

# Adaptive Active Classification of Cell Assay Images

Nicolas Cebron and Michael R. Berthold

ALTANA Chair for Bioinformatics and Information Mining  
Department of Computer and Information Science  
University of Konstanz  
Box M 712, 78457 Konstanz, Germany  
{cebron, berthold}@inf.uni-konstanz.de

**Abstract.** Classifying large datasets without any a-priori information poses a problem in many tasks. Especially in the field of bioinformatics, often huge unlabeled datasets have to be explored mostly manually by a biology expert. In this work we consider an application that is motivated by the development of high-throughput microscope screening cameras. These devices are able to produce hundreds of thousands of images per day. We propose a new adaptive active classification scheme which establishes ties between the two opposing concepts of unsupervised clustering of the underlying data and the supervised task of classification. Based on Fuzzy  $c$ -means clustering and Learning Vector Quantization, the scheme allows for an initial clustering of large datasets and subsequently for the adjustment of the classification based on a small number of carefully chosen examples. Motivated by the concept of active learning, the learner tries to query the most informative examples in the learning process and therefore keeps the costs for supervision at a low level. We compare our approach to Learning Vector Quantization with random selection and Support Vector Machines with Active Learning on several datasets.

## 1 Introduction

Traditionally, a classifier is built on a given set of labeled training data. This is known as supervised learning, as the classifier gets supervision in the form of labeled instances. This can be very useful in many settings - however, sometimes a large pool of unlabeled data is available and the cost of determining the class label for all these examples is prohibitively high. An example for such a setting may be the categorization of web pages, where we have a small set of labeled webpages and a large set of unlabeled examples.

One traditional technique to make use of unlabeled data is clustering - grouping objects that are similar to each other. It is classically used to reveal the underlying structure of the given data. The most important advantage of this method is that it can be used without any supervision by the user. This technique is known as unsupervised learning.

There are also semi-supervised learning techniques that take advantage of a small pool of labeled examples that help to guide the algorithm; they are still

influenced by the unlabeled data. Examples for techniques that use a small set of labeled examples in clustering can be found in [15] and [1].

A more recent approach is the concept of active learning [4]. Active learning handles the setting where a large pool of unlabeled samples is available and where we have access to a (usually noiseless) oracle, often a human expert, that can determine the class label of an instance. The examples to query are chosen by the learner with a certain strategy so as to optimize the prediction accuracy while at the same time keeping the number of queries low.

In this work, we consider a more special setting that is based on the classification of cell assay images (see Section 4). In our scenario, a large number of unlabeled images of cell assays are available, whereas we only have a human biology expert who is able to provide us with class labels for each cell image.

As we do not have any labeled instances at the beginning, we introduce a new approach that establishes ties between the opposed methods of unsupervised and supervised learning. First, the dataset is explored to find the groupings (hopefully related to possible clusters of the same class) whereas in the second step, the accuracy of the classifier is optimized by querying “useful” examples.

In Section 2, we recapitulate the concept of active learning. Section 3 describes the Fuzzy  $c$ -means algorithm with noise detection and the Learning Vector Quantization algorithm, which formed the foundation for our proposed adaptive active clustering scheme that is described in more detail at the end of this section. A useful application for this scheme – mining of cell assay images – is explained in Section 4. We study the behavior of our algorithm and compare it to other methods in Section 5, before we draw conclusions in Section 6.

## 2 State of the Art

In many classification tasks it is common that a large pool of unlabeled examples  $U$  is available whereas the cost of getting a label for an example is high. The concept of active learning [4] tackles this problem by enabling a learner to pose specific queries, chosen from an unlabeled dataset. In this setting, we assume that we have access to a noiseless oracle that is able to predict the class label of a certain sample. Given an unlabeled dataset  $U$ , a labeled dataset  $L$  and a set of possible labels  $C$ , we can describe an active learner as a tuple  $(f, q)$ .  $f : L \cup U \mapsto C$  is the classifier, trained on the labeled (and sometimes also the unlabeled) data. The query function  $q$  makes a decision based on the currently labeled samples, which examples from  $U$  should be chosen for labeling. The active learner returns a new classifier  $f'$  after each pool query or a fixed number of pool queries.

Many active learning strategies for different kinds of algorithms exist. In [4], a selective sampling is performed according to where the most general and the most specific hypotheses disagree. The hypotheses were implemented using feed-forward neural networks with backpropagation. Active Learning with Support Vector Machines (SVM) has also become very popular. The expensive learning process for the SVM can be reduced by querying examples with a certain

strategy. In [16], the query function chooses the next unlabeled data point closest to the decision hyperplane in the kernel induced space. Support Vector Machines with active learning have been widely used for image retrieval problems [12] [17] or in the drug discovery process [18]. However, they require at least some labeled examples from the start, which usually results in some randomly chosen examples to be queried, which is rather inefficient.

To make use of the underlying distribution of the given unlabeled data, we use an approach that clusters the data. To date, research on approaches that combine clustering and active learning has been sparse.

In [1], a clustering of the dataset is obtained by first exploring the dataset with a *Farthest-First-Traversal* and providing *must-link* and *cannot-link* constraints. In the second *Consolidate*-phase, the initial neighborhoods are stabilized by picking new examples randomly from the dataset and again by providing constraints for a pair of data points.

In [7], an approach for active semi-supervised clustering for image database categorization is investigated. It includes a cost-factor for violating pairwise constraints in the objective function of the Fuzzy *c*-means algorithm. The active selection of constraints looks for samples at the border of the least well-defined cluster in the current iteration.

However, our approach differs from the others in the way that the data is preclustered before supervision enhances the classification accuracy. Thus, our scheme is able to first explore and later finetune the classification of a large unlabeled dataset in an efficient and accurate way.

### 3 Active Classification

In this section, we present our new active classification scheme. Starting with a short revision of the Fuzzy *c*-means algorithm (with noise detection) in 3.1 and the Learning Vector Quantization (LVQ) algorithm in 3.2, we propose a query function in 3.3 and put the pieces together for our adaptive active classification scheme in 3.4.

#### 3.1 Fuzzy *c*-Means with Noise Detection

The Fuzzy *c*-means (FCM) algorithm [2] is a well-known unsupervised learning technique that can be used to reveal the underlying structure of the data based on a similarity measure. Fuzzy clustering allows each data point to belong to several clusters, with a degree of membership to each one. We use the extended version from [5] for the added detection of noise.

Let  $T = \vec{x}_i$ ,  $i = 1, \dots, |T|$  be a set of feature vectors for the data items to be clustered,  $W = \vec{w}_k$ ,  $k = 1, \dots, c$  a set of  $c$  clusters.  $V$  is the matrix with coefficients where  $v_{i,k}$  denotes the membership of  $\vec{x}_i$  to cluster  $k$ . Given a distance function  $d$ , the fuzzy *c*-means algorithm with noise detection iteratively minimizes the following objective function with respect to  $v$  and  $w$ :

$$J_m = \sum_{i=1}^{|T|} \sum_{k=1}^c v_{i,k}^m d(\vec{w}_k, \vec{x}_i)^2 + \delta^2 \sum_{i=1}^{|T|} \left( 1 - \sum_{k=1}^c v_{i,k} \right)^2. \quad (1)$$

$m \in (1, \infty)$  is the fuzzification parameter and indicates to what extent the clusters are allowed to overlap each other. The first term corresponds to the normal fuzzy  $c$ -means objective function, whereas the second term arises from the noise cluster.  $\delta$  is the distance from every data point to the auxiliary noise cluster (thus, there are  $c + 1$  cluster with the extra cluster serving as the noise cluster). This distance can either be fixed or updated in each iteration according to the average interpoint distances. Objects that are not close to any of the cluster centers  $\vec{w}_k$  are therefore detected as having a high membership to the noise cluster.  $J_m$  is subject to minimization under the constraint

$$\forall i : 0 \leq \sum_{k=1}^c v_{i,k} \leq 1. \quad (2)$$

FCM is often used when there is no a-priori information available and thus can serve as an overview technique. Based on the prototypes obtained from the FCM algorithm, we can classify the dataset by first querying the class label for each cluster prototype and then by assigning each data point the class label of the closest prototype. A common problem is that the cluster structure does not necessarily correspond to the distribution of the classes in the dataset. Therefore, we have to update the cluster prototypes subsequently. As the fuzzy  $c$ -means algorithm does not provide any way to do this, we use the Learning Vector Quantization algorithm for this task, which is introduced in the next section.

### 3.2 Learning Vector Quantization

Learning Vector Quantization [11] is a so-called competitive learning method. The algorithm works as follows: for each training pattern, the nearest prototype is identified and updated. The update depends on the class label of the prototype and the training pattern. If they possess the same class label, the prototype is moved closer to the pattern, otherwise it is moved away. The learning rate  $\epsilon$  controls the movement of the prototypes. The learning rate is decreased during the learning phase, a technique known as *simulated annealing* [10]. The LVQ algorithm terminates if the prototypes stop changing significantly.

One basic requirement in the LVQ algorithm is that we can provide a class label for each training point  $\vec{x}_i$  that is randomly sampled. We assume that the training set is unlabeled – however an expert can provide us with class labels for some selected examples. As we can only label a small set of examples, we need to optimize the queries with a strategy to boost the classification accuracy while keeping the number of queries at a low level. In the next section, we propose a query function that attempts to solve this problem.

### 3.3 Selection of Patterns to Query

The selection of patterns is of particular importance as it influences the performance of the classification. Assuming access to a noiseless oracle it is vital to gain as much information as possible from the smallest possible number of

examples. We propose a sampling scheme that covers two aspects: exploration and exploitation. This coincides with the ideas proposed in [14] that an active learning scheme should not only refine the decision boundaries but also needs to verify the current hypothesis. The prior data distribution plays an important role. In [3] the authors propose to minimize the expected error of the learner:

$$\int_x E_T [(\hat{y}(x; D) - y(x))^2 | x] P(x) dx \quad (3)$$

where  $E_T$  denotes the expectation over  $P(y|x)$  and  $\hat{y}(x; D)$  the learner's output on input  $x$  given training set  $D$ . If we act on the assumption that the underlying structure found by the FCM algorithm already inheres an approximate categorization, we can select further examples by querying data points at the classification boundaries. That means we approximately take into account the prior data distribution  $P(x)$ .

**Exploration.** In order to have information about the general class label of the cluster itself that represents our current hypothesis, we perform a technique known as *Cluster Mean selection* [6]. Each cluster is split into subclusters. Subsequently, the nearest neighbor of each subcluster prototype is selected for the query procedure. If a subcluster is not homogeneous – meaning, the labels of the subclusters in the current cluster are different – this subcluster is split up again until the labels are homogeneous or we have reached a given recursion depth.

**Exploitation.** We assume that the most informative data points lie between clusters of different classes that are not well separated from each other. We call these regions *areas of possible confusion*. This coincides with the findings and results in [6] and [13].

To identify the data points that lie on the frontier between two clusters, we propose a new procedure that is easily applicable in the fuzzy setting. Rather than dynamically choosing only one example for the labeling procedure in each step (which would make it too slow), we focus on a selection technique that selects a whole batch of  $N$  samples to be labeled. Note that a data item  $\vec{x}_i$  is considered as belonging to cluster  $k$  if  $v_{i,k}$  is the highest among its membership values. If we consider the data points between two clusters, they must have an almost equal membership to both of them. The selection is performed in two steps: first, all data points are ranked according to their memberships to cluster prototypes with different classes. Then, the most diverse examples are chosen from this pool of examples. The ranking is based on the fuzzy memberships and can be expressed for each data point  $\vec{x}_i$  as follows:

$$\text{Rank}(\vec{x}_i) = 1 - (\min |v_{i,k} - v_{i,l}|) \quad \forall k, l = 1, \dots, c \wedge k \neq l \quad (4)$$

Note that we also take into account the class label of each cluster. Only if the clusters correspond to different classes, the rank is computed. After all data points have been ranked, we can select a subset with high ranks to perform the

next step: diversity selection. This prevents the active clustering scheme from choosing points that are too close to each other (and therefore are altogether not that interesting). We refer to the *farthest-first-traversal* [8] usually used in clustering. It selects the most diverse examples by choosing the first point at random and the next points as farthest away from the current set of selected instances. The distance  $d$  from a data point  $x$  to the set  $S$  is defined as  $d(S, x) = \min_{y \in S} d(x, y)$ , known as the *min-max-distance*.

### 3.4 Adaptive Active Classification

Our adaptive active classification procedure is based on a combination of the techniques mentioned above. All steps are listed in Algorithm 1. We start to cluster our dataset with the fuzzy  $c$ -means algorithm, because we expect dense regions in the feature space to occur which are likely to bear the same class label. Therefore, the fuzzy  $c$ -means algorithm should give us a good initialization and prevents us from labeling unnecessary instances in the first querying step. The noise detection in the clustering procedure serves the same purpose: rare data points that represent borderline cases should not be selected, as these noise labels would influence classification in a negative way. Furthermore, these samples would be useless for classification. Note that in this way we have the possibility to present strange cases to the user, which is often also of interest. After a batch of  $N$  examples has been selected from within each cluster and at the borders of the clusters, the user interaction takes place: the expert has to label the selected examples. The newly labeled samples are then added to the current set of labeled samples  $L$ . After this step, the cluster prototypes can be moved based on the training set  $L$ .

---

#### Algorithm 1. Adaptive Active Clustering Procedure

---

- 1:  $L \leftarrow 0$
  - 2: Perform the fuzzy  $c$ -means algorithm with noise detection (unsupervised).
  - 3: Filter out data points belonging to noise cluster.
  - 4: **while** Classification accuracy not satisfactory **do**
  - 5:   Select  $N$  training examples within the clusters and at the borders.
  - 6:   Ask the user for the labels of these samples, add them to  $L$ .
  - 7:   Move the prototypes according to  $L$  (supervised).
  - 8:   Decrease the learning rate  $\epsilon$ .
  - 9: **end while**
- 

One open question is when to stop the movement of the prototypes. The simulated annealing in the LVQ algorithm will stop the movement after a certain number of iterations. However, an acceptable solution may be found earlier, which is why we propose further stopping criteria:

**Validity Measures:** Such measures can give us information on the quality of the clustering [19]. We employ the within cluster variation and the between

cluster variation as an indicator. This descriptor can be useful for the initial selection of attributes. Naturally, the significance of this method decreases with the proceeding steps of labeling and adaptation of the cluster prototypes.

**Classification Gradient:** We can make use of the already labeled examples to compare the previous to the newly obtained results. After the labels of the samples inside and between the clusters have been obtained, the cluster prototypes are moved. The new classification of the dataset is derived by assigning to each data point the class of its closest cluster prototype. By comparing the labels given by the user to the newly obtained labels from the classification, we can calculate the ratio of the number of correctly labeled samples to the number of falsely labeled examples.

**Tracking:** Another indicator for acceptable classification accuracy is to track the movement of the cluster prototypes. If they stop moving because new examples do not augment the current classification, we can stop the procedure.

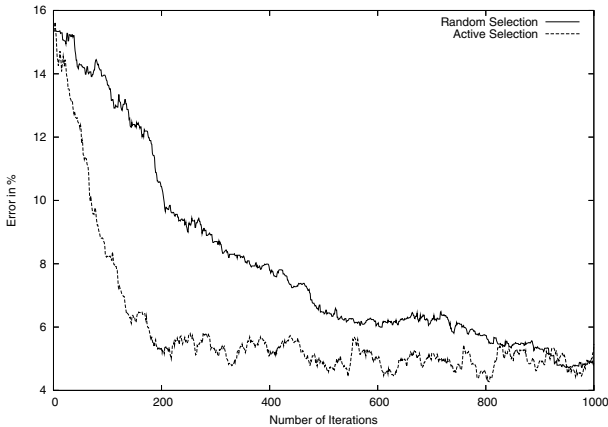
**Visual Inspection:** If the data points are linked to images (as in the setting we describe in Section 4), we can make use of this additional information. Instead of presenting the numerical features, we select the corresponding image of the data tuple that is closest to the cluster prototype. We display the images with the highest membership to the actual cluster and the samples at the boundary between two clusters if they are in different classes. The decision whether the classification accuracy needs improvement can be made by the user based on this visual inspection.

## 4 Application: Cell Assay Mining

The development of high throughput imaging instruments, e.g. fluorescence microscope cameras, resulted in them becoming a promising tool to study the effect of drug candidates on different cell types. These devices are able to produce hundreds of thousands of images per day.

The goal of the cell assay image mining is to label a few selected cell images by hand and to automatically label the vast majority of the images afterwards. In order to obtain a classification of one image, it is segmented into small subimages, each containing one cell of the original image. The segmentation allows us to consider the cells separately in order to distinguish between different reactions of cells in the same image. When most of the small subimages are classified, a classification of the original image can be made by a majority decision.

Our Cell Assay Image Mining System consists of three major elements: the segmentation module, the feature extraction module, and the classification element. Obviously the number of data points is very large; because we segment thousands of images into small subimages, we reach an order of millions of images. This dataset is an ideal instance for an active learning scheme. In this setting, the oracle is represented by a biology expert who is able to provide a class label for a cell image that is shown to him.



**Fig. 1.** Error plot using different sampling strategies for 1 pattern per time

The classification of new images is obtained by classifying each individual cell within the given image. Each cell is assigned to a cluster and its corresponding class. The proportion of the distribution of the different classes is the decisive factor for classifying the whole image. If a clear majority decision can be made, the image is not considered further. Borderline cases with equal distributions of classes are sorted into a special container to be assessed manually by the biology expert. It becomes apparent that this approach allows for a rather high fault tolerance, as a human will have no objections to label a few images by hand rather than to risk a misclassification.

## 5 Experimental Results

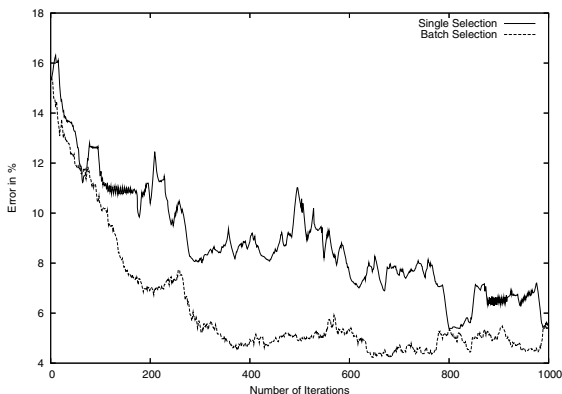
To demonstrate the behavior of our adaptive active classification scheme, we first want to show the behavior of our algorithm on artificial data; in the second section we show examples with real world cell image data.

### 5.1 Artificial Data

The artificial data used in this section is a 2-dimensional dataset consisting of several classes that overlap in the feature space. The class distribution is skewed, taking arbitrary shapes.

In the first setting, we compare our approach to random selection usually used in the LVQ algorithm. As our prototypes are all well initialized, we omit the exploration step (the initial Fuzzy  $c$ -means (sub)clustering and labeling) and only focus on the exploitation step of our active classification scheme.

The query function we introduced prevents the LVQ algorithm from choosing instances that are not relevant for classification. The error plot in Figure 1 shows that the active selection leads to a significantly faster convergence of the



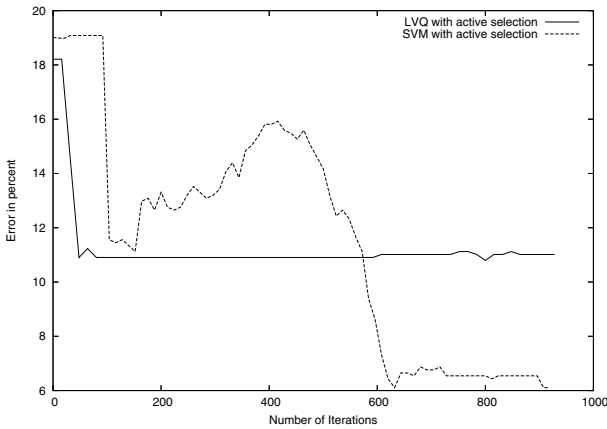
**Fig. 2.** Classification error of our active classification scheme against active Support Vector Machine on the two-class problem

classification, especially at the first iterations. This is exactly our goal as we want to keep the user interaction at a low level.

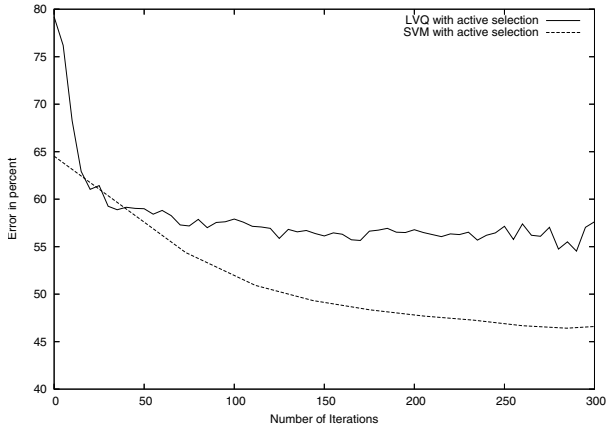
Another issue that we want to take a look at is the benefit of batch sampling. One could argue that it is enough to determine the most interesting point at each iteration and then to move the prototypes. We perform a batch sampling that allows a diversity selection to be carried out for performance reasons as well. The benefit of batch sampling is demonstrated in Figure 3, where we plot the error in percent for sampling just one data point at each iteration versus sampling multiple points in each iteration. In fact, the single sampling approach performs much worse than batch selection in this case.

## 5.2 Cell Assay Image Data

As the cell image data we use is confidential, we show results on a different dataset from the same application area from the NISIS pap-smear competition [9]. The task is to classify pre-stages of cervical cancer in cells before they progress to invasive carcinoma. The data consist of 917 images of Pap-smear cells, classified carefully by cyto-technicians and doctors. Each single cell image is described by 20 numerical features, and the cells fall into 7 classes. A basic data analysis [9] includes linear classification results, in order to provide lower bounds on the acceptable performance of other classifiers. We compared our approach to an approach with a Support Vector Machine with active learning [16]. However, note that the active SVM is initialized differently by choosing random examples from each class. In our setting of cell assay image mining, where we have no labeled instances at the beginning, this step would not be possible, and a random initialization of the SVM would increase the number of queries for the active SVM significantly. Note also, that the performance of the active SVM depends heavily on the chosen kernel function. We used a polynomial kernel, with which the active SVM performed best.



**Fig. 3.** Single sampling vs. batch sampling (10 examples per batch selection)



**Fig. 4.** Classification error of our active classification scheme against active Support Vector Machine on the seven-class problem

The original pap-smear cell dataset with 7 classes can be transformed into a 2-class problem. The results of the comparison between our scheme and the active SVM are shown in Figure 2. The classification error of the linear classifier (trained on 90% of the data) is approximately 10%. As can be seen, both classifiers can reach this performance, the active SVM reaches a classification error of approximately 6% after approximately 600 queries. Our adaptive active classification scheme reaches an error of approximately 11% on the data, however it reaches this performance considerably faster.

On the original 7-class problem, we compared our scheme to active SVM after 300 steps. Since SVMs built binary classifiers, for each class an independent SVM has to be trained against all other classes. Therefore computation for the

optimization of the SVM was not feasible with more steps. Naturally, the batch size of queries for the active SVM is much higher than for our scheme, as we need examples for all classes in each iteration. The results of the comparison can be seen in Figure 4. The classification error of the linear classifier has been given with approximately 40%. Neither the SVM nor our scheme reach this accuracy after 300 queries. The active SVM has an error of approximately 45% whereas our scheme reaches approximately 56%.

From these results we conclude, that our adaptive active classification scheme is a promising approach to tackle the problem of cell assay classification. Its performance is superior to random selection and comparable with a Support Vector Machine with Active Learning on the two-class problem. For the multi-class problem performance is still acceptable but lower than the active SVM. However, the active SVM requires carefully chosen kernels and some pre-labeled examples. Our approach is also computationally more efficient, which is essential for our application where we need to classify tens of millions of cell images.

## 6 Conclusions

In this work, we have addressed the problem of classifying a large dataset when only a few labeled examples can be provided by the user. We have introduced a new adaptive active classification scheme that starts with the fuzzy  $c$ -means algorithm for an initial clustering. The classification of the dataset is obtained by labeling the cluster prototypes and assigning the label of the closest prototype to all data points. We have proposed to move the cluster prototypes, similar to the Learning Vector Quantization (LVQ) method to obtain results closer to the expectation of the user. From the unlabeled pool of instances, new examples are chosen by a query function that makes use of the fuzzy memberships to the cluster prototypes combined with a diversity selection. Based on the labels of the selected examples at the borders between clusters and the labeled examples inside clusters, the prototypes are moved. We have shown that the misclassification rate can be improved more quickly. We have discussed an application in the mining of cell assay images, where the data often inherits the aforementioned properties.

## Acknowledgments

This work was supported by the DFG Research Training Group GK-1042 "Explorative Analysis and Visualization of Large Information Spaces".

## References

1. S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. *Proceedings of the SIAM International Conference on Data Mining (SDM-2004)*, 2004.
2. J. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

3. D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. *Advances in Neural Information Processing Systems*, 7:705–712, 1995.
4. D. A. Cohn, L. Atlas, and R. E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.
5. R. N. Davé. Characterization and detection of noise in clustering. *Pattern Recogn. Lett.*, 12(11):657–664, 1991.
6. B. Gabrys and L. Petrakieva. Combining labelled and unlabelled data in the design of pattern classification systems. *International Journal of Approximate Reasoning*, 2004.
7. N. Grira, M. Crucianu, and N. Boujemaa. Active semi-supervised clustering for image database categorization. *Content-Based Multimedia Indexing*, 2005.
8. Hochbaum and Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985.
9. J. Jantzen, J. Norup, G. Dounias, and B. Bjerregaard. Pap-smear benchmark data for pattern classification, 2006.
10. S. Kirkpatrick, C. D. G. Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
11. T. Kohonen. *Self-Organizing Maps*. Springer Verlag, Heidelberg, 1995.
12. T. Luo, K. Kramer, D. Goldgof, L. Hall, S. Samson, A. Remsen, and T. Hopkins. Active learning to recognize multiple types of plankton. *Journal of Machine Learning Research*, pages 589–613, 2005.
13. H. Nguyen and A. Smeulders. Active learning using pre-clustering. *ICML*, 2004.
14. T. Osugi, D. Kun, and S. Scott. Balancing exploration and exploitation: A new algorithm for active machine learning. *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 330–337, 2005.
15. W. Pedrycz and J. Waletzky. Fuzzy clustering with partial supervision. *IEEE Transactions on systems, man and cybernetics Part B: Cybernetics*, 27:177–185, 1997.
16. G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. *ICML Proceedings, 17th International Conference on Machine Learning*, pages 839–846, 2000.
17. L. Wang, K. L. Chan, and Z. h. Zhang. Bootstrapping SVM active learning by incorporating unlabelled images for image retrieval. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:629–634, 2003.
18. M. K. Warmuth, G. Raetsch, M. Mathieson, J. Liao, and C. Lemmen. Support vector machines for active learning in the drug discovery process. *Journal of Chemical Information Sciences*, pages 667–673, 2003.
19. M. Windham. Cluster validity for fuzzy clustering algorithms. *Fuzzy Sets and Systems*, 5:177–185, 1981.