Proc. 8th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD), Sydney, AU, 2004

Clustering Multi-Represented Objects with Noise

Karin Kailing, Hans-Peter Kriegel, Alexey Pryakhin, and Matthias Schubert

Institute for Computer Science
University of Munich
Oettingenstr. 67, 80538 Munich, Germany
{kailing|kriegel|pryakhin|schubert}@dbs.informatik.uni-muenchen.de

Abstract. Traditional clustering algorithms are based on one representation space, usually a vector space. However, in a variety of modern applications, multiple representations exist for each object. Molecules for example are characterized by an amino acid sequence, a secondary structure and a 3D representation. In this paper, we present an efficient density-based approach to cluster such multi-represented data, taking all available representations into account. We propose two different techniques to combine the information of all available representations dependent on the application. The evaluation part shows that our approach is superior to existing techniques.

1 Introduction

In recent years, the research community spent a lot of attention to clustering resulting in a large variety of different clustering algorithms [1]. However, all those methods are based on one representation space, usually a vector space of features and a corresponding distance measure. But for a variety of modern applications such as biomolecular data, CAD- parts or multi-media files mined from the internet, it is problematic to find a common feature space that incorporates all given information. Molecules like proteins are characterized by an amino acid sequence, a secondary structure and a 3D representation. Additionally, protein databases such as Swissprot [2] provide meaningful text descriptions of the stored proteins. In CAD-catalogues, the parts are represented by some kind of 3D model like Bezier curves, voxels or polygon meshes and additional textual information like descriptions of technical and economical key data. We call this kind of data *multi-represented data*, since any data object might provide several different representations that may be used to analyze it.

To cluster multi-represented data using the established clustering methods would require to restrict the analysis to a single representation or to construct a feature space comprising all representations. However, the restriction to a single feature space would not consider all available information and the construction of a combined feature space demands great care when constructing a combined distance function.

In this paper, we propose a method to integrate multiple representations directly into the clustering algorithm. Our method is based on the density-based clustering algorithm DBSCAN [3] that provides several advantages over other algorithms, especially when analyzing noisy data. Since our method employs a separated feature space for each representation, it is not necessary to design a new suitable distance measure for each new application. Additionally, the handling of objects that do not provide all possible representations is integrated naturally without defining dummy values to compensate for the missing representations. Last but not least, our method does not require a combined index structure, but benefits from each index that is provided for a single representation. Thus, it is possible to employ highly specialized index structures and filters for each representation.We evaluate our method for two example applications. The first is a data set consisting of protein sequences and text descriptions. Additionally, we applied our method to the clustering of images retrieved from the internet. For this second data set, we employed two different similarity models.

The rest of the paper is organized as follows. After this introduction, we present related work. Section 3 formalizes the problem and introduces our new clustering method. In our experimental evaluation that is given in section 4, we introduce a new quality measure to judge the quality of a clustering with respect to a reference clustering and display the results achieved by our method in comparison with the other mentioned approaches. The last section summarizes the paper and presents some ideas for future research.

2 Related Work

There are several problems that are closely related to the clustering of multi-represented data. Data mining of multi-instance objects [4] is based on the precondition that each data object might be represented by more than one instance in a common data space. However, all instances that are employed are elements of the same data space and multi-instance objects were predominantly treated with respect to classification not to clustering.

A similar setting to the clustering of multi-represented objects is the clustering of heterogenous or multi-typed objects [5, 6] in web mining. In this setting, there are also multiple databases each yielding objects in a separated data space. Each object within these data spaces may be related to an arbitrary amount of data objects within the other data spaces. The framework of reinforcement clustering employs an iterative process based on an arbitrary clustering algorithm. It clusters one dedicated data space while employing the other data spaces for additional information. It is also applicable for multi-represented objects. However, due to its dependency on the data space for which the clustering is started, it is not well suited to solve our task. Since to the best of our knowledge reinforcement clustering is the only other clustering algorithm directly applicable to multi-represented objects, we use it for comparison in our evaluation section.

Our approach is based on the formal definitions of density-connected sets underlying the algorithm DBSCAN [3]. Based on two input parameters (ε and

k), DBSCAN defines dense regions by means of core objects. An object $o \in DB$ is called *core object*, if its ε-neighborhood contains at least k objects. Usually clusters contain several core objects located inside a cluster and border objects located at the border of the cluster. In addition, objects within a clusters must be “density-connected”. DBSCAN is able to detect arbitrarily shaped clusters by one single pass over the data. To do so, DBSCAN uses the fact, that a density-connected cluster can be detected by finding one of its core-objects o and computing all objects which are density-reachable from o. The correctness of DBSCAN can be formally proven (cf. lemmata 1 and 2 in [3], proofs in [7]).

3 Clustering Multi-Represented Objects

Let DB be a database consisting of n objects. Let $R := \{R_1, ..., R_m\}$ be the set of different representations existing for objects in DB. Each object $o \in DB$ is therefore described by maximally m different representations, i.e. $o := \{R_1(o), R_2(o), ..., R_m(o)\}$. If all different representations exist for o, than $|o| = m$, else $|o| < m$. The distance function is denoted by $dist$. We assume that $dist$ is symmetric and reflexive. In the following, we call the ε_i-neighborhood of an object o in one special representation R_i its local ε-neighborhood w.r.t. R_i.

Definition 1 (local ε_i-neighborhood w.r.t R_i).
Let $o \in DB$, $\varepsilon_i \in I\!R^+$, $R_i \in R$, $dist_i$ the distance function of R_i. The local ε_i-neighborhood w.r.t. R_i *of o, denoted by $\mathcal{N}_{\varepsilon_i}^{R_i}(o)$, is defined by*

$$\mathcal{N}_{\varepsilon_i}^{R_i}(o) = \{x \in DB \mid dist_i(R_i(o), R_i(x)) \leq \varepsilon_i\}.$$

Note that ε_i can be chosen optimally for each representation. The simplest way of clustering multi-represented objects, is to select one representation R_i and cluster all objects according to this representation. However, this approach restricts data analysis to a limited part of the available information and does not use the remaining representations to find a meaningful clustering. Another way to handle multi-represented objects is to combine the different representations and use a combined distance function. Then any established clustering algorithm can be applied. However, it is very difficult to construct a suitable combined distance function that is able to fairly weight each representation and handle missing values. Furthermore, a combined feature space, does not profit from specialized data access structures for each representation.

The idea of our approach is to combine the information of all different representations as early as possible, i.e. during the run of the clustering algorithm, and as late as necessary, i.e. after using the different distance functions of each representation. To do so, we adapt the core object property proposed for DBSCAN. To decide whether an object is a core object, we use the local ε-neighborhoods of each representation and combine the results to a global neighborhood. Therefore, we must adapt the predicate direct density-reachability proposed for DBSCAN. In the next two subsections, we will show how we can use the concepts of union and intersection of local neighborhoods to handle multi-represented objects.

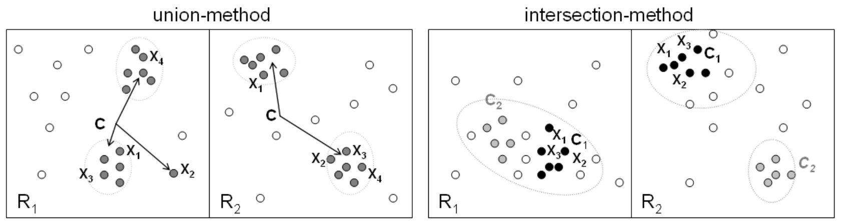

Fig. 1. The left figure displays local clusters and a noise object that are aggregated to a multi-represented cluster **C**. The right figure illustrates, how the intersection-method divides a local clustering into clusters C_1 and C_2.

3.1 Union of Different Representations

This variant is especially useful for sparse data. In this setting, the clusterings in each single representation will provide several small clusters and a large amount of noise. Simply enlarging ε would relief the problem, but on the other hand, the separation of the clusters would suffer. The union-method assigns objects to the same cluster, if they are similar in at least one of the representations. Thus, it keeps up the separation of local clusters, but still overcomes the sparsity. If the object is placed in a dense area of at least one representation, it is still a core object regardless of how many other representations are missing. Thus, we do not need to define dummy values. The left part of figure 1 illustrates the basic idea. We adapt some of the definitions of DBSCAN to capture our new notion of clusters. To decide whether an object o is a union core object, we unite all local ε_i-neighborhoods and check whether there are enough objects in the global neighborhood, i.e. whether the global neighborhood of o is dense.

Definition 2 (union core object).
Let $\varepsilon_1, \varepsilon_2, ..., \varepsilon_m \in I\!R^+$, $k \in I\!N$. An object $o \in DB$ is called union core object*, denoted by $\text{CoreU}^k_{\varepsilon_1,..,\varepsilon_m}(o)$, if the union of all local ε-neighborhoods contains at least k objects, formally:*

$$\text{CoreU}^k_{\varepsilon_1,..,\varepsilon_m}(o) \Leftrightarrow \Big| \bigcup_{R_i(o)\in o} \mathcal{N}^{R_i}_{\varepsilon_i}(o) \Big| \geq k.$$

Definition 3 (direct union-reachability).
Let $\varepsilon_1, \varepsilon_2, .., \varepsilon_m \in I\!R^+$, $k \in I\!N$. An object $p \in DB$ is directly union-reachable *from $q \in DB$ if q is a union core object and p is an element of at least one local $\mathcal{N}^{R_i}_{\varepsilon_i}(q)$, formally:*

$$\text{DirReachU}^k_{\varepsilon_1,..,\varepsilon_m}(q,p) \Leftrightarrow \text{CoreU}^k_{\varepsilon_1,..,\varepsilon_m}(q) \wedge \exists\, i \in \{1,..,m\} : R_i(p) \in \mathcal{N}^{R_i}_{\varepsilon_i}(q).$$

The predicate direct union-reachability is obviously symmetric for pairs of core objects, because the $dist_i$ are symmetric distance functions. Thus, analogously to DBSCAN reachability and connectivity can be defined.

3.2 Intersection of Different Representations

The intersection method is well suited for data containing unreliable representations, i.e. there is a representation, but it is questionable, whether it is a good description of the object. In those cases, the intersection-method requires that a cluster should contain only objects which are similar according to all representations. Thus, this method is useful, if all different representations exist, but the derived distances do not adequately mirror the intuitive notion of similarity. The intersection-method is used to increase the cluster quality by finding purer clusters.

To decide, whether an object o is an intersection core object, we examine, whether o is a core object in each involved representation. Of course, we use different ε-values for each representation to decide, whether locally there are enough objects in the ε-neighborhood. The parameter k is used to decide, whether globally there are still enough objects in the ε-neighborhood, i.e. the intersection of all local neighborhoods contains at least k objects.

Definition 4 (intersection core object).
Let $\varepsilon_1, \varepsilon_2, ..., \varepsilon_m \in I\!R^+$, $k \in I\!N$. An object $o \in DB$ is called intersection core object*, denoted by $\text{CoreIS}^k_{\varepsilon_1,..,\varepsilon_m}(o)$, if the intersection of all its local ε_i-neighborhoods contain at least k objects, formally:*

$$\text{CoreIS}^k_{\varepsilon_1,..,\varepsilon_m}(o) \Leftrightarrow |\bigcap_{i=1,..,m} \mathcal{N}^{R_i}_{\varepsilon_i}(o)| \geq k.$$

Using this new property, we can now define direct intersection-reachability in the following way:

Definition 5 (direct intersection-reachability).
Let $\varepsilon_1, \varepsilon_2, ..., \varepsilon_m \in I\!R^+$, $k \in I\!N$. An object $p \in DB$ is directly intersection-reachable *from $q \in DB$ if q is an intersection core object and p is an element of all local $\mathcal{N}_\varepsilon(q)$, formally:*

$$\text{DirReachIS}^k_{\varepsilon_1,..,\varepsilon_m}(q,p) \Leftrightarrow \text{CoreIS}^k_{\varepsilon_1,..,\varepsilon_m}(q) \wedge \forall i = 1,..,m : R_i(p) \in \mathcal{N}^{R_i}_{\varepsilon_i}(q) .$$

Again, reachability and connectivity can be defined analogously to DBSCAN. The right part of figure 1 illustrates the effects of this method.

3.3 Determination of Density Parameters

In [3], a heuristic is presented to determine the ε-value of the "thinnest" cluster in the database. This heuristic is based on a diagram that represents sorted *knn*-distances of all given objects. In the case of multi-represented objects, we have to choose ε for each dimension separately, whereas k can be chosen globally. A user determines a value for global k. The system computes the *knn*-distance diagrams for the given global k (one diagram for every representation). The user has to choose a so-called border object o for each representation. The ε for the

i-th representation is given by the knn-distance of the border object of R_i. Let us note that this method still allows a certain range of ε-values to be chosen. The selection should mirror the different requirements of the proposed methods. For the union method, it is more advisable to chose a lower or conservative value, since its characteristic demands that the elements of the local ε-neighborhood should really be similar. For the intersection-method, the ε-value should be selected progressively, i.e. at the upper rim of the range. This selection reflects that the objects of a cluster need not be too similar for a single representation, because it is required that they are similar with respect to all representations.

4 Performance Evaluation

To demonstrate the capability of our method, we performed a thorough experimental evaluation for two types of applications. We implemented the proposed clustering algorithm in Java 1.4. All experiments were processed on a work station with a 2.6 GHz Pentium IV processor and 2 GB main memory.

4.1 Deriving Meaningful Groupings in Protein Databases

The first set of experiments was performed on protein data that is represented by amino-acid sequences and text descriptions. Therefore, we employed entries of the Swissprot protein database [2] belonging to 5 functional groups (cf. Table 1) and transformed each protein into a pair of feature vectors. Each amino acid sequence was mapped into a 436 dimensional feature space. The first 400 features are 2-grams of successive amino-acids. The last 36 dimensions are 2-grams of 6 exchange groups that the single amino-acids belong to [8]. To compare the derived feature vectors, we employed Euclidian distance. To process text documents, we rely on projecting the documents into the feature space of relevant terms. Documents are described by a vector of term frequencies weighted by the inverse document frequency (TFIDF) [9]. We chose 100 words of medium frequency as relevant terms and employed cosine distance to compare the TFIDF-vectors. Since Swissprot entries provide a unique mapping to the classes of Gene Ontology [10], a reference clustering for the selected proteins was available. Thus, we are able to measure a clustering of Swissprot entries by the degree it reproduces the class structure provided by Gene Ontology.

To have an exact measure for this degree, we employed the class entropy in each cluster. However, there are two effects that have to be considered to obtain a

Table 1. Description of the protein data sets.

	Set 1	**Set 2**	**Set 3**	**Set 4**	**Set 5**
Name	Isomerase	Lyase	Signal Transducer	Oxidoreductase	Transferase
Classes	16	35	39	49	62
Objects	501	1640	2208	3399	4086

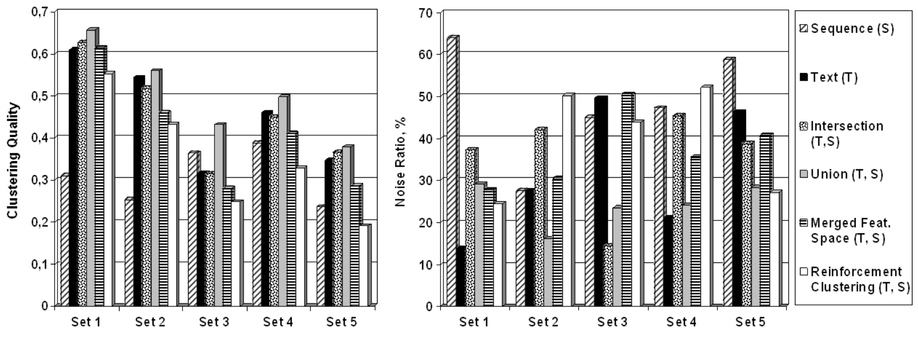

Fig. 2. Clustering quality and noise ratio.

fair measure of a clustering with noise. First, a large cluster of a certain entropy should contribute more to the overall quality of the clustering than a rather small cluster providing the same quality. The second effect is that a clustering having a 5 % noise ratio should be ranked higher than a clustering having the same average entropy for all its clusters, but contains 50 % noise.

To consider both effects we propose the following quality measure for comparing different clusterings with respect to a reference clustering.

Definition 6. *Let O be the set of data objects, let $C = \{C_i | C_i \subset O\}$ be the set of clusters and let $K = \{K_i | K_i \subset O\}$ be the reference clustering of O. Then we define:*

$$Q_K(C) = \sum_{C_i \in C} \frac{|C_i|}{|O|} \cdot (1 + entropy_K(C_i))$$

where $entropy_K(C_i)$ denotes the entropy of cluster C_i with respect to K.

The idea is to weight every cluster by the percentage of the complete data objects being part of it. Thus, smaller clusters are less important than larger ones and a clustering providing an extraordinary amount of noise can contribute only the percentage of clustered objects to the quality. Let us note that we add 1 to the cluster entropies. Therefore, we measure the reference clustering K with the quality score of 1 and a worst case clustering – e.g. no clusters are found at all– with the score of 0. To relate the quality of the clustering achieved by our methods to the results of former methods, we compared it to 4 alternative approaches. First, we clustered text and sequences separately using only one of the representations. A second approach combines the features of both representations into a common feature space and employs the cosine distance to relate the resulting feature vectors. As the only other clustering method that is able to handle multi-represented data, we additionally compared reinforcement clustering using DBSCAN as underlying cluster algorithm. For reinforcement clustering, we ran 10 iterations and tried several values of the weighting parameter α. The local ε-parameters were selected as described above and we chose

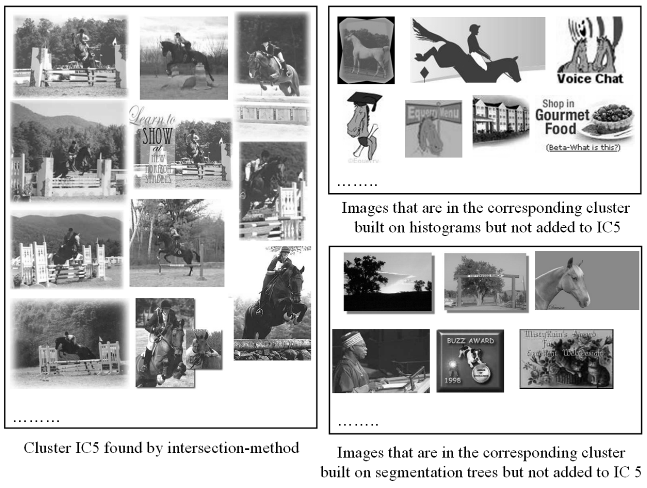
Voice Chat
Shop in Gourmet Food
(Beta-What is this?)
BUZZ AWARD
1998
………
Cluster IC5 found by intersection-method
Images that are in the corresponding cluster built on histograms but not added to IC5
Images that are in the corresponding cluster built on segmentation trees but not added to IC 5

Fig. 3. Example of an image cluster. The left rectangle contains images clustered by the intersection-method. The right rectangles display additional images that were grouped with the corresponding cluster when clustering the images with respect to a single representation.

$k = 2$. To consider the different requirements of both methods, for each data set a progressive and a conservative ε-value was determined. All approaches were run for both settings and the best results are displayed.

The left diagram of figure 2 displays the derived quality for those 4 methods and the two variants of our method. In all five test sets, the union-method using conservative ε-values outperformed any of the other algorithms. Furthermore, the noise ratio for each data set was between 16% and 28% (cf. figure 2, right), indicating that the main portion of the data objects belongs to some cluster. The intersection method using progressive ε-parameters performed comparably well, but was to restrictive to overcome the sparseness of the data as good as the union-method.

4.2 Clustering Images by Multiple Representations

Clustering image data is a good example for the usefulness of the intersection-method. A lot of different similarity models exists for image data, each having its own advantages and disadvantages. Using for example text descriptions of images, one is able to cluster all images related to a certain topic, but these images must not look alike. Using color histograms instead, the images are clustered according to the distribution of color in the image. But as only the color information is taken into account a green meadow with some flowers and a green billiard table with some colored shots on it, can of course not be distinguished by this similarity model. On the other hand, a similarity model taking content information into account might not be able to distinguish images of different colors.

Our intersection approach is able to get the best out of all these different types of representations. Since the similarity in one representation is not really sound, the intersection-method is well-suited to find clusters of better quality for this application. For our experiments, we used two different representations. The first representation was a 64-dimensional color histogram. In this case, we used the weighted distance between those color histograms, represented as a quadratic form distance function as described for example in [11]. The second representation were segmentation trees. An image was first divided into segments of similar color by a segmentation algorithm. In a second step, a tree was created from those segments by iteratively applying a region-growing algorithm which merges neighboring segments, if their colors are alike. In [12] an efficient technique is described to compute the similarity between two such trees using filters for the complex edit-distance measure.

As we do not have any class labels to measure the quality of our clustering, we can only describe the results we achieved. In general, the clusters we got using both representations were more accurate than the clusters we got using each representation separately. Of course, the noise ratio increased for the intersection-method. Due to space limitations we only show one sample cluster of images we found with the intersection-method (see Figure 3). Using this method, very similar images are clustered together. When clustering each single representation, a lot of additional images were added to the corresponding cluster. As one can see, using the intersection-method only the most similar images of both representations still belong to the cluster.

5 Conclusions

In this paper, we discussed the problem of clustering multi-represented objects. A multi-represented object is described by a set of representations where each representation belongs to a different data space. Contrary to existing approaches our proposed method is able to cluster this kind of data using all available representations without forcing the user to construct a combined data space. The idea of our approach is to combine the information of all different representations as early as possible and as late as necessary. To do so, we adapted the core object property proposed for DBSCAN. To decide whether an object is a core object, we use the local ε-neighborhoods of each representation and combine the results to a global neighborhood. Based on this idea, we proposed two different methods for varying applications. For sparse data, we introduced the union-method that assumes that an object is a core object, if k objects are found within the union of its local ε-neighborhoods. Respectively, we defined the intersection-method for data where each local representation yields rather big and unspecific clusters. Therefore, the intersection-method requires that at least k objects are within the intersection of all local ε-neighborhoods of a core object. In our experimental evaluation, we introduced an entropy based quality measure that compares a given clustering with noise to a reference clustering. Employing this quality measure, we demonstrated that the union method was most suitable to overcome the sparsity of a given protein data set. To demonstrate the ability of the intersection method to increase the cluster quality, we applied it to a set of images using two different similarity models.

For future work, we plan to examine applications providing more than two representations. We are especially interested, in clustering proteins with respect to all of the mentioned representations. Another interesting challenge is to extend our method to an multi-instance and multi-representation clustering. In this setting each object may be represented by several instances in some of the representations.

References

1. Han, J., Kamber, M.: "Data Mining: Concepts and Techniques". Morgan Kaufman (2001)
2. Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.C., Estreicher, A., Gasteiger, E., Martin, M., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., Schneider, M.: "The SWISS-PROT Protein Knowledgebase and its Supplement TrEMBL in 2003". Nucleic Acid Research **31** (2003) 365–370
3. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: Proc. KDD'96, Portland, OR, AAAI Press (1996) 291–316
4. Weidmann, N., Frank, E., Pfahringer, B.: "A Two-Level Learning Method for Generalized Multi-instance Problems". In: Proc. ECML 2003, Cavtat-Dubrovnik,Cr. (2003) 468–479
5. Wang, J., Zeng, H., Chen, Z., Lu, H., Tao, L., Ma, W.: "ReCoM: reinforcement clustering of multi-type interrelated data objects. 274-281". In: Proc. SIGIR 2003, July 28 - August 1, Toronto, CA, ACM (2003) 274–281
6. Zeng, H., Chen, Z., Ma, W.: "A Unified Framework for Clustering Heterogeneous Web Objects". In: Proc. 3rd WISE 2002, 12-14 December, Singapore, IEEE Computer Society (2002) 161–172
7. Sander, J., Ester, M., Kriegel, H.P., Xu, X.: "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications". In: Data Mining and Knowledge Discovery, An International Journal, Kluwer Academic Publishers (1998) 169–194
8. Deshpande, M., Karypis, G.: "Evaluation of Techniques for Classifying Biological Sequences". In: Proc. PAKDD'02. (2002) 417–431
9. Salton, G.: "Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer". Addison-Wesley (1989)
10. Consortium, T.G.O.: "Gene Ontology: Tool for the Unification of Biology". Nature Genetics **25** (2000) 25–29
11. Hafner, J., Sawhney, H., W., E., Flickner, M., Niblack, W.: Efficient color histogram indexing for quadratic form distance functions. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **17 (7)** (1995) 729–736
12. Kailing, K., Kriegel, H.P., Schönauer, S., Seidl, T.: Efficient similarity search for hierachical data in large databases. In: to appear in Proc. EDBT 2004. (2004)