saves the user from finding a suitable parameter setting.

Comparison with Entropy. Using the quality criterion Entropy (E) in conjunction with the proposed forward search algorithm in [6], none of the correct subspaces were found. In all cases, the subspace selection method stops at a dimensionality of 2. Possibly, an exhaustive search examining all possible subspaces could produce better results. However, this approach obviously yields unacceptable run times. Applied to the metabolome data, the biologically relevant 1D subspaces are ranked low.

6. Conclusion

In this paper, we introduced a new method to subspace clustering called SURFING which is more or less parameterless and — in contrast to most recent approaches — does not rely on a global density threshold. SURFING ranks subspaces of high dimensional data according to their interestingness for clustering. We empirically showed that the only input parameter of SURFING is stable in a broad range of settings and that SURFING does not favor subspaces of a certain dimensionality. A comparative experimental evaluation shows that SURFING is an efficient and accurate solution to the complex subspace clustering problem. It outperforms recent subspace clustering methods in terms of effectivity.

Acknowledgment

Parts of this work is supported by the German Ministry for Education, Science, Research and Technology (BMBF) (grant no. 031U112F) and by the Austrian Industrial Research Promotion Fund FFF (grand no. HITT-10 UMIT).

References

[1] C. Aggarwal and P. Yu. "Finding Generalized Projected Clusters in High Dimensional Space". In *Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'00)*, 2000.

[2] C. C. Aggarwal and C. Procopiuc. "Fast Algorithms for Projected Clustering". In *Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'99)*, 1999.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications". In *Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'98)*, 1998.

[4] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. "OPTICS: Ordering Points to Identify the Clustering Structure". In *Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'99)*, 1999.

[5] C. Baumgartner, C. Böhm, D. Baumgartner, G. Marini, K. Weinberger, B. Olgemöller, B. Liebl, and A. A. Roscher. "Supervised machine learning techniques for the classification of metabolic disorders in newborns". *Bioinformatics*, 2004. in press.

[6] M. Dash, K. Choi, P. Scheuermann, and H. Liu. "Feature Selection for Clustering – A Filter Solution". In *Proc. IEEE Int. Conf. on Data Mining (ICDM'02)*, 2002.

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, 1996.

[8] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Academic Press, 2001.

[9] K. Kailing, H.-P. Kriegel, P. Kröger, and S. Wanka. "Ranking Interesting Subspaces for Clustering High Dimensional Data". In *Proc. 7th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'03)*, 2003.

[10] C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. M. Murali. "A Monte Carlo Algorithm for Fast Projective Clustering". In *Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'02)*, 2002.

[11] P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher. "Comprehensive Identification of Cell Cycle-Regulated Genes of the Yeast Saccharomyces Cerevisiae by Microarray Hybridization.". *Molecular Biolology of the Cell*, 9:3273–3297, 1998.