Integrating Input from Human Experts into Prototype-based Classifier Learning

Petra Perner

Institute of Computer Vision and Applied Computer Sciences, IBaI
PSF 301114, 04251 Leipzig, Germany
pperner@ibai-institut.de

Abstract. An expert is able to tell the system developer in many image-related tasks what a prototypical image should look like. Usually he will choose several prototypes for one class, but he cannot provide a good and large enough sample set for the class to train a classifier. Therefore, we mapped his technical procedure into a technical system based on proper theoretical methods that assist him in acquiring the knowledge about his application and furthermore in developing a classifier for his task. This system helps him to learn about the clusters and the borderlines of the clusters even when the data are very noisy as is the case for microscopic cell images in drug discovery, where it is unclear if the drug will produce the expected result on the cell parts.
We describe in this paper the necessary functions that a prototype-based classifier should have. We also use the expert's estimated similarity as a new knowledge piece and based on that we optimize the similarity. The test of the system was carried out on a new application on microscopic cell image analysis - the study of the internal mitochondrial movement of cells. The aim was to discover the different dynamic signatures of mitochondrial movement. Three results of this movement were expected: tubular, round, and dead cells. Based on our results we can show the success of the developed method.

Keywords: Internal Mitochondrial Movement, Cell Biology, Similarity Measure, Case-Based Reasoning, Prototype-Based Classification, Knowledge Acquisition, Feature Subset Selection, Prototype Selection, Adjustment Theory

1 Introduction

Prototypical classifiers have been successfully studied for medical applications by Schmidt and Gierl [1], by Perner [2] for image interpretation and by Nilsson and Funk [3] on time-series data. The simple nearest-neighbor approach [4], as well as hierarchical indexing and retrieval methods [5], have been applied to the problem. It has been shown that an initial reasoning system could be built up based on prototypical cases. The systems are useful in practice and can acquire new cases for further reasoning during utilization of the system. Prototypical images are a good starting point for the development of an automated image classifier [6]. This knowledge is often collected by human experts in image catalogues. We describe, based on a task for the study of the internal mitochondrial movement of cells [7], how such a classifier in combination with image analysis can be used for incremental knowledge acquisition and automatic classification. The work enhances our previous work on a prototype-based classifier [2] by introducing the expert's estimated similarity as a new knowledge piece and a new function that adjusts this similarity and the automatically calculated similarity by the system in order to improve the system accuracy. The test of the system is done on a new application on cell image analysis - the study of the internal mitochondrial movement of cells.

The classifier is set up based on prototypical cell appearances in the image such as for e.g. „healthy cell", „dead cell", and „cell in transition stage". For these prototypes are calculated image features based on a random set theory that describes the texture on the cells. The prototype is represented then by the feature-value pair and the class label. These settings are taken as initial classifier settings, in order to acquire the knowledge about the dynamic signatures.

The importance of the features and the feature weights are learned by the protoclass-based classifier [2]. After the classifier is set up each new cell is then compared by the protoclass-based classifier and the similarity to the prototypes is calculated. If the similarity is high the new cell gets the label of the prototype. If the similarity to the prototypes is too low, then there is evidence that the cell is in transition stage and a new prototype has been found. With this procedure we can learn the dynamic signature of the mitochondrial movement.

In Section 2 we present the methods for our prototype-based classifier. The material is described in Section 3 for the internal mitochondrial movement of cells. In Section 4 is presented the methodology for the knowledge acquisition based on a prototype-based classification. Results are given in Section 5 and finally in Section 6 conclusions are presented.

2 ProtoClass Classifiers

A prototype-based classifier classifies a new sample according to the prototypes in the data base and selects the most similar prototype as output of the classifier. A proper similarity measure is necessary to perform this task, but in most applications there is no a-priori knowledge available that suggests the right similarity measure. The method of choice to select the proper similarity measure is therefore to apply a subset of the numerous similarity measures known from statistics to the problem and to select the one that performs best according to a quality measure such as, for example, the classification accuracy. The other choice is to automatically build the similarity metric by learning the right features and feature weights. The latter one we chose as one option to improve the performance of our classifier.

When people collect prototypes to construct a dataset for a prototype-based classifier, it is useful to check if these prototypes are good prototypes. Therefore a function is needed to perform prototype selection and to reduce the number of prototypes used for classification. This results in better generalization and a more noise-tolerant classifier. If an expert selects the prototypes, this can result in bias and possible duplicates of prototypes causing inefficiencies. Therefore a function to assess a collection of prototypes and identify redundancy is useful.

Finally, an important variable in a prototype-based classifier is the value used to determine the number of closest cases and the final class label.

Consequently, the design-options for the classifier to improve its performance are prototype selection, feature-subset selection, feature weight learning and the 'k' value of the closest cases (see Figure 1).

We assume that the classifier can start in the worst case with only one prototype per class. By applying the classifier to new samples the system collects new prototypes. During the lifetime of the system it will chance its performance from an oracle-based classifier, which will classify the samples roughly into the expected classes, to a system with high performance in terms of accuracy.

In order to achieve this goal we need methods that can work on a low number of prototypes and on large number of prototypes. As long as we have only a few prototypes feature subset selection and learning the similarity might be the important features the system needs. If we have more prototypes we also need prototype selection.

For the case with a low number of prototypes we chose methods for feature subset selection based on the discrimination power of features. We use the feature based calculated similarity and the pair-wise similarity rating of the expert and apply the adjustment theory [11] to fit the similarity value more to the true value.
For a large number of prototypes we choose a decremental redundancy-reduction algorithm proposed by Chang [8] that deletes prototypes as long as the classification accuracy does not decrease. The feature-subset selection is based on the wrapper approach [9] and an empirical feature-weight learning method [10] is used. Cross validation is used to estimate the classification accuracy. A detailed description of our prototype-based classifier ProtoClass is given in [2]. The prototype selection, the feature selection, and the feature weighting steps are performed independently or in combination with each other, in order to assess the influence these functions have on the performance of the classifier. The steps are performed during each run of the cross-validation process.

The classifier schema shown in Figure 1 is divided in the design phase (Learning Unit) and the normal classification phase (Classification Unit). The classification phase starts after we have evaluated the classifier and determined the right features, feature weights, the value for 'k' and the cases.

Our classifier has a flat data base instead of a hierarchical one that makes it easier to conduct the evaluations.

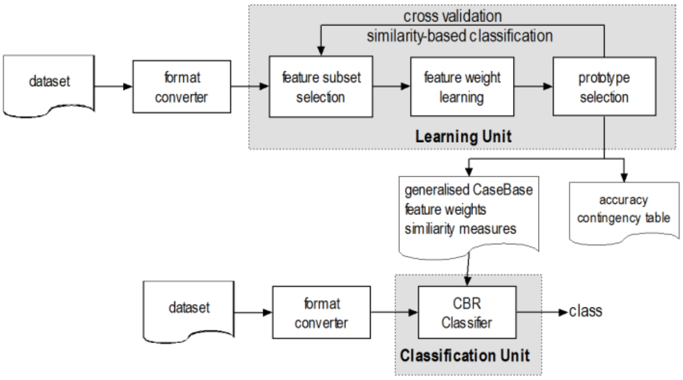


Fig. 1. Prototype-based Classifier

2.1 Classification Rule

Assume we have n prototypes that represent m classes of the application. Then, each new sample is classified based on its closeness to the n prototypes. The new sample is associated with the class label of the prototype that is the closest one to sample.

More precisely, we call $x_n' \in \{x_1, x_2, \ldots, x_i, \ldots x_n\}$ a closest case to x if $\min d(x_i, x) = d(x_n', x)$, where $i=1,2,\ldots,n$.

The rule chooses to classify x into category C_l, where x_n' is the closest case to x and x_n' belongs to

class C_l with $l \in \{1, \ldots, m\}$.

In the case of the k-closest cases we require k samples of the same class to fulfill the decision rule. As a distance measure we can use any distance metric. In this work we used the city-block metric.

The pair-wise similarity measure Sim_{ij} among our prototypes shows us the discrimination power of the chosen prototypes based on the features.

The calculated feature set must not be the optimal feature subset. The discrimination power of the features must be checked later. For a low number of prototypes we can let the expert judge the similarity $SimE_{ij}$ between $i, j \in \{1, \ldots, n\}$ the prototypes. This gives us further information about the problem which can be used to tune the designed classifier.

2.2 Using Expert's Judgment on Similarity and the Calculated Similarity to Adjust the System

Humans can judge the similarity $SimE_{ij}$ among objects on a rate between *0* (identity) and *1*(dissimilar). We can use this information to adjust the system to the true system parameters [11].

Using the city-block distance as distance measure, we get the following linear system of equations:

$$SimE_{ij} = \frac{1}{N}\sum_{l=1}^{N} a_l \left| f_{il} - f_{jl} \right| \tag{1}$$

with $i, j \in \{1, \ldots, n\}$, f_{il} the feature *l* of the *i-th* prototype and *N* the number of features.

The feature a_l is the normalization of the feature to the range *{0,1}* with $a_l = \frac{1}{\left| f_{\max,l} - f_{\min,l} \right|}$ that is calculated from the prototypes. That this is not the true range of the feature value is clear since we have a low number of samples. The factor a_l is adjusted closer to the true value by the least square method using expert`s $SimE_{ij}$:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(SimE_{ij} - \frac{1}{N}\sum_{l=1}^{N} a_l \left| f_{il} - f_{jl} \right| \right)^2 \Rightarrow Min! \tag{2}$$

with the restriction $0 \le a_l \le \frac{1}{\left| f_{\max,l} - f_{\min l} \right|}$.

3 Methodology

Figure 2 summarizes the knowledge-acquisition process based on protoclass-based classification.

We start with one prototype for each class. This prototype is chosen by the biologist based on the appearance of the cells. It requires that the biologist has enough knowledge about the processes going on in cell-based assays and can decide what kind of reaction the cell is showing.

The discrimination power of the prototypes is checked first based on the feature values measured from the cells and the chosen similarity measure. Note that we calculated a large number of features for each cell. However, using many features does not mean that we will achieve a good discrimination power between the classes. It is better to come up with one or two features for small sample sizes in order to ensure a good performance of the classifier. The expert manually estimates the similarity between the prototypes and inputs these values into the system. The result of this process is the selection of the right similarity measure and the right number of features. With this information is set up a first classifier and applied to real data.

Each new data gets associated with the label of the classification. Manually we evaluate the performance of the classifier. The biologist gives the true or gold label for the sample seen so far. This is kept into a data base and serves as gold standard for further evaluation. During this process the expert will sort out wrongly classified data. This might happen because of too few prototypes for one class or because the samples should be divided into more classes. The decision what kind of technique should be applied is made based on the visual appearance of the cells. Therefore, it is necessary to display the prototypes of the classes and the new samples. The biologist sorts these samples based on the visual appearance. That this is not easy to do by humans is clear and needs some experiences in describing image information [6]. However, it is a standard technique in psychology, in particular in gestalts psychology, and known as categorizing or card sorting. As a result of this process we come up with more prototypes for one class or with new classes and at least one prototype for these new classes.

The discrimination power needs to get checked again based on this new data set. New features, a new number of prototypes or a new similarity measure might be the output. The process is repeated as long as the expert is satisfied with the result. As a result of the whole process we get a data set of samples with true class labels, the settings for the protoclass-based classifier, the important features and the real prototypes. The class labels represent the categories of the cellular processes going on in the experiment. The result can now be taken as a knowledge acquisition output. Just for discovering the categories or the classifier can now be used in routine work at the cell-line.

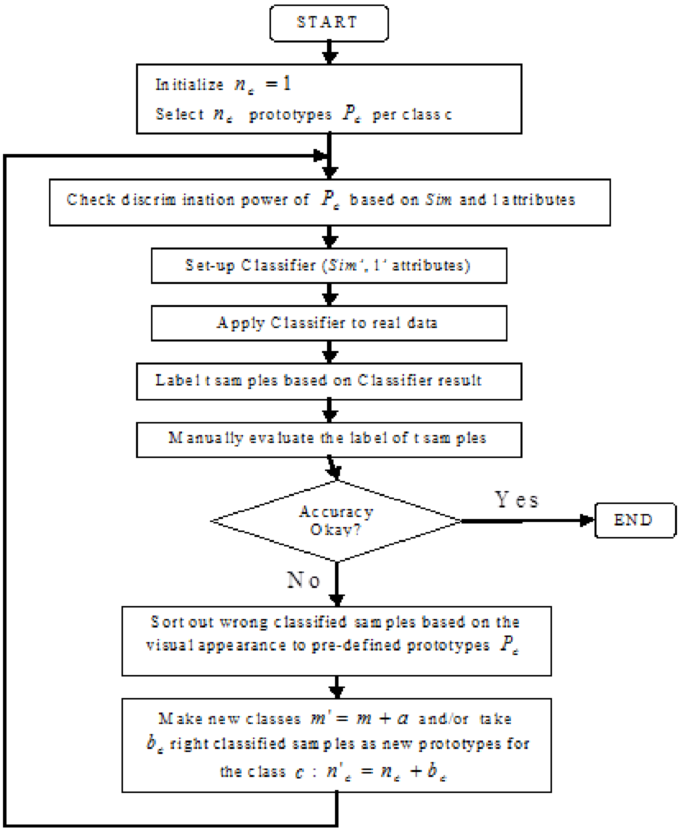


Fig. 2. Methodology for Prototype-based Classification

4 THE APPLICATION

After the assay has been set up, it is not quite clear what the appearances of the different phases of a cell are. This has to be learnt during the use of the system.

Based on their knowledge the biologists set up several descriptions for the classification of the mitochondria. They grouped these classes into the following classes: tubular cells, round cells and dead cells. For the appearance of these classes see images in Figure 3.

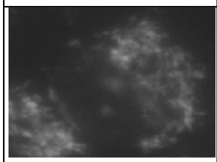
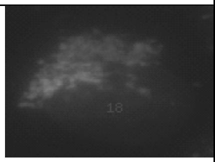
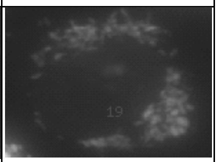
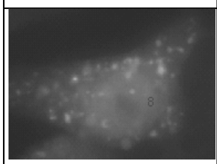
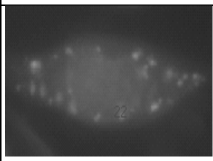
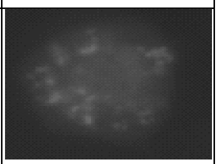
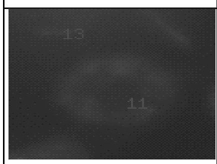
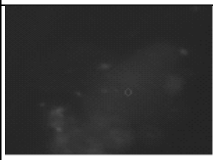
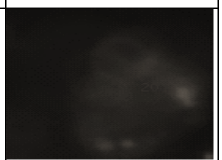
Class Tubular		
B10_1	B10_18	B10_19
		
Class Round		
B03_8	B03_22	B03_26
		
Class Death		
B03_11	B06_0	B06_20
		

Fig. 3. Sample Images for three Classes (top Class Tubular, middle Class Round, bottom Class Death)

Then prototypical cells were selected and the features were calculated with the software tool *CellInterpret* [12]. The expert rated the similarity between these prototypical images.

Our data set consist of 223 instances with the following class partition: 36 instances of class *Death*, 120 instances of class *Round*, 47 instances of class *Tubular*, and 114 features for each instance.

The expert chose for each class a prototype shown in Figure 4. The test data set for classification has then 220 instances. For our experiments we also selected 5 prototypes pro class respectively 20 prototypes pro class. The associate test data sets do not contain the prototypes.

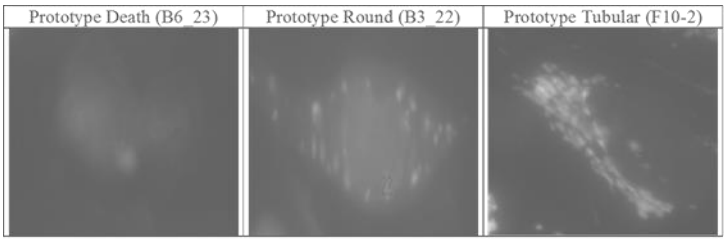


Fig. 4. The Prototypes for the classes Death, Round and Tubular

5 Results

Figure 5 shows the accuracy for classification based on different numbers of prototypes for all features and Fig. 6 shows the accuracy for a test set based on only the three most discriminating features. The test shows that the classification accuracy is not so bad for only three prototypes, but with the number of prototypes the accuracy increases. The selection of the right subset of features can also improve the accuracy and can be done based on the method presented in Section 2 for a low number of samples. The right chosen number of closest cases k can also help to improve accuracy, but cannot be applied if we only have three prototypes or less in the data base.

Figure 7 shows the classification results for the 220 instances started without adjustment meaning the weights are equal to one (1;1;1) and with adjustment based on expert`s rating where the weights are (0.00546448; 0.00502579; 0.00202621) as an outcome of the minimization problem.

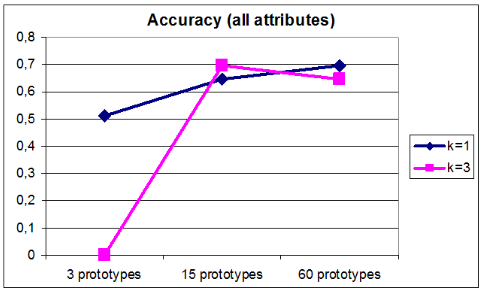


Fig. 5. Accuracy versus Prototypes and for two different feature subsets; Accuracy for different number of prototypes using all features

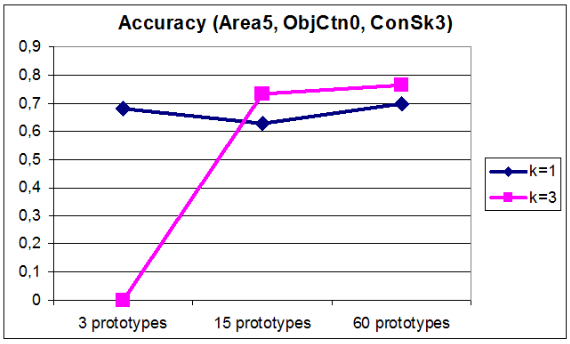


Fig. 6. Accuracy versus Prototypes and for two different feature subsets; Accuracy for different number of prototypes using 3 features (Area5, ObjCtn0, ConSk3)

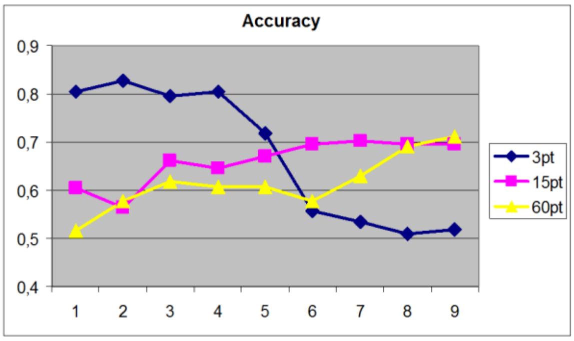


Fig. 7. Accuracy depending on choice of features (k=1)

Table 1. Difference values between 3 Prototypes using the 3 features (ObjCnt0, ArSig0, ObjCnt1) and the judged difference values by the expert

	B6_23	B03_22	F10_2
B6_23	0	0,669503257 (0,8)	0,989071038 (0,6)
B03_22	0,669503257 (0,8)	0	0,341425705 (0,9)
F10_2	0,989071038 (0,6)	0,341425705 (0,9)	0

Table 1 shows the difference values of three prototypes and in clips the judged difference values by the expert. The result shows that accuracy can be improved by applying the adjustment theory and especially the class specific quality is improved by applying the adjustment theory (see Fig. 8).

The application of the methods for larger samples set did not bring any significant reduction in the number of prototypes (see Fig. 9) or in the feature subset (see Fig. 10).

The prototype selection method reduced the number of prototypes only by three prototypes. We take it as an indication that we have not yet the enough prototypes and that the accuracy of the classifier can be improved by collecting more prototypes.

In Summary, we have shown that the chosen methods are valuable methods for a prototype-based classifier and can improve the classifier performance. For future work we will do more investigations on the adjustment theory as a method to learn the importance of features based on a low number of features and for feature subset selection for a low number of samples.

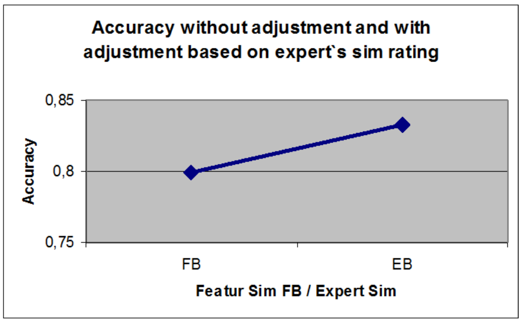


Fig. 8. Accuracy with and without adjustment theory

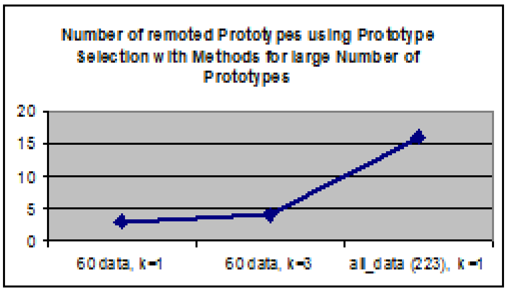


Fig. 9. Number of removed Prototypes

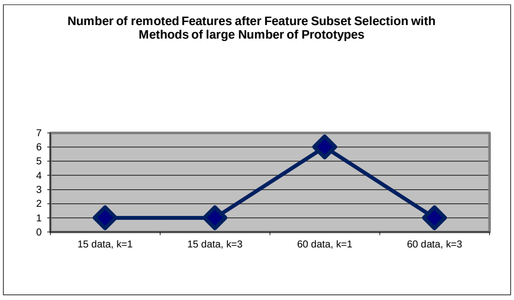


Fig. 10. Number of removed Features after Feature Subset Selection

6 CONCLUSIONS

We have presented our results on a prototype-based classification. Such a method can be used for incremental knowledge acquisition and classification. Therefore the classifier needs methods that can work on low numbers of prototypes and on on large numbers of prototypes. Our result shows that feature subset selection based on the discrimination power of a feature is a good method for low numbers of prototypes. The adjustment theory in combination with an expert similarity judgment can be taken to learn the true feature range in case of few prototypes. If we have a large number of prototypes an option for prototype selection is needed that can check for redundant prototypes.

The system can start to work on a low number of prototypes and can instantly collect samples during the usage of the system. These samples get the label of the closest case. The system performance improves the more prototypes the system has in its data base. That means an iterative process of labeled sample collection based on prototype based classification is necessary, followed by a revision of these samples after some time, in order to sort out wrongly classified samples until the system performance has been stabilized.

The test of the system is done on a new application on cell image analysis, the study of the internal mitochondrial movement of cells.

References

1. R. Schmidt and L. Gierl, "Temporal Abstractions and Case-Based Reasoning for Medical Course Data: Two Prognostic Applications, " in Machine Learning and Data Mining in Pattern Recognition, MLDM2001, edited by P. Perner, lnai 2123, Springer-Verlag: Berlin Heidelberg, p. 23-34, 2001.
2. Perner, P.: *Prototype-Based Classification*, Applied Intelligence 28, 238-246 (2008)
3. M. Nilsson and P. Funk, "A Case-Based Classification of Respiratory Sinus Arrhythmia," in Advances in Case-Based Reasoning, ECCBR 2004, edited by P. Funk and P.A. Gonzalez Calero, lnai 3155, Springer-Verlag: Berlin Heidelberg, p. 673-685, 2004.
4. D.W. Aha, D. Kibler, and M.K. Albert, "Instance-based Learning Algorithm," Machine Learning, 6(1):37-66, 1991.
5. P. Perner, "Incremental Learning of Retrieval Knowledge in a Case-Based Reasoning System," in: K.D. Ashley and D.G. Bridge (Eds.), Case-Based Reasoning - Research and Development, Springer Verlag 2003, LNAI 2689, pp. 422-436
6. Sachs-Hombach, Kl.: *Bildbegriff und Bildwissenschaft*. In: Gerhardus, D., Rompza, S. (Eds.) kunst - gestaltung - design, Heft 8, pp. 1-38, Verlag St. Johann, Saarbrücken (2002)
7. Krausz E., Prechtl, St., Stelzer, E.H.K., Bork, P., Perner, P.: *Quantitative Measurement of dynamic time dependent cellular events*. Project Description (May 2006)
8. Chang, C.-L.: Finding Prototypes for Nearest Neighbor Classifiers. IEEE Trans. on Computers. C-23 (11) (1974)
9. Perner, P.: *Data Mining on Multimedia Data.* LNCS, vol. 2558, Springer Verlag (2002)

10. Little, S., Colantonio. S., Salvetti, O., Perner, P.: Evaluation of Feature Subset Selection, Feature Weighting, and Prototype Selection for Biomedical Applications. J. Software Engineering & Applications 3, 39-49 (2010)
11. Niemeier; W.: Ausgleichsrechnung, de Gruyter, Berlin New York (2008)
12. Perner, P.: *Novel Computerized Methods in System Biology -Flexible High-Content Image Analysis and Interpretation System for Cell Images.* In: Perner, P. Salvetti, O. (Eds.) Advances in Mass Data Analysis of Images and Signals in Medicine, Biotechnology, Chemistry and Food Industry, MDA 2008, lnai, vol. 5108, pp. 139-157, Springer Verlag (2008)