Mining of Cell Assay Images Using Active Semi-Supervised Clustering [*]

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

Abstract

Classifying large datasets without any a-priori information poses a problem especially in the field of bioinformatics. In this work, we explore the problem of classifying hundreds of thousands of cell assay images obtained by a high-throughput screening camera. The goal is to label a few selected examples by hand and to automatically label the rest of the images afterwards. We deal with three major requirements: first, the model should be easy to understand, second it should offer the possibility to be adjusted by a domain expert, and third the interaction with the user should be kept to a minimum. We propose a new active clustering scheme, based on an initial Fuzzy c-means clustering and Learning Vector Quantization. This scheme can initially cluster large datasets unsupervised and then allows for adjustment of the classification by the user. Furthermore, we introduce a framework for the classification of cell assay images based on this technique. Early experiments show promising results.

1. Introduction

The development of high-throughput imaging instruments, e. g. fluorescence microscope cameras, resulted in them becoming the major tool to study the effect of agents on different cell types. These devices are able to produce more than 50.000 images per day; up to now, cell images are classified by a biological expert who writes a script to analyze a cell assay. As the appearance of the cells in different assays change, the scripts must be adapted individually. Finding the relevant features to classify the cell types correctly can be difficult and time-consuming for the user.

The aim of our work is to design classifiers that are both able to learn the differences between cell types and are easy to interpret. As we are dealing with non-computer experts, we need models that can be grasped easily. We use the concept of clustering to reduce the complexity of our image dataset. Cluster analysis techniques have been widely used in the area of image database categorization.

Especially in our case, we have many single cell images with similar appearance that may nevertheless be categorized in different classes. Another case might be that the decision boundary between active and inactive is not reflected in the numerical data that is extracted from the cell image. Furthermore, the distribution of the different cell types in the whole image dataset is very likely to be biased. Therefore, the results of an automatic classification based on an unsupervised clustering may not be satisfactory, thus we need to adapt the clustering so that it reflects the desired classification of the user.

As we are dealing with a large amount of unlabeled data, the user should label only a small subset to train the classifier. Choosing randomly drawn examples from the dataset would render the classifier biased toward the underlying distribution of different kinds of cells in the cell assay images. Instead of picking redundant examples, it would be better to pick those that can ”help” to train the classifier.

This is why we try to apply the concept of active learning to this task, where our learning algorithm has control over which parts of the input domain it receives information about from the user. This concept is very similar to the human form of learning, whereby problem domains are examined in an active manner.

To this date, research on approaches that combine clustering and active learning is 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 in-

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

cludes a cost-factor for violating pairwise constraints in the objective function of the Fuzzy c-means algorithm. The active selection of constrains looks for samples at the border of the least-well defined cluster in the current iteration.

Our approach is similar to the latter one, although we do not update the cluster centers in each iteration but after an initial fuzzy c-means clustering.

In Section 2, we briefly recapitulate the fuzzy c-means algorithm, Section 3 describes our approach for the active selection of constraints, and the moving of the cluster prototypes. In Section 4, we introduce our prototype of a Cell Assay Image Mining System with its subcomponents for the image processing, before presenting first experimental results in Section 5.

2. Fuzzy c-means

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. Fuzzy clustering allows each data point to belong to several clusters, with a degree of membership to each one.

Let $T = \vec{x}_i$ $, i = 1, \ldots, m$ be a set of feature vectors for the data items to be clustered, $W = \vec{w}_k, k = 1, \ldots, c$ 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 iteratively minimizes the following objective function with respect to v and w:

$$J_m = \sum_{i=1}^{|T|} \sum_{k=1}^{c} v_{i,k}^m d(\vec{w}_k, \vec{x}_i)^2 \tag{1}$$

$m \in (1, \infty)$ is the fuzzification parameter and indicates how much the clusters are allowed to overlap each other. J_m is subject to minimization under the constraint

$$\forall i : \sum_{k=1}^{c} v_{i,k} = 1 \tag{2}$$

FCM is often used when there is no a-priori information available and thus can serve as an exploratory technique. A common problem is that the cluster structure does not necessarily correspond to the classes in the dataset. This is why the FCM algorithm is used only as a preprocessing technique. The fuzzy memberships $v_{i,k}$ prove useful for the selection of datapoints at the border between clusters as we will see in Section 3.1.

3. Active Learning

In order to improve the performance of the classification based on the initial, unsupervised clustering, we aim to guide the clustering process. Because we have no a-priori information about the class distribution in the dataset, we need to adapt the cluster prototypes so that they closer model the boundaries between the classes. This is done in two steps: 1. Labeling of a few ”interesting” examples and 2. moving the prototypes according to these labels. These steps are discussed in detail in the following sections.

3.1. Selection of Constraints

We presume that we have access to a user (in our case the biological expert) who can give us labels for different data points. Another option would be that the user can define a constraint between a given pair (x_i, x_j) of data points. The assets and drawbacks of giving labels vs. constraints are discussed in [3].

We assume that the most informative data points lie between clusters that are not well separated from each other, so-called areas of possible confusion. This coincides with the findings and results in [6] and [13]. The prior data distribution plays an important role, [4] proposes to minimize the expected error of the learner:

$$\int_x E_T \left[(\widehat{y}(x; D) - y(x))^2 | x \right] P(x) dx \tag{3}$$

where E_T denotes the expectation over $P(y|x)$ and $\widehat{y}(x; D)$ 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 better examples by querying data points at the classification boundaries. That means we take into account the prior data distribution $P(x)$.

In order to have information about the general class label of the cluster itself, we let the user label the cluster centers using for each the nearest neighbor in the dataset, a technique known as ”Cluster Mean selection” [6]. If more than one example per cluster shall be labeled, one can either split the corresponding cluster into subclusters, or alternatively select prototypes near to the one that was selected first.

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 one example for the labeling procedure, 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. Given a threshold $t \in [0, 1]$ the condition can be expressed as follows:

$$|v_{i,k} - v_{i,l}| < t, \quad v_{i,k}, v_{i,l} \geq \frac{1}{c} \tag{4}$$

In order to find N samples to query, we start with a high value for t and reduce it iteratively until we have obtained a set of $\leq N$ samples. Figure 1 shows an example of three clusters and the selected examples generated by this procedure.

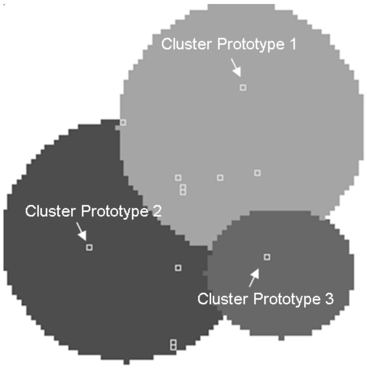

Figure 1. Three initial clusters and the selected examples

Having obtained the labels for the cluster centers and for a few confusing examples between them, we propose a new method in the next section to adapt the actual clusters to better match the underlying decision boundaries.

3.2. Learning Vector Quantization

The learning vector quantization algorithm [12] is a well-known competitive learning method. The outline of the algorithm is given in Algorithm 1. The LVQ algorithm needs the class information for all training examples. Since we can provide the class information only for a few selected examples, we need to optimize the selection of them.

Algorithm 1 LVQ algorithm

```
1: Choose $R$ initial prototypes for each class $m_1(k)$, $m_2(k), \ldots, m_R(k), k = 1, 2, \ldots, K$, e. g. by sampling $R$ training points at random from each class.
2: Sample a training point $\vec{x}_i$ randomly (with replacement) and let $m_j(k)$ denote the closest prototype to $\vec{x}_i$. Let $g_i$ denote the class label of $\vec{x}_i$ and $g_j$ the class label of the prototype.
3: if $g_i = g_j$ then {that is they belong to the same class}
4:     move the prototype toward the training point: $m_j(k) \leftarrow m_j(k) + \epsilon(\vec{x}_i - m_j(k))$, where $\epsilon$ is the learning rate.
5: end if
6: if $g_i \neq g_j$ then {that is they belong to different classes}
7:     move the prototype away from the training point: $m_j(k) \leftarrow m_j(k) + \epsilon(\vec{x}_i - m_j(k))$
8: end if
9: Repeat step 2, decreasing the learning rate $\epsilon$ to zero with each iteration.
```

3.3. Adaptive Active Learning

Our approach to optimize the LVQ algorithm includes the selective sampling scheme given in the previous section. It is dependent on an initial clustering and constitutes an extension to the LVQ-algorithm to choose the best examples. The total active clustering process is outlined in Algorithm 2.

Algorithm 2 Adaptive Active Clustering Procedure

```
1: Perform the fuzzy $c$-means algorithm (unsupervised).
2: Select $N$ training examples with the most similar membership to several clusters.
3: Ask the user for the labels of these samples.
4: Move the prototypes according to the label of the prototype and the samples.
5: Evaluation: If classification is better or matches the expected error then stop.
6: Repeat step 2, decreasing the learning rate $\epsilon$ to zero with each iteration.
```

Our initial prototypes in step 1 are the ones obtained from a fuzzy c-means clustering. Having the labels for each cluster prototype, we can select the next candidates for the query procedure along each border between two clusters. Which datapoints are selected depends on the chosen number N of examples we want to query. If N is small (approximately the number of clusters), some inter-cluster relationships will not be queried because the corresponding clusters are clearly separated in the feature space. As our approach focuses on separating classes given an initial "meaningful" clustering, this does not pose a problem. In each phase of our adaptation of the LVQ algorithm, we move the cluster centers according to a batch of N training points. For each point from the sample set we determine the two clusters that it lies in between and let the user label this point. We only update the two cluster prototypes that are involved at this point and leave the rest unchanged. We repeat the step of selecting a batch of training examples, then move the cluster centers for each point, decreasing the learning rate ϵ in each iteration. The process of selected examples and their influence on the prototypes can be seen clearly in Figure 2, where we assume that the majority of examples selected have the class label of cluster 2. The question is when to stop the movement of the cluster centers. 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, that is why a second stopping criterion is introduced.

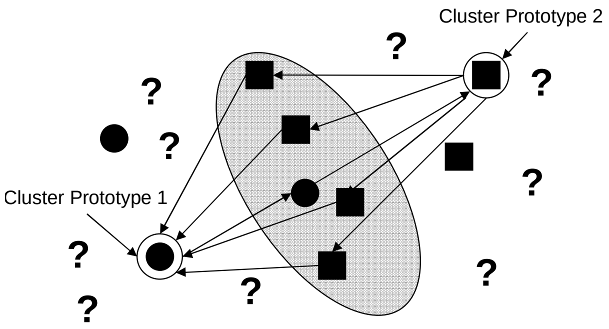

Figure 2. Points selected and their influence on the cluster prototypes

We make use of the already labeled examples to compare the previous to the newly obtained results. After the labels of the samples between cluster centers 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.

4. Application: Cell Assay Image Mining

Our Cell Assay Image Mining System consists of three major elements: The segmentation module, the feature extraction module, and the classification element. Based on a modular workflow system, the user can choose and interact with the different modules and create a dataflow. This allows the user to enable and try out different settings interactively. Different modules for feature extraction or segmentation can be integrated. Figure 3 gives an overview of a typical workflow.

In the following sections, we focus on the different modules in more detail.

4.1. Segmentation

In order to calculate the features for each cell individually, the cell assay image has to be segmented. We prefer this approach in contrast to [10], because we need to identify interesting substructures in one image. The segmentation allows us to consider the cells separately in order to distinguish between different reactions of cells in the same image.

Unfortunately, the appearance of different cell types can vary dramatically. Therefore, different methods for segmentation have to be applied according to the different cell types. Work to segment and subdivide cells into the cell nucleus and cytoplasm based on seeded region growing is currently under progress. We follow the same assumption as in the approach from [11] that is, the nucleus can be detected more easily.

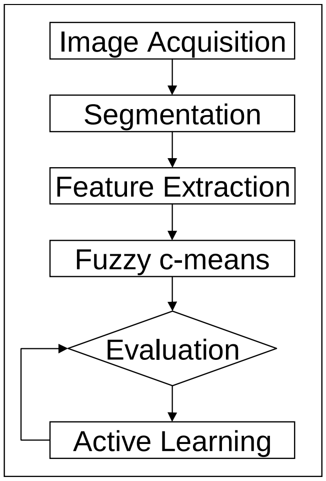

Figure 3. Workflow

4.2. Feature Extraction

The feature extraction module calculates features of a cell image based on the histogram (first order statistics) and based on the texture (second order statistics). The histogram features comprise the mean, variance, skewness, kurtosis, and entropy of the histogram.

The 14 texture features from Haralick [8] represent statistics of the co-occurrence matrix of the gray level image. Four co-occurrence matrices from horizontal, vertical, diagonal, and antidiagonal direction are averaged to achieve rotation invariance. These features provide information about the smoothness, contrast or randomness of the image - or more general statistics about the relative positions of the gray levels within the image.

4.3. Classification

The classification module comprises the initial fuzzy c-means clustering, the cluster evaluation and the Active Learning Module. As described in Section 3, we utilize the FCM to obtain our first set of cluster prototypes. The evaluation of the actual clustering can be based on several factors:

Cluster Validity Measures can give us information of the quality of the clustering [15]. We employ the within cluster variation and the between cluster variation as an indicator. This descriptor can be useful for the initial selection of features. Naturally, the significance of this method decreases with the proceeding steps of labeling and adaptation of the cluster prototypes.

Visual Cluster Inspection allows the user to make a judgment of the clustering quality. 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. This approach is obviously prone to mistakes due to wrong human perception and should therefore be used only as an overview technique.

Evaluation in Adaptive Active Learning is performed as described in Section 3.3 where we judge the new classification based on the previously labelled examples. This method is the most suitable method to evaluate the quality of the classification. It also allows for the possibility to show the progress of the classification, so that the user can decide whether he wants to continue or not.

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 biological 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

As is the nature of the active learning problem, we do not have a large labeled dataset available to test the performance of our scheme. Therefore, we have created several artificial test sets to evaluate our classifier.

The first test set demonstrates the mode of action of our new active clustering scheme and is shown in Figure 1. It is a 3-dimensional artificial test set consisting of 4036 samples. The class label is indicated as the brightness. This dataset shows a typical problem where one class is underrepresented and the decision boundaries of the unsupervised clustering are not optimal because of the bias of the data. Figure 4 shows the class boundaries of the cluster prototypes and the decision boundaries. The optimum of 10 steps has been performed, selecting $N = 5$ examples in each iteration. As can clearly be seen, the active clustering scheme improves the positions of the cluster prototypes significantly to reduce the classification error. On the other hand, too many steps decrease the performance. An overview of the number of steps and the misclassification rate is shown in Table 1. As we can see, the bias of the classifier can be reduced and the decision boundaries between overlapping classes in the feature space can be optimized.

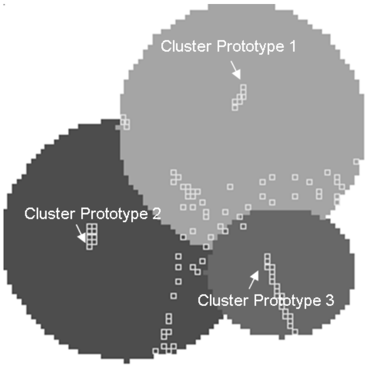

Figure 4. Movement of cluster prototypes over time and set of additionally labeled examples

# steps	Misclassification rate
0	16.13 %
5	11.89%
8	8.94%
9	8.57%
10	8.00%
11	8.45%
12	8.82%
15	10.16%
25	25.54%

Table 1. Number of steps vs. Misclassification Rate

In our second test set, we added some noise to the clusters to test how distortion of class labels at the border influences the moving of the cluster prototypes. The effect on this dataset is shown in Figure 5. With the increasing noise at the border between clusters, the misclassification rate based on the initial unsupervised clustering naturally increases, too. With 10 steps and $N = 5$ labeled examples on the borders, we improved the misclassification rate from 17.95 % to 9.97 %. We increased the noise at the borders (see Figure 6) with the result that the misclassification rate improved from 27.77 % to 17.96 % . In this case, the initial clustering had two cluster prototypes that belonged

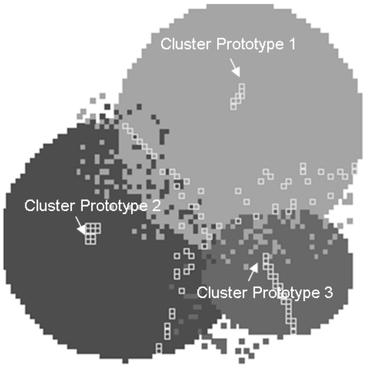

Figure 5. Movement of cluster prototypes over time and set of additionally labeled examples with noise

to the same class. This could be neutralized by requerying the labels for the cluster prototypes in each step of our adaptive active clustering procedure. This seems also useful to explore new classes in the dataset that have not yet been found if not enough clusters have been used. We observed this phenomenon in the Ionosphere-dataset from the UCI Repository [5], too. Having used four clusters to classify the data, only one class has been found from the initial fuzzy c-means clustering. The additional queries allowed to find the second class and due to this we were able to shift the classification accuracy. However, this is not the major goal of our algorithm.

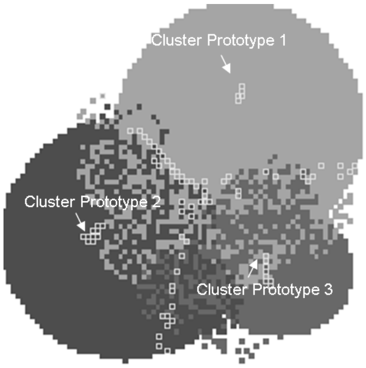

Figure 6. Movement of cluster prototypes over time and set of additionally labeled examples with more noise

6. Related Work

There have been a number of approaches to perform partial supervision in the clustering process. In the aforementioned works from [1] and [7], the objective function of the fuzzy c-means algorithm is extended by a cost factor for violating pairwise constraints. In the work of [14], labeled patterns are incorporated in the objective function of the Fuzzy ISODATA algorithm. All these approaches take a set of labeled patterns or constraints as input before the clustering process is started. These samples are selected randomly.

In [9], a very similar approach to our own work has been proposed that selects the points to query based on the Voronoi diagram that is induced by the reference vectors. The datapoints to query are selected from the set of Voronoi vertices with different strategies. However, our approach differs from all others in the way that the data is preclustered before supervision enhances the classification accuracy and the queries can be obtained in a fast and simple way.

7. Conlusions and Future Work

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 shown that the fuzzy c-means algorithm is well applicable for stable initial clustering and that it has the advantage that data points on the border can easily be detected by scanning through their memberships to the cluster prototypes. Based on the labels of the selected examples at the borders between clusters and the labeled cluster prototypes, we have proposed to move the cluster prototypes, similar to the Learning Vector Quantization (LVQ) method. We have shown that the misclassification rate can be improved, especially when the class distributions are skewed. We have discussed an application in the mining of cell assay images, where the data often inherits the aforementioned properties.

Future work needs to be done to optimize the number N of queries that are posed during the active clustering process. It would be desirable to pose just as many queries as necessary. Another important point are wrong classifications given by the user. Examples that contradict each other in terms of the model by their given labels could be requeried to be able to filter out wrong human classifications.

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, R. Caruana, and A. McCallum. Semi-supervised clustering with user feedback, 2003.

[4] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. *Advances in Neural Information Processing Systems*, 7:705–712, 1995.

[5] C. B. D.J. Newman, S. Hettich and C. Merz. UCI repository of machine learning databases, 1998.

[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] R. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on systems, man and cybernetics*, 1973.

[9] M. Hasenjäger and H. Ritter. Active learning with local models. *Neural Processing Letters*, 7:107–117, 1998.

[10] K. Huang and R. F. Murphy. Automated classification of subcellular patterns in multicell images without segmentation into single cells. *IEEE Intl Symp Biomed Imaging (ISBI)*, pages 1139–1142, 2004.

[11] T. R. Jones, P. Golland, and A. Carpenter. Voronoi-based segmentation on manifolds, 2005.

[12] T. Kohonen. *Self-Organizing Maps*. Springer Verlag, Heidelberg, 1995.

[13] H. Nguyen and A. Smeulders. Active learning using pre-clustering. *ICML*, 2004.

[14] W. Pedrycz and J. Waletzky. Fuzzy clustering with partial supervision. *IEEE Transactions on systems, man and cybernetics Part B: Cybernetics*, 27.

[15] M. Windham. Cluster validity for fuzzy clustering algorithms. *Fuzzy Sets and Systems*, 5:177–185, 1981.