10th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'06), Singapore, 2006

An EM-Approach for Clustering Multi-Instance Objects

Hans-Peter Kriegel, Alexey Pryakhin, and Matthias Schubert

Institute for Informatics
University of Munich
D-80538 Munich, Germany
{kriegel,pryakhin,schubert}@dbs.ifi.lmu.de

Abstract. In many data mining applications the data objects are modeled as sets of feature vectors or multi-instance objects. In this paper, we present an expectation maximization approach for clustering multi-instance objects. We therefore present a statistical process that models multi-instance objects. Furthermore, we present M-steps and E-steps for EM clustering and a method for finding a good initial model. In our experimental evaluation, we demonstrate that the new EM algorithm is capable to increase the cluster quality for three real world data sets compared to a k-medoid clustering.

1 Introduction

In modern data mining applications, the complexity of analyzed data objects is increasing rapidly. Molecules are analyzed more precisely and with respect to all of their possible spatial conformations [1]. Earth observation satellites are able to take images with higher resolutions and in a variety of spectra which was not possible some years before. Data mining started to analyze complete websites instead of single documents [2]. All of these application domains are examples for which the complexity demands a richer object representation than single feature vectors. Thus, for these application domains, an object is often described as a set of feature vectors or a multi-instance (MI) object. For example, a molecule can be represented by a set of feature vectors where each vector describes one spatial conformation or a website can be analyzed as a set of word vectors corresponding to its HTML documents.

As a result the research community started to develop techniques for multi-instance learning that where capable to analyze multi-instance objects. One of the first publications in this area [1, 3] was focussed to a special task called multi-instance learning. In this task the appearance of one positive instance within a multi-instance object is sufficient to indicate that the object belongs to the positive class. Besides classical multi-instance learning, some approaches like [4, 5] aim at more general problems. However, all of the mentioned approaches are based on a setting having a set of labeled bags to train a learning algorithms.

In this paper, we focus on clustering unlabeled sets of feature vectors. To cluster those objects, the common approach so far is to select some distance

measures for point sets like [6, 7] and then apply a distance-based clustering algorithm e.g. k-medoid methods like CLARANS [8] or a density-based algorithm like DBSCAN[9]. However, this approach does not yield expressive cluster models. Depending on the used algorithm, we might have some representative for some cluster, but we do not have a good model for describing the mechanism behind this clustering. To overcome this problem, we will refer to the model of multi-instance objects that was introduced in [5] stating that a multi-instance object of a particular class (or in our problem each cluster) needs to provide instances belonging to a certain concept or several concepts. We will adapt this view of multi-instance objects to clustering. Therefore, we propose a statistical model that is based on 2 steps. In the first step, we use a standard EM Clustering algorithm on the union set of all multi-instance objects. Thus, we determine a mixture model describing the instances of all multi-instance objects. Assuming that each of the found clusters within each mixture model corresponds to some valid concept, we now can derive distributions for the clustering of multi-instance objects. For this second step, we assume that a multi-instance object containing k instances can be modeled as k draws from the mixture model over the instances. Thus, each cluster of multi-instance objects is described by a distribution over the instance clusters derived in the first step and some prior probability. For example, for the classical multi-instance learning task, it can be expected that there is at least one instance cluster that is very unlikely to appear in the multi-instance clusters corresponding to the negative bags.

The rest of the paper is organized as following: In section 2, we will survey previous work in data mining with multi-instance objects and give a brief introduction to EM clustering. Section 3 will describe our statistical model for multi-instance data. In section 4, this model is employed for EM clustering. To demonstrate the usefulness of our approach, section 5 contains the results on several real world data sets. Section 6 concludes the paper with a summary and directions for future work.

2 Related Work

Data Mining in multi-instance objects has so far been predominantly examined in the classification section. In [1] Dietterich et al. defined the problem of multi-instance learning for drug prediction and provided a specialized algorithm to solve this particular task by learning axis parallel rectangles. In the following years, new algorithms increasing the performance for this special task were introduces [3]. In [5] a more general method for handling multi-instance objects was introduced that is applicable for a wider variety of multi-instance problems. This model considers several concepts for each class and requires certain cardinalities for the instances belonging to the concepts in order to specify a class of MI objects. Additionally, to this model [10] proposes more general kernel functions for MI comparing MI objects.

For clustering multi-instance objects, it is possible to use distance functions for sets of objects like [6, 7]. Having such a distance measure, it is possible to

cluster multi-instance objects with k-medoid methods like PAM and CLARANS [11] or employ density-based clustering approaches like DBSCAN [9]. Though this method yields the possibility to partition multi-instance objects into clusters, the clustering model consists of representative objects in the best case. Another problem of this approach is that the selection of a meaningful distance measure has an important impact of the resulting clustering. For example, netflow-distance [7] demands that all instances within two compared objects are somehow similar, whereas for the minimal Hausdorff [12] distance the indication of similarity is only dependent on the closest pair.

In this paper, we introduce an algorithm for clustering multi-instance objects that optimizes probability distributions to describe the data set. Part of this work is based on expectation maximization (EM) clustering for ordinary feature vectors using Gaussians. Details about this algorithm can be found in [13]. In [14], a method for producing a good initial mixture is presented which is based on multiple sampling. It is empirically shown that using this method, the EM algorithm achieves accurate clustering results.

3 A Statistical Model for Multi-Instance Objects

In this section, we will introduce our model for multi-instance clustering. Therefore, we will first of all define the terms instance and multi-instance (MI) object.

Definition 1 (instance and MI object). *Let F be a feature space. Then, $i \in F$ is called an instance in F. A multi-instance (MI) object o in F is given by an arbitrary sized set of instances $o = i_1, .., i_k$ with $i_j \in F$. To denote the unique MI object an instance i belongs to, we will write $MiObj(i)$.*

To cluster multi-instance objects using an EM approach, we first of all need a statistical process that models sets of multi-instance objects. Since multi-instance objects consist of single instances in some feature space, we begin with modeling the data distribution in the feature space of instances. Therefore, we first of all define the instance set of a set of multi-instance objects:

Definition 2 (Instance Set). *Given a database DB of multi-instance Objects $o = i_1, \ldots, i_k$, the corresponding instance set $I_{DB} = \bigcup_{DB} o$ is the union of all multi-instance objects.*

To model the data distribution in the instance space, we assume a mixture model of k independent statistical processes. For example, an instance set consisting of feature vectors could be described by a mixture of Gaussians.

Definition 3 (Instance Model). *Let DB be a data set consisting of multi-instance objects o and let I_{DB} be its instance set. Then, an instance model IM for DB is given by a mixture model of k statistical processes that can be described by a prior probability $Pr[k_j]$ for each component k_j and the necessary parameters for the process corresponding to k_j, e.g. a mean vector μ_j and co-variance matrix M_j for Gaussian processes.*

After describing the instance set, we can now turn to the description of multi-instance objects. Our solution is based on the idea of modeling a cluster of multi-instance objects as a multinomial distribution over the components of the mixture model of instances. For each instance and each concept, the probability that the instance belongs to this concept is considered as result of one draw. If the number n of instances within an object o is considered to be important as well, we can integrate this into our model as well by considering some distribution over the number of draws, e.g. a binomial distribution. To conclude, a mixture model of multi-instance clusters can be described by a set of multinomial distributions over the components of a mixture model of instances. A multi-instance object is thus derived in the following way:

1. Select a multi-instance cluster c_i w.r.t. some prior distribution over the set of all clusters C.
2. Derive the number of instances n within the multi-instance object w.r.t some distribution depending on the chosen cluster c_i.
3. Repeat n-times:
   (a) Select some model component k_j within the mixture model of instances w.r.t. the multi-instance cluster specific distribution.
   (b) Generate an instance, w.r.t. to the distribution corresponding to component k_j.

Formally, the underlying model for multi-instance data sets can be defined as follows:

Definition 4 (Multi-Instance Model). *A multi-instance model M over the instance model IM is defined by a set C of l processes over I_{DB}. Each of these processes c_i is described by a prior probability $Pr[c_i]$, a distribution over the number of instances in the bag $Pr[Card(o)\ |c_i]$ and an conditional probability describing the likelihood that a multi-instance object o belonging to process c_i contains an instance belonging to the component $k_l \in IM$. The probability of an object o in the model M is calculated as following:*

$$Pr[o] = \sum_{c_i \in C} Pr[c_i] \cdot \ Pr[Card(o)|c_i] \cdot \prod_{i \in o} \prod_{k \in MI} Pr[k|c_i]^{Pr[k|i]}$$

The conditional probability of process c_i under the condition of a given multi-instance object o can be calculated by:

$$Pr[c_i|o] = \frac{1}{Pr[o]} \cdot Pr[c_i] \cdot \ Pr[Card(o)|c_i] \cdot \prod_{i \in o} \prod_{k \in MI} Pr[k|c_i]^{Pr[k|i]}$$

Let us note that the occurrence of an instance within the data object is only dependent on the cluster of instances it is derived from. Thus, we do not assume any dependencies between the instances of the same objects. Another important characteristic of the model is that we assume the same set of instance clusters for all multi-instance clusters. Figure 3 displays an example of a two dimensional multi-instance data set corresponding to this model. This assumption leads to the following 3 step approach for multi-instance EM clustering.

4 EM-Clustering for Multi-Instance Objects

After introducing a general statistical process for multi-instance objects, we will now introduce an EM algorithm that fits the distribution parameters to a given set of multi-instance objects. Our method works in 3 steps:

1. Derive a Mixture Model for the Instance Set.
2. Calculate a start partitioning.
3. Use the new EM algorithm to optimize the start partitioning.

4.1 Generating a Mixture Model for the Instance Set

To find a mixture of the instance space, we can employ a standard EM approach as proposed in section 2. For general feature vectors, we can describe the instance set as a mixture of Gaussians. If the feature space is sparse using a mixture of multinomial processes usually provides better results. If the number of clusters in the instance is already known, we can simply employ EM clustering. However, if we do not know how many clusters are hidden within the instance set, we need to employ a method for determining a suitable number of processes like [15].

4.2 Finding a Start Partitioning of Multi-Instance Objects

After deriving a description of the instance space, we now determine a good start partitioning for the final clustering step. A good start partitioning is very important for finding a good cluster model. Since EM algorithms usually do not achieve a global maximum likelihood, a suitable start partitioning has an important impact on both, the likelihood of the cluster and the runtime of the algorithm. The versions for EM in ordinary feature spaces often use k-means clustering for finding a suitable start partitioning. However, since we cluster sets of instances instead of single instances, we cannot use this approach directly.

To overcome this problem, we proceed as follows. For each multi-instance object we determine a so-called confidence summary vector in the following way.

Definition 5 (Confidence Summary Vector). *Let IM be an instance model over database DB containing k processes and let o be a multi-instance object. Then the confidence summary vector $\overrightarrow{csv}(o)$ of o is a k dimensional vector that is calculated as follows:*

$$csv_j(o) = \sum_{i \in o} Pr[k_j] \cdot Pr[i|k_j]$$

After building the confidence summary vector for each object, we can now employ k-means to cluster the multi-instance objects. Though the resulting clustering might not be optimal, the objects within one cluster should yield similar distributions over the components of the underlying instance model.

4.3 EM for Clustering Multi-Instance Objects

In this final step, the start partitioning for the data set is optimized using the EM algorithm. We therefore describe a suitable expectation and maximization step and then employ an iterative method. The likelihood of the complete model M can be calculated by adding up the log-likelihoods of the occurrence of each data object in each clusters. Thus, our model is (locally) optimal if we obtain a maximum for the the following log-likelihood term.

Definition 6 (Log-Likelihood for M).

$$E(M) = \sum_{o \in DB} \log \sum_{c_i \in M} Pr[c_i|o]$$

To determine $Pr[c_i|o]$, we proceed as mentioned in definition 4. Thus, we can easily calculate $E(M)$ in the expectation step for a given set of distribution parameters and an instance model. To improve the distribution parameters, we employ the following updates to the distribution parameters in the maximization step:

$$W_{c_i} = Pr[c_i] = \frac{1}{Card(DB)} \sum_{o \in DB} Pr[c_i|o]$$

where W_{c_i} denotes the prior probability of a cluster of multi-instance objects.

To estimate the number of instances contained in an MI object belonging to cluster c_i, we can employ a binomial distribution determined by the parameter l_{c_i}. The parameters are updated as follows:

$$l_{c_i} = \frac{\sum_{o \in DB} Pr[c_i|o] \cdot Card(o)}{Card(DB)} \cdot \frac{1}{MAXLENGTH}$$

where $MAXLENGTH$ is the maximum number of instances for any MI object in the database.

Finally, to estimate the relative number of instances drawn from concept k_j for MI objects belonging to cluster c_i, we derive the parameter updates in the following way:

$$P_{k_j,c_i} = Pr[k_j|c_i] = \frac{\sum_{o \in DB} (Pr[c_i|o] \cdot \sum_{u \in o} Pr[u|k_j])}{\sum_{o \in DB} Pr[c_i|o]}$$

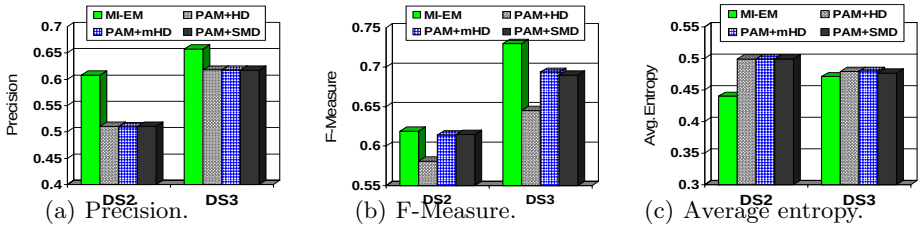
Using these update steps, the algorithm is terminated after the improvement of $E(M)$ is less than a given value σ. Since the last step of our algorithm is a modification of EM clustering based on multinomial processes, our algorithm always converges against a local maximum value for $E(M)$.

5 Evaluation

All algorithms are implemented in Java 1.5. The experiments described below are carried out on a work station that is equipped with two 1.8 GHz Opteron processors and 8 GB main memory.

	Data Set 1 (DS1)	**Data Set 2 (DS2)**	**Data Set 3 (DS3)**
Name	Brenda	MUSK 1	MUSK 2
Number of MI-Objects	6082	92	102
Average Number of Instances per MI-Object	1.977	5.2	64.7
Number of MI-Object classes	6	2	2

Table 1. Details of the test environments



(a) Precision. (b) F-Measure. (c) Average entropy.

Fig. 1. Effectiveness evaluation on DS2 and DS3 where no. of clusters is 2.

Our experiments were performed on 3 different real world data sets. The properties of each test bed are illustrated in Table 1. The Brenda data set contains of enzymes taken from the protein data bank (PDB) [1]. Each enzyme comprises several chains given by amino acid sequences. In order to derive feature vectors from the amino acid sequences, we employed the approach described in [16]. The basic idea is to use local (20 amino acids) and global (6 exchange groups) characterization of amino acid sequences. In order to construct a meaningful feature space, we formed all possible 1-grams for each kind of characteristic. This approach provided us with 26 dimensional histograms for each chain. To obtain the class labels for each enzyme we used a mapping from PDB to the enzyme class numbers from the comprehensive enzyme information system BRENDA [2].

MUSK 1 and MUSK 2 data sets come from UCI repository [17] and describe a set of molecules. The MI-objects in MUSK 1 and MUSK 2 data sets are judged by human experts to be in musks or non-musks class. The feature vectors of MUSK data sets have 166 numerical attributes that describe these molecules depending on the exact shape or conformation of the molecule.

To measure the effectiveness, we considered the agreement of the calculated clusterings to the given class systems. To do so, we calculated three quality measures namely precision, F-measure and average entropy. In order to calculate the precision and F-Measure, we proceeded as follows. For each cluster c_i found by a clustering algorithm, its class assignment $Class(c_i)$ is determined by the class label of objects belonging to c_i that are in the majority. Then, we calculated

[1] http://www.rcsb.org/pdb/
[2] http://www.brenda.uni-koeln.de/

the Precision within all clusters w.r.t. the determined class assignments by using the following formulas.

$$Precision = \frac{\sum_{c_i \in C} Card(\{o | (c_i = \arg\max_{c_j \in C} Pr[c_j|o]) \wedge Class(o) = Class(c_i)\})}{Card(DB)}$$

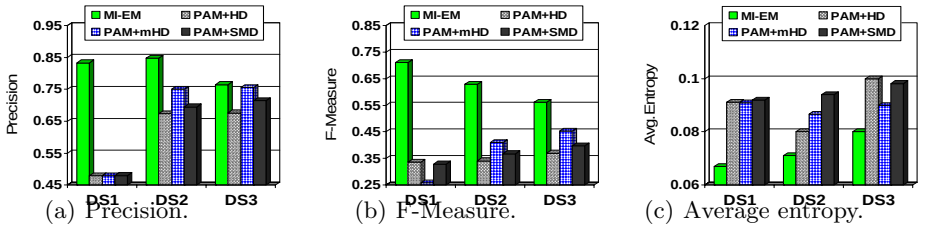
$$Avg.Entropy = \sum_{c_i \in C} (Card(c_i) * (- \sum_{Class_j} p_{j,i} log(p_{j,i}))) / Card(DB)$$

In addition, we measured the average entropy over all clusters. This quality measure is based on the impurity of a cluster c_i w.r.t. the class labels of objects belonging to c_i. Let $p_{j,i}$ be the relative frequency of the class label $Class_j$ in the cluster c_i. We calculate average entropy as following.

In order to demonstrate that the proposed clustering approach for multi-instance objects outperforms standard clustering algorithms working on a suitable distance functions, we compared precision, F-Measure and average entropy of the MI-EM with that of k-medoid clustering algorithm (PAM). To enable cluster analysis of multi-instance objects by PAM, we used the Hausdorff distance (HD)[6], the minimum Hausdorff distance (mHD)[12] and the Sum of Minimum Distances (SMD)[6]. Due to the fact that the data set DS1 has 6 classes and the data sets DS2 and DS3 have 2 classes, we investigated the effectiveness of the cluster analysis where the number of clusters is equal to or slightly than the number of the desired classes. Thus, we set in our experiments the number of clusters equal to 6 and 8 for DS1, and equal to 2, 6 and 8 for the data sets DS2 and DS3. The results of our comparison are illustrated in Figures 1,3 and 2.

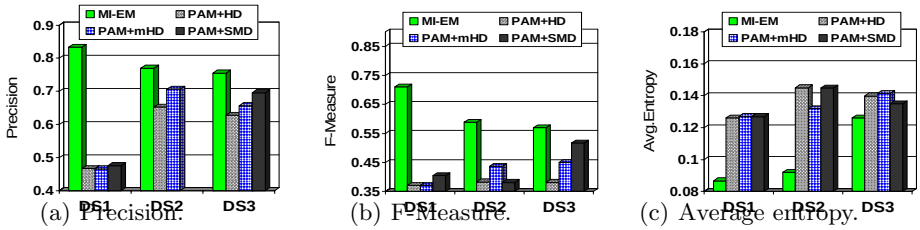
In all our experiments, PAM working on distance functions suitable for multi-instance objects achieved a significantly lower precision than MI-EM. For example, the MI-EM algorithm reached a precision of 0.833 on DS1 and the number of clusters equal to 8 (cf. Figure 2(a)). In contrast to the result of MI-EM, the precision calculated for clusterings found by all competitors lies between 0.478 and 0.48. Furthermore, MI-EM obtained in all experiments higher or comparable values of F-Measures. This fact indicates that the cluster structure found by applying of the proposed EM-based approach is more exact w.r.t. precision and recall than that found by PAM with 3 different MI distance functions. For example, the F-Measure calculated for MI-EM clustering of DS2 with 8 clusters is 0.63 whereas PAM clustering with different MI distance functions shows values between 0.341 and 0.41 (cf. Figure 2(b)). Finally, the values of average entropy observed by the MI-EM results are considerably lower than those of PAM on HD, mHD and SMD. The lower values of average entropy imply a lower level of impurity in the cluster structures detected by applying MI-EM.

To summarize, the values of the different quality measures observed on real world data sets when varying the number of clusters show that the proposed EM-based approach for cluster analysis of MI-objects outperforms the considered competitors w.r.t. effectiveness.



(a) Precision. (b) F-Measure. (c) Average entropy.

Fig. 2. Effectiveness evaluation on DS1, DS2 and DS3 where no. of clusters is 8.



(a) Precision. (b) F-Measure. (c) Average entropy.

Fig. 3. Effectiveness evaluation on DS1, DS2 and DS3 where no. of clusters is 6.

6 Conclusions

In this paper, we described an approach for statistical clustering of MI objects. Our approach models instances as members of concepts in some underlying feature space. Each concept is modeled by a statistical process in this feature space, e.g. a Gaussian. A multi-instance object can now be considered as the result of selecting several times a concept and generating an instance with the corresponding process. Clusters of multi-instance objects can now be described as multinomial distributions over the concepts. In other words, different clusters are described by having different probabilities for the underlying concepts. An additional aspect is the length of the MI object. To derive MI clusters corresponding to this model, we introduce a three step approach. In the first step we derive a mixture model describing concepts in the instance space. The second step finds a good initialization for the target distribution by subsuming each MI object by a so-called confidence summary vector (csv) and afterwards clustering these csvs using the k-means method. In the final, step we employ a final EM clustering step optimizing the distribution for each cluster of MI objects. To evaluate our method, we compared our clustering approach to clustering MI objects with the k-medoid clustering algorithm PAM for 3 different similarity measures. The results demonstrate that the found clustering model offers better cluster qualities w.r.t. to the provided reference clusterings.

References

1. Dietterich, T., Lathrop, R., Lozano-Perez, T.: "Solving the multiple instance problem with axis-parallel rectangles". Artificial Intelligence **89** (1997) 31–71
2. Kriegel, H.P., Schubert, M.: "Classification of websites as sets of feature vectors". In: Proc. IASTED Int. Conf. on Databases and Applications (DBA 2004), Innsbruck, Austria. (2004)
3. Zhou, Z.H.: "Multi-Instance Learning: A Survey". Technical Report, AI Lab, Computer Science a. Technology Department, Nanjing University, Nanjing, China (2004)
4. Ruffo, G.: Learning single and multiple instance decision tree for computer security applications. PhD thesis, Department of Computer Science, University of Turin, Torino,Italy (2000)
5. 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
6. Eiter, T., Mannila, H.: "Distance Measures for Point Sets and Their Computation". Acta Informatica **34** (1997) 103–133
7. Ramon, J., Bruynooghe, M.: "A polynomial time computable metric between points sets". Acta Informatica **37** (2001) 765–780
8. Han, J., Kamber, M.: "Data Mining Concepts and Techniques". Morgan Kaufmann Publishers (2001)
9. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: Proc. Int. Conf. on Knowledge Discovery and Data Mining (KDD). (1996) 291–316
10. Gärtner, T., Flach, P., Kowalczyk, A., Smola, A.: "Multi-Instance Kernels". (2002) 179–186
11. Ng, R., Han, J.: "Efficient and Effective Clustering Methods for Spatial Data Mining". In: Proc. Int. Conf. on Very Large Databases (VLDB). (1994) 144–155
12. Wang, J., Zucker, J.: "Solving Multiple-Instance Problem: A Lazy Learning Approach". (2000) 1119–1125
13. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Academic Press (2001)
14. Fayyad, U., Reina, C., Bradley, P.: "Initialization of Iterative Refinement Clustering Algorithms". In: Proc. Int. Conf. on Knowledge Discovery in Databases (KDD). (1998)
15. Smyth, P.: Clustering using monte carlo cross-validation. In: KDD. (1996) 126–133
16. Wang, J.T.L., Ma, Q., Shasha, D., Wu, C.H.: New techniques for extracting features from protein sequences. IBM Syst. J. **40** (2001) 426–441
17. D.J. Newman, S. Hettich, C.B., Merz, C.: UCI repository of machine learning databases (1998)