Identifying Rare Classes with Sparse Training Data

Mingwu Zhang, Wei Jiang, Chris Clifton, and Sunil Prabhakar

Department of Computer Science, Purdue University
West Lafayette, IN 47907-2107, USA
{mzhang2, wjiang, clifton, sunil}@cs.purdue.edu

Abstract. Building models and learning patterns from a collection of data are essential tasks for decision making and dissemination of knowledge. One of the common tools to extract knowledge is to build a classifier. However, when the training dataset is sparse, it is difficult to build an accurate classifier. This is especially true in biological science, as biological data are hard to produce and error-prone. Through empirical results, this paper shows challenges in building an accurate classifier with a sparse biological training dataset. Our findings indicate the inadequacies in well known classification techniques. Although certain clustering techniques, such as seeded k-Means, show some promise, there are still spaces for further improvement. In addition, we propose a novel idea that could be used to produce more balanced classifier when training data samples are very limited.

1 Introduction

With the explosion of data, data mining techniques gain much attention for their promise in building models and learning patterns from a collection of data. These tasks are essential for decision making and dissemination of knowledge in many areas. Well-known learning techniques such as association rule mining, classification and clustering have been successfully applied in many applications.

Recently, biological science has emerged as a challenging area to apply data mining techniques. One common problem in this field is that given a dataset of which a small fraction has class labels, we need to identify class labels for the other data items [1]. To solve this problem, we can either use supervised (e.g., classification) or unsupervised (e.g., clustering) learning methods. To apply classification techniques, the data with class labels can be treated as a training dataset, and a classifier can be constructed from it. Then, the classifier is used to predict class labels for the rest of unlabeled data items. On the other hand, clustering techniques can also be adopted to achieve this task. For example, assume the total number of class labels in the dataset is known. The dataset can be clustered first, then each unlabeled data is assigned to the majority class label in its cluster. (Thereafter, we term the set of data with class labels as a training dataset and the rest of data as unlabeled data or dataset.)

R. Wagner, N. Revell, and G. Pernul (Eds.): DEXA 2007, LNCS 4653, pp. 751–760, 2007.

Although these techniques can be applied directly, different techniques produce various results. Therefore, how to choose the best method or design a suitable method for a specific domain is challenging. Before making any decision, we first need to understand the characteristics of biological data. Generally speaking, collecting biological data requires major efforts and years of research, and biological data are noisy and error-prone. Thus, it is very likely that the training dataset are *sparse*: either the size of the training dataset is small or the training dataset contains incomplete class information.

For example, in cell wall genomics research, the mutants of cell wall synthesis are extremely valuable to study the genes responsible for biosynthesis of the cell wall and the genes that regulate the cell wall biosynthesis pathways. Traditional experimental methods to find the mutants are time consuming and labor intensive. Although techniques such as Fourier Transform InfraRed microspectroscopy (FTIR) followed by Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA) has been successfully applied to rapidly identify mutants [1], one common challenge biologists faced is the fact that certain mutants might not have visually abnormal or known phenotypes. In other words, there may not exist any training data for these mutants even though they are detectable. These mutants that do not have known phenotypes are very valuable to biologists because their mutations may be in the regulatory component of the cell wall biosynthetic pathways. In addition, these unidentified mutants could be much less common than the known mutants. The problem appears when the training dataset has many data samples for common mutants but very few or none for rare mutants that are potentially important. Consequently, classifiers built on this kind of sparse training dataset are biased toward the common mutants and could be useless in identifying rare mutants. It would be a great loss for biological science if these potentially valuable mutants cannot be discovered.

Another issue that has not been addressed in data mining community is that the training data may not be reliable and contain errors. Some biological experiments (e.g., Yeast 2-hybrid Assay, Mass Spectrometry) are known to produce a large number of false positives. If the results of these experiments are used as the training data for supervised learning, the classifier could be defective because it is built on unreliable training data. Another source of uncertainty comes from the computational methods extensively used in bioinformatics. With the development of high throughput experiment techniques, biologists more and more rely on data mining and machine learning methods to rapidly and automatically process the data. For example, Swiss-Prot is a curated protein function database [2]. To alleviate the intense labor of manually curating protein function annotations, scientists explore using decision tree to predict the functions of the protein sequences [3]. The function annotations in the Swiss-Prot database are used as training data. As Peter Karp points out in [4], some function annotations are computed using computational methods such as BLAST [5] and may not be reliable. Because of the complexity and inherent uncertainty of the biological data, collected biological data samples are very unlikely to be complete and accurate. Therefore, when training data samples are sparse, developing learning techniques that can discover rare important classes and tolerate the noise in the training data has great value.

These examples highlight the nature of biological data in that the training dataset portion is sparse and some rare data may have great value in biological research. Under our problem domain, classification techniques generally perform better than clustering techniques if sufficient and unbiased training data are available. When training data are sparse, the computed classifier is likely biased. We expect that such a classifier is likely to ignore rare class labels. This could lead to potential loss in research. Therefore, with sparse training data, clustering techniques could be the better option in assigning more correct class labels without ignoring rare class items.

Through empirical results, this paper shows challenges faced by biologists in building an accurate classifier with a sparse training dataset. Our findings indicate that when the training dataset is sparse, well known classification techniques are inadequate in producing accurate classifiers. Using them to discover rare classes is almost impossible. Semi-supervised clustering techniques, such as seeded k-means, show some promise in identifying rare classes, but there are still spaces for further improvement. Based on these observations, we also propose a novel idea that could be used to identify rare classes when training data samples are very limited. The paper is organized as follows: Section 2 presents a brief overview of related works. Section 3 provides empirical results showing inadequacies of common classification techniques to detect rare classes when the training dataset is sparse. Section 4 proposes a novel idea in hopes that better learning techniques can be designed to produce unbiased classifiers. Section 5 concludes the paper with lessons learned and future research directions.

2 Related Work

Machine learning and data mining methods can be classified into supervised and unsupervised learnings. Supervised learning requires a training dataset while unsupervised learning does not. Lately, semi-supervised learning has gained increasing attention [6,7] because semi-supervised learning promises the advantage of both supervised and unsupervised methods. In particular, semi-supervised clustering tries to use a small number of labeled data to guide the clustering process. By incorporating the domain knowledge in the clustering process, one hopes that the result of semi-supervised clustering will be better than totally ignoring this information. In [6], unlike the traditional k-means algorithm, instead of using random seed, the initial seeds are the mean of each class of the labeled data.

However, this approach cannot be applied directly to the problem presented in this paper because they assume that every class labels are included in the training dataset. In section 3, we leverage this work and show how to choose the seeds when training data contain incomplete information. A related problem often emerging in biological application is Single Class Classification (SCC). In [8], SCC is defined as distinguishing one class of the data from the universal set of multiple classes. In our problem domain, because we would like to identify rare classes from multiple classes, without training data for the rare classes, single-class approaches cannot be applied.

3 Seeded k-Means vs. Classification Techniques

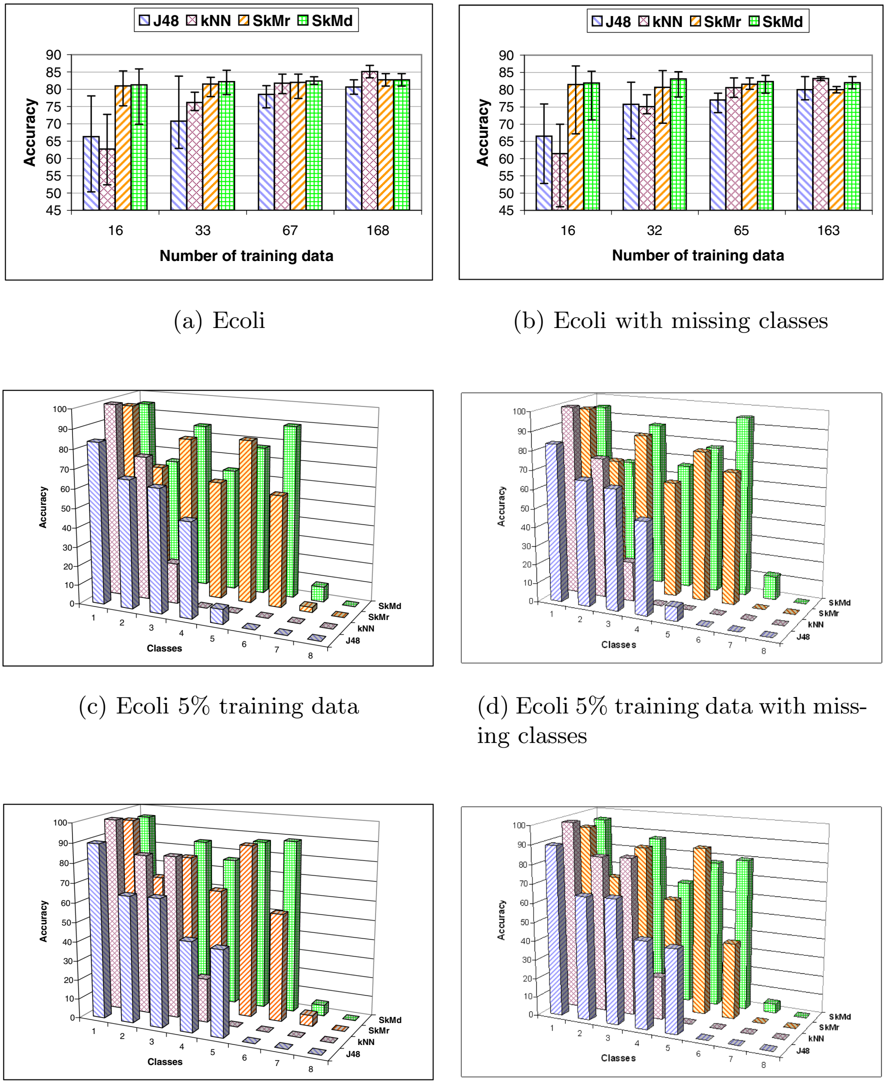
Here, we show that when training data is sparse, well-known classification techniques rarely produce accurate and unbiased classifiers. We also point out that with careful choice of seeds, seeded k-means (SkM) performs better in classifying unknown data instances and identifying rare class instances with a sparse training dataset. By sparse training data, we mean that either the size of the training dataset is small or the training dataset contains incomplete class informations due to errors occurred during data collection process. For the rest of this section, we distinguish these two cases and present our findings independently.

The experiments are done using two datasets, Ecoli and Yeast datasets from UC Irvine Machine Learning Repository [9]. Ecoli dataset contains 336 instances, 7 numeric attributes and 8 classes: cp (143), im (77), pp (52), imU (35), om (20), omL (5), imL (2) and imS (2) (the number in parentheses is the number of instances belonging to that class). The Yeast dataset contains 1462 instances, 8 attributes and 10 classes: CYT (463), NUC (429), MIT (244), ME3 (163), ME2 (51), ME1 (44), EXC (37), VAC (30), POX (20) and ERL (5).

As stated in Section 1, different techniques produce various results. Our experiments focus on three commonly used methods: decision tree (C4.5) [10], k-nearest neighbor (kNN) [11,12], and seeded k-means (SkM) [6]. Based on our own observations, the generic k-means did not produce better results than SkM. Hence, we only show SkM's results. In addition, when there are missing class labels in the training data, SkM cannot be used directly because the number of seeds it picks is equal to the number of distinct class labels in the training dataset. To get around this issue, we propose two variations of SkM: SkM^r and SkM^d. When there are missing class labels, SkM^r chooses the rest of cluster centers randomly (as with the basic k-means) and SkM^d chooses the rest of cluster centers by picking the seed with largest Euclidean distance to the chosen cluster centers, randomly choosing the seed if there are multiple candidates.

Both C4.5 and kNN were used in [13] to classify Ecoli and Yeast datasets, where it was reported that the two classification techniques are most effective for these datasets. We choose the same k values (for kNN) as those used in [13]. First, the training data and test data are generated using Weka [14] to create the stratified n-fold cross-validation. Since we are interested in the situation when little training data is available, we use one fold of data as the training data to build the classifier and n-1 folds of data to test the classifier. In order to fairly compare the clustering techniques with the classifiers, only the test data are used to estimate the accuracy (or precision) and confusion matrices.

Fig. 1 shows the results for Ecoli dataset. Fig. 1 (a) is related to the original Ecoli dataset, and it presents the accuracy changes with the number of data samples varying from 16 to 168 (or 5% to 50%). Each error bar indicates maximum, minimum and average values. Note that the label J48 in the fig-



(a) Ecoli

(b) Ecoli with missing classes

(c) Ecoli 5% training data

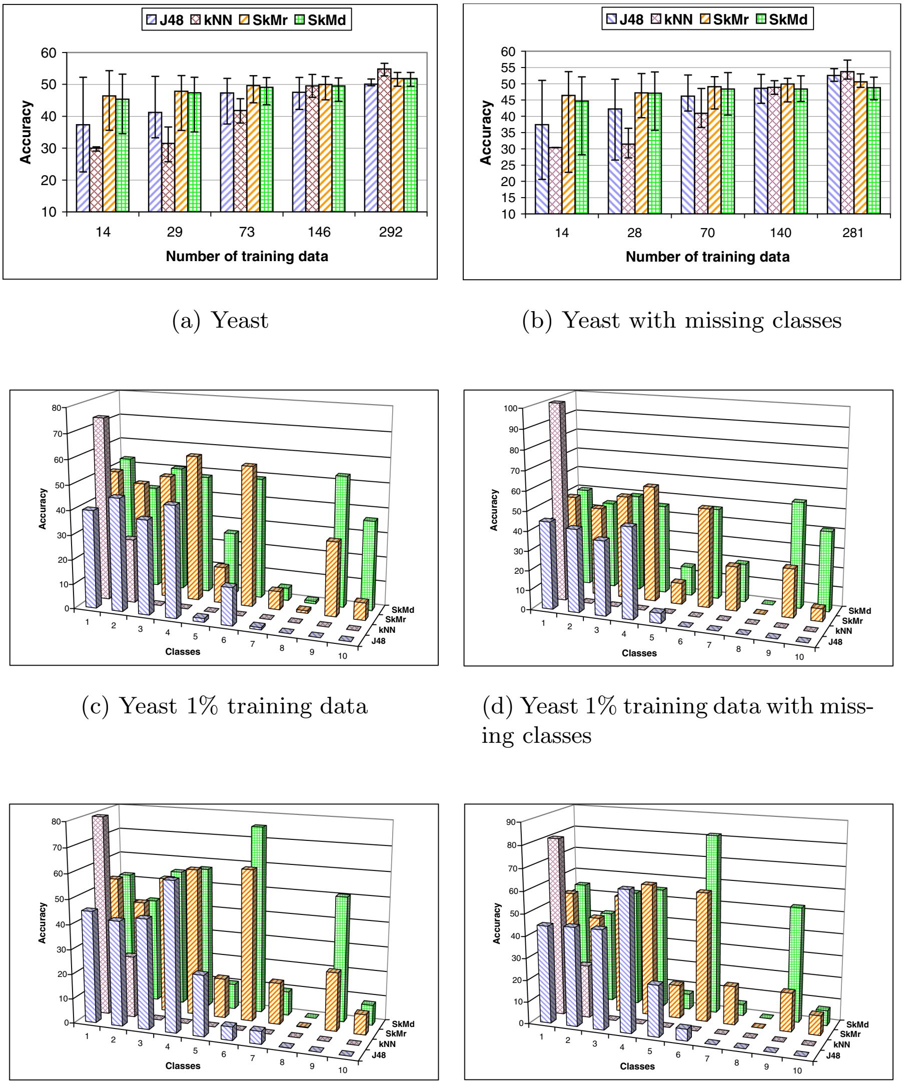
(d) Ecoli 5% training data with missing classes

(e) Ecoli 10% training data

(f) Ecoli 10% training data with missing classes

Fig. 1. Ecoli dataset: Accuracy is computed based on test data only. Subfigures (a) and (b) indicate overall accuracy of each algorithm, and the error bars indicate maximum, minimum and average values across n-1 folds of test data. The rest of subfigures indicate overall accuracy regarding each individual class.

ure indicates a Java implementation of C4.5. The figure shows when the number of training data is small, C4.5 and kNN perform worse than SkM^d and SkM^r. As



(a) Yeast

(b) Yeast with missing classes

(c) Yeast 1% training data

(d) Yeast 1% training data with missing classes

(e) Yeast 2% training data

(f) Yeast 2% training data with missing classes

Fig. 2. Yeast dataset: Accuracy is computed based on test data only. Subfigures (a) and (b) indicate overall accuracy of each algorithm, and the error bars indicate maximum, minimum and average values across n-1 folds of test data. The rest of subfigures indicate overall accuracy regarding each individual class. In addtion, subfigures (d) and (f) are related to the case where class 8 (VAC), class 9 (POX) and class 10 (ERL) have been removed from the training data.

the number of the training data increases, the classifiers outperform both SkM^d and SkM^r. This confirms that when training data are adequate, classification techniques are well suited for the tasks. Since SkM^d performs better than SkM^r, we can be certain that the choice of seeds does make a difference in the outcome.

Fig. 1 (c) and (e) present the accuracy in each class with 5% and 10% of the training data respectively. The figures reveal that the classifiers fail to discover the rare classes, such as class 6 (omL), class 7 (imL), and class 8 (imS), while SkM^d and SkM^r successfully identify class 6 and class 7. For class 5 (om), SkM^d and SkM^r substantially perform better than the two classifiers. In particularly, the SkM^d outperforms SkM^r in rare classes. Since SkM^d chooses the seeds that are furthest away from the known seeds, it has a higher chance of picking a seed that is close to the true center of the rare class. SkM^r chooses a random seed for the missing class, this random seed could be actually in a known cluster and is a bad seed for the missing class. The reason that the classifiers does not perform well is that when the training data is sparse, the training dataset contains none or few data items belonging to rare classes.

In order to test the performance when the training dataset does not contain some rare classes, we remove the most scarce classes from the dataset. In the case of Ecoli dataset, the data of three classes 6, 7 and 8 are removed. The training and test data are generated as described before. Then the rare classes are added back to the test data. Fig. 1 (b)[1] shows that the classifiers perform better as the number of training data increases. Fig. 1 (d) and (f) show that the classifiers fail to discover any rare classes as expected even when the size of the training dataset increases. We conclude that SkM^d and SkM^r outperform classifiers for the rare classes and that SkM^d outperforms SkM^r in rare classes. Fig. 2 shows the results for Yeast dataset indicating similar trends as Ecoli dataset. Particularly, Fig. 2 (d) and (f) show that SkM^d outperforms SkM^r in class 9 (POX) and class 10 (ERL) but fails to identify class 8 (VAL). In [13], even when the whole dataset was used to construct kNN classifier, VAL could not be identified. Thus, we suspect that the training data related to class 8 are too similar to other classes.

4 Entropy-Based Semi-supervised Learning

As discussed in Section 3, when the training dataset is sparse (less than 70 data samples), seeded k-means can classify data more accurate than C4.5 and kNN algorithms. In addition, it can also identify more instances that belong to rare classes. Seeded k-means only utilizes the labeled instances at the initial stage of the algorithm, so here we propose a novel semi-supervised approach that uses labeled instances during each execution round to make a more reasonable and logical choice when assigning a data instance to clusters. We term this new approach as entropy k-means (EkM).

[1] Note that the numbers indicating the sizes of training data are slightly different between the top two sub-figures. This is because the underlying datasets are modified slightly to fit our experiments.

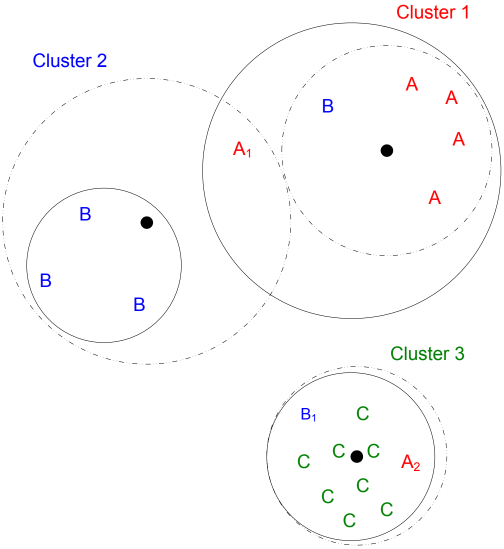


Fig. 3.

The intuition behind the EkM is shown in Figure 3. In this example, the broken line circle represents the clusters before one iteration and solid line circle represents the clusters after the iteration. The distance from the A_1 [2] to the center of cluster 1 is more than that of cluster 2. Under the k-means algorithm, A_1 will be assigned to cluster 2. However, since most data in cluster 2 have class label B and most data in cluster 1 have class label A, it is apparently more reasonable to assign A_1 to cluster 1. Because class labels of some data are known, we want the data having the same class labels stay in the same cluster as much as possible. Having this goal in mind, we incorporate entropy into our decision making process. EkM works as follows: given a small number of data items having class labels, EkM decides a cluster for a data item based on a *score* metric that combines both the distance-based similarity metric and the entropy of the cluster to which the item is added. The metric is defined as:

$$s_i^j = p \cdot D_i^j + q \cdot E_i^j \tag{1}$$

$$E_i^j = \sum_{m=1}^{n} -P_{mi}^j \lg P_{mi}^j \tag{2}$$

In Equation 1, s_i^j represents the *score* of data item t_i related to the center of cluster C_j, D_i^j is the Euclidean distance between C_j's center and t_i; E_i^j is the entropy of C_j when t_i is added. If t_i is labeled, E_i^j is calculated, otherwise the entropy does not change. In Equation 2, P_{mi}^j is the probability a given class

[2] A_1 has class label A. The subscription is used to identify data element.

has been assigned to cluster C_j. n is the number of distinct classes in C_j. The lowest s_i^j value indicates t_i is assigned to C_j during current iteration, and p, q are coefficients to adjust the weight between distance and entropy. This *score* function determines if the distances to the centers of different clusters are similar, the cluster having lower entropy will win. Nevertheless, if the distance to a cluster is very small and adding the data item to this cluster will increase its entropy, this could indicate that the label actually is not correct. Consequently, EkM will allow this item to go into a different cluster with the correct class label. We expect that EkM will perform better in classifying and identifying rare classes.

The distance and entropy can offset each other. Fig. 3 shows a situation where distance could dominate entropy. A_2 and B_1 are assigned to cluster 3. Although we would like the data items having the same labels stay together, the distance from A_2 to cluster 1 is much larger than the distance to cluster 3. The same is true for B_1. In this case, the distance is too large to overcome and entropy has little impact on the score metric; as a result, B_1 and A_2 should be assigned to cluster 3 if we believe Euclidean distance is a good measure of similarity between objects. On the other hand, if the distance from A_1 to cluster 1 is close enough to that from A_1 to cluster 2, the entropy will guide A_1 to cluster 1.

Our work is still in preliminary stage, but it did show some promise on certain datasets. Several issues need to be solved before giving a full evaluation of EkM: using the *score* metric, convergence is not guaranteed because EkM is no longer an EM based algorithm. Also, how to decide the values for p and q (in equation 1) is another challenge. Our experiments conducted on Ecoli and Yeast datasets show the magnitudes of distance and entropy are very similar, so we set $p = q = 0.5$. In other words, distance and entropy have the same weight in calculation of the *score* metric. In general, we think p and q are dataset dependent. Furthermore, overfitting could cause potential problems in identifying rare classes when using classification techniques. Since our proposed approach is a combination of classification and clustering, our approach might be less likely to cause overfitting. We will investigate this issue extensively in the future.

Another issue is that using this as an enhancement on k-means only affects the center of the cluster. In reality, clusters may have different sizes or shapes; using the (limited) class data to adjust size/shape of clusters as well as the center would have even greater promise. We have started with a k-means basis due to the success of k-means clustering in our problem domain, but K-means algorithm assumes that K is known in advance, which may not be true for biological applications. (e.g., the number of types of mutants are not known). Density-based and hierarchical clustering algorithms are more suitable. We believe the entropy-based idea can be used to guide density-based or hierarchical clustering as well. The difficulty is to avoid over-reliance on the known data (leading to the same problem of not recognizing rare classes that standard classifiers face), while still getting full benefit. The simplicity of k-means makes this less of a problem; further research is needed to see how this can affect other techniques.

5 Conclusion / Future Work

We have shown that when rare classes have few instances or are completely missing in the training data, classification techniques using this training data perform poorly to identify rare classes. We also showed seeded k-means can be adopted in our problem domain, but the choice of seeds makes a difference. In the future, we will systematically and theoretically investigate the best ways to choose these seeds. Under the semi-supervised learning framework, we proposed a novel idea that incorporates entropy into the *score* metric to guide the clustering process. The preliminary results show some promise in identifying rare classes, and we will thoroughly investigate this idea and apply it to a real application in cell wall genomics. Since many clusters in biological data do not have a spherical shape, we will extend this idea into density-based clustering techniques.

References

1. Chen, L., Carpita, N., Reiter, W., Wilson, R., Jeffries, C., McCann, M.: A rapid method to screen for cell-wall mutants using discriminant analysis of fourier transform infrared spectra. The plant Journal 16(3), 385–392 (1998)
2. Apweiler, R., Bairoch, A., Wu, C.H., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M.J., Natale, D.A., O'Donovan, C., Redaschi, N., Yeh, L.L.: Uniprot: the universal protein knowledgebase. Nucleic Acids Research 32, D115–D119 (2004)
3. Kretschmann, E., Fleischmann, W., Apweiler, R.: Automatic rule generation for protein annotation with the c4.5 data mining algorithm applied on swiss-prot. Bioinformatics 17(10), 920–926 (2001)
4. Karp, P.D.: What we do not know about sequence analysis and sequence databases. BioInformatics 14(9), 753–754 (1998)
5. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.: Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acids Res. 25(17), 3389–3402 (1997)
6. Basu, S., Banerjee, A., Mooney, R.J.: Semi-supervised clustering by seeding. In: ICML, pp. 27–34 (2002)
7. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: ICML (2004)
8. Yu, H.: Svmc: Single-class classification with support vector machines. In: IJCAI, pp. 567–574 (2003)
9. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)
10. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Francisco (1993)
11. Duda, R., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley & Sons, Chichester (1973)
12. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, San Diego (1990)
13. Horton, P., Nakai, K.: Better prediction of protein cellular localization sites with the *k* nearest neighbors classifier. In: Proc Int Conf Intell Syst Mol Biol., pp. 147–152 (1997)
14. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Fransisco (1999)