

A Novel Visualization Approach for Data-Mining-Related Classification

Christin Seifert Elisabeth Lex
Know-Center Graz
{cseifert,elex}@know-center.at

Abstract

Classification and categorization are common tasks in data mining and knowledge discovery. Visualizations of classification models can create understanding and trust in data mining models. However, existing visualizations are often complex or restricted to specific classifiers and attributes. In this work, we propose an intuitive visualization system to observe and understand classification processes and results. Our system can handle multiple classes, nominal and numeric attributes, and supports all classifiers whose predictions can be interpreted as probabilities. We state that the possibility to observe the training process of a classifier boosts the understanding of classification results also for non-expert users. In combination with an intuitive visualization, we provide a system to generate in-depth understanding of classification processes and results. Our simulations revealed that the system could support the user to better understand a classifier's decision, and to gain insights into classification processes.

1. Introduction

Information Visualization is widely used in knowledge discovery and data mining to show the results of mining processes. In these fields, visualizations support users to comprehend results of data mining algorithms. However, they cannot clarify why a certain decision was made by the data mining model. For this, not only the results need to be visualized but also the underlying data mining model. Visualizing models generates trust and understanding for users [17] and was identified as one of the key challenges in data mining [11]. A core technique of data mining is classification. Classification predicts categories for unseen data samples based on previously seen samples with known category. In order to understand a classification result, four details are crucial: (i) How many training samples were used to build the current classifier? How does the classifier label the test examples? (ii) How well does the classifier perform? (iii) Are there any conspicuousities regarding

the classes or the samples? (iv) What was the decision for a sample based on? Additionally, observing a classification procedure could enable the user to answer questions like: How many training samples are necessary to get a stable classifier? Are there any “problem” samples for which the classifier constantly changes its decision?

In previous research, e.g. [1, 6, 15], several visualizations were proposed to depict classification models. These visualizations are either model-specific or generic. Model-specific visualizations are tailored to a particular classification model and can therefore be adapted exactly to the model's properties. However, the classification model cannot be exchanged on demand. For example, a Naive Bayes classifier cannot be substituted with a Decision Tree even when a Decision Tree would probably perform better on the data. Examples for model-specific visualizations show results of Naive Bayes classifiers [2], Decision Tables [3], Decision Trees [1], and Support Vector Machines [7]. Generic visualizations are not restricted to a specific classification model and the underlying classifier can be exchanged if necessary. Yet, the generic visualizations proposed in previous research are applicable only to certain classification models. Examples for such models are additive [14], associative [6] and binary probabilistic classifiers [15]. In [14] a visualization of additive classifiers is proposed. The system visualizes the classifier decisions, the evidence for the decisions, and the sources of the evidence. Thus, it allows the user to precisely understand the decision of a classifier. However, the system is restricted to only a small subset of classifiers and well-established models like Decision Trees and non-linear Support Vector Machines are not supported. Chodos and Zaiane proposed a visualization of associative classifiers [6] and their rule sets. Yet, the supported classification models are outperformed by more recent approaches [12] and they are limited to only nominal attributes. For binary probabilistic classifiers, Rheingans and desJardins [15] introduced a visualization based on self-organizing maps (SOM). This visualization can be used for classification models whose output can be mapped to a probability distribution. However, two problems remain: first, the visualization supports only binary classifica-

tion. Second, the SOM-based visualization imposes a high cognitive load to the user [15].

In our work, we developed a system to visualize classification models and to display classification results. Our visualization component is inspired by the RadViz feature visualization of Hoffman [9, 10]. The aim of our work is to provide a simple and intuitive system to comprehend classification processes and results even for users with no strong background in the field of data mining. Due to the shortcomings discussed earlier, we develop a system that can handle multiple classes and nominal and numeric attributes. Also, our system supports all classifiers whose predictions can be interpreted as probabilities. We state that the possibility to observe the training process of a classifier boosts the understanding of classification results also for non-expert users. In combination with an intuitive visualization, we aim to provide a system to generate in-depth understanding of classification processes and results.

The remainder of the paper is structured as follows: Section 2 provides a brief introduction into classification as a supervised machine learning technique. Section 3 describes the proposed visualization. In Section 4, we apply the visualization to an example dataset for an object recognition task. Finally, we conclude the paper and propose future work.

2. Supervised Machine Learning

Algorithms of supervised machine learning learn a function from items in a dataset, called the training set, whereas each item is assigned a target label. The learned function either outputs a continuous value (regression) or a predicted label for an item (classification) [5]. Most common supervised machine learning algorithms are Support Vector Machines, Naive Bayes, K-Nearest Neighbor and Decision Trees (see [12] for a review of well-known methods of supervised machine learning). In our work we focus on classifiers that either output a probability distribution or whose outputs can be mapped to a probability distribution. The most likely label for an item is defined by the class with the highest probability (Maximum a Posteriori, MAP). A probability distribution for a sample i is given by $p^i = (p_1^i, \dots, p_c^i)$, $\sum_{j=1}^c p_j^i = 1$ whereas c corresponds to the number of classes. For classification we use three data sets, a training set, an evaluation set, and a test set. The classifier is computed on a training set that contains labeled items. The purpose of the evaluation set is to measure the quality of the classifier by common statistical means such as accuracy, precision, and recall by comparing the classifier output to the label stored in the data set. For an overview of the applied measures refer to [16]. In order to get a prediction of the performance for unseen data items, the items of the evaluation set should not be contained in the train-

ing set. The test set contains unlabeled items that should be labeled by the classifier.

3. Classification Process Visualization

The Classification Process Visualization consists of the proposed (static) visualization of classification results, the Class Radial Visualization, and mechanisms for a detailed interactive analysis. The single components are described in the next sections.

3.1. Class Radial Visualization

In the Class Radial Visualization all class items are displayed equally distributed around the perimeter of a circle. Each class item is represented by a square and its border is drawn in the unique class color. Initially, all squares are unfilled. Then, the square of the class with the maximum number of assigned test items is completely filled with its class color. The squares of all other classes are filled with their colors proportionally to the number of assigned items. Currently, the class items are placed according to the order of the classes in the input set. The trained classifier is used to calculate the a-posteriori probability distribution for each test item. Each test item is initially positioned in the center of the circle. The direction vectors from the test item to all class items are computed. The initial length of these vectors is set to the radius of the circle. Then, each direction vector is weighted by the according class probability. The vector sum of the weighted direction vectors yields the final position of the test item. A test item is then colored according to the class with the highest confidence. For example, a test item with the probability distribution $p = (p_1, \dots, p_c)$, $p_i = 1$ and $p_j = 0 \forall j \neq i$, $1 \leq i, j \leq c$ lies exactly on the location of the class item associated with class i (c denotes the number of classes). In this case the classifier is 100% sure that the test item belongs to class i . On the contrary, a test item whose a-posteriori probability corresponds to a uniform distribution is placed in the center of the circle. This indicates that the classifier cannot assign the item to a class. The color of the item is usually defined by the class with maximum confidence. If the probability distribution does not have a unique maximum, the class of the first maximum found defines the color of the item.

Yet, the Class Radial Visualization can be ambiguous due to a dimensionality problem. The visualization space is two-dimensional in contrast to the dimensionality of the probability distribution space that depends on the number of classes c . Clearly, for $c > 2$ the mapping from the probability distribution to the two-dimensional visualization space cannot be an injective function. More specifically, different probability distributions can map to the same

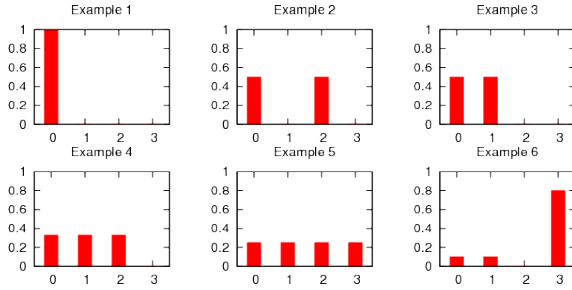


Figure 1. Histograms of different examples in the 4-class case.

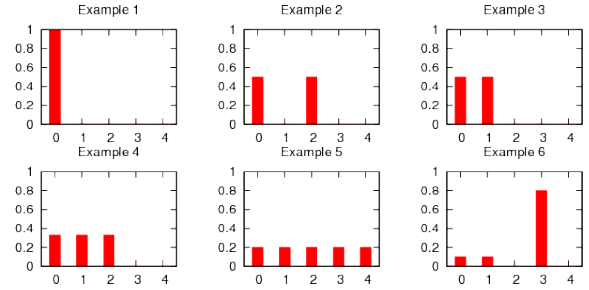


Figure 2. Histograms of different examples in the 5-class case.

position in the visualization space. This ambiguity is illustrated in the following examples: Figure 1 depicts six example distributions for $c = 4$, and Figure 2 for $c = 5$. The x-axes show the class labels and the y-axes the corresponding classifier confidences. In Figure 3(a) the layout for the examples of Figure 1 is shown. It can be seen that Example 2 ($p = (0.5, 0, 0.5, 0)$) and Example 5 ($p = (0.25, 0.25, 0.25, 0.25)$) map to the same location, the center of the circle. However, when comparing the distributions for Example 2 and 5, it is obvious that the classifier is more unsure for Example 5: Example 5 is assigned to all four classes with equal confidence, whereas Example 2 is assigned to only two classes with equal confidence. In Figure 3(b) Examples 2 and 5 for $c = 5$ are shown. With five classes the visualization of Examples 2 and 5 is no longer ambiguous, because they map to different locations. Yet, other probability distributions might exist that exhibit ambiguities. Examples that can be reliably predicted (e.g. Examples 1 and 6) are clearly mapped next to the according class item.

To sum up, the examples show that the visualization enables the user to distinguish between items that could be clearly classified versus items with no confident prediction. Yet, in the latter case, the visualization does not reflect all degrees of the classifier’s uncertainty. Hoffman [8] also reports the ambiguity problem and proposed several variants of the RadViz visualization, which, for a given dataset, could reduce the overlapping problem. However, the general problem of mapping a higher dimensional space to a lower-dimensional still remains. In our work, we resolve the problem of ambiguity and extend our system by adding interaction mechanisms. It displays additional information when the user moves the mouse over a test item. Lines are drawn from the test item to all class items, while line thickness indicates the confidence for the respective class. Note that a line to a class is only drawn when the confidence value is above a given threshold θ . For our simulations we set $\theta = 0.1$.

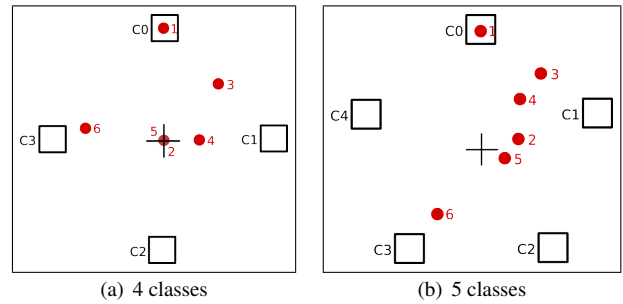


Figure 3. Layout for the examples histograms of Figures 1 and 2

3.2. Visualization System

The Class Radial Visualization is integrated in a visualization system that provides interactive mechanisms for a deep analysis of classification results and procedures. Figure 4 shows screenshots of the application. A typical workflow is: The user loads a training set, a test set, and an evaluation set into the system. If the test data includes class labels it can also be used as an evaluation set. Then, the user chooses a classification model (KNN, SVM, ..) and a step size. The step size denotes the number of samples the classifier is trained on in each step. After each training step the classifier is evaluated on the full evaluation set, the visualization is drawn, and the performance values and plots are updated.

The visualization system provides insights into classification results and processes on three levels, (i) the classifier level, (ii) the class level, and (iii) the test item level. First, on classifier level, the Class Radial Visualization gives a visual overview of the classifier performance. For example, a chaotic mix of colors for the test items distributed randomly over the area indicates a bad classifier performance. In addition to the Class Radial Visualization, common performance measures for classifiers calculated on the evaluation dataset are displayed (right-hand side of Figure 4(a)).

The contingency table shows the classifier output related to the expected values (ground truth). Further performance measures such as accuracy, precision and recall for the current classifier are listed (for an explanation of the measures see [16]). The plot shows the selected measure over the number of training samples and also includes curves for the trivial rejector and acceptor - virtual classifiers that assign all evaluation items to most/least occurring class. The advantage of the visualization is that it can be instantly comprehended by the user, yet it is less accurate than the performance measures and plots. Second, on class level, we show general information and performance measures for each class when the mouse pointer is moved over a class item in the Class Radial Visualization. For an example, refer to Figure 5(a). Third, on test item level, the spatial position of an item in the visualization indicates how confident it could be classified. As this visualization is ambiguous by nature (as described in Section 3.1), the a-posteriori probability values are displayed by lines to the corresponding class item when moving the mouse pointer over the item. An example of this is shown in Figure 4(a). Further, we show the original data of the test item when moving the mouse pointer over the item. For example, in Figure 4(a) the original data is depicted by the image in the center containing the bottle. It depends on the data whether showing original data is feasible and usable: data for image recognition tasks can be presented easily, whereas for other scenarios like text classification it is necessary to find an appropriate data representation.

4. Simulations and Results

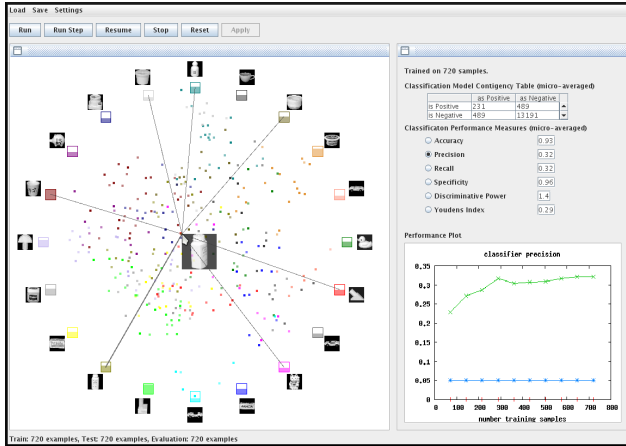
For our simulations we have chosen an object recognition task. A classifier should predict object labels for unseen images given a set of training images that contain different objects. We used the image database COIL-20 [13], which is a standard object recognition database. The database consists of 1440 images, 72 images for each of the 20 different objects. Each image shows the object from a different angle, while the angle is increased by 5 degrees in subsequent views. We split the database in training and test set by taking the 0, 10, 20, ..., 350 degree image for training and the 5, 15, 15, ..., 355 degree image for testing. We also have the correct class label information for the test set, so we used this data set for evaluation. We tested two different feature transformation methods: PCA [18] and Fisher-LDA [4] and two different classification models: K-Nearest Neighbor (KNN) with $K = 10$ and Naive Bayes (NB). For the feature transformation methods, LDA features are known to outperform PCA features for object recognition tasks. The aim of our simulations is to find out whether this is reflected in the visualization. Regarding the classifiers, KNN and NB perform similar in general, it depends on the data

at hand which model yields better results [12]. We assume that with our visualization we gain insights into the classifier performance as well as the classification process. For the classifiers, we used the implementation of the WEKA machine learning library [19].

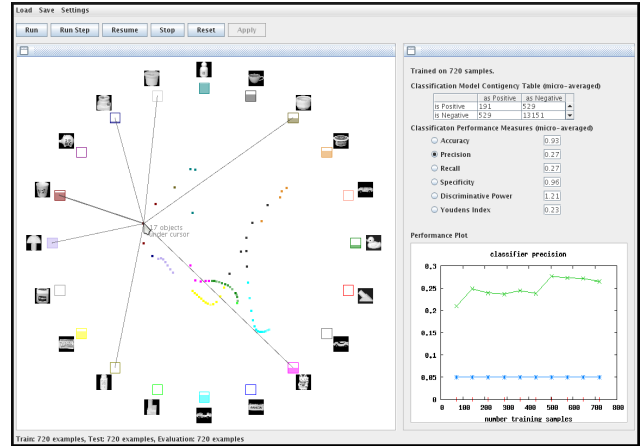
For each of the four trials – PCA+KNN, PCA+NB, LDA+KNN, LDA+NB – we set the step size to 72 items (10% of the training set) and apply our procedure stepwise until the classifier has seen all trainings items. The screenshots in Figures 4 and 5 show the resulting visualization for all combinations at the end of training.

Figure 4(a) illustrates the results for training the KNN classifier on the PCA features. It is obvious that the classifier cannot learn a reliable model from the PCA data. The items are widely distributed and the colors are mixed. This indicates that the classifier cannot confidently predict most of the items. In Figure 4(a) an item is selected and via mouse over its image (the original data) and its class probabilities are shown. A user can instantly see that the classifier was not able to correctly assign this item to one class because seven lines are drawn. The visual information on classifier level correlates well with the measures calculated on the evaluation set (given on the right-hand side of the figure). A precision of only 0.32 is reached for this trial. Figure 4(b) shows the result for the NB classifier on the PCA feature set. Although there seems to be less chaos in the Class Radial Visualization compared to the KNN case, the NB classifier also has a low precision value of 0.27. In the NB case, the visual information on classifier level does not correlate well with the performance measure results. Therefore, to correctly assess the classifier performance it is necessary to also consider the class level and the test item level: It can be seen that the squares of nine out of 20 classes are not filled at all. This means that no test item is labeled with these classes. Also, many test items are mapped to the same location which can be seen via mouse over.

In contrary, the classification based on LDA features generates a completely different layout (see Figure 5). Almost all test items are placed near their corresponding class items and the center of the circle is empty. Both KNN and NB perform well on the LDA features. This can be observed in the Class Radial Visualization as well as in the performance table. The precision value is 0.88 for KNN and 0.89 for NB. From the performance plots on the right-hand sides of Figures 5(a) and 5(b) we conclude that NB needs less training items (72 items) than KNN (210 items) to reach the final precision on the evaluation set. Figures 5(a) and 5(b) also show the detailed information for a single class. Obviously the KNN missed some of the objects “cup-with-handle” (TP-rate = 0.69) but made no false assignment (FP-rate=0). The NB classifier found more correct objects “cup with handle” (TP-rate = 0.89) but classified at least one item incorrectly as “cup-with-handle” (FP-rate=0.01).

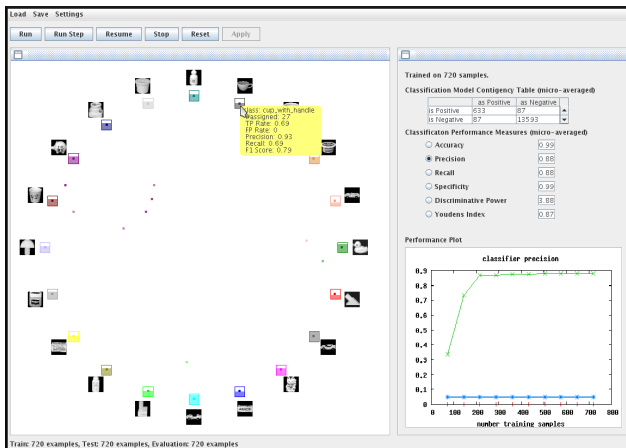


(a) PCA features + K-Nearest Neighbor classifier

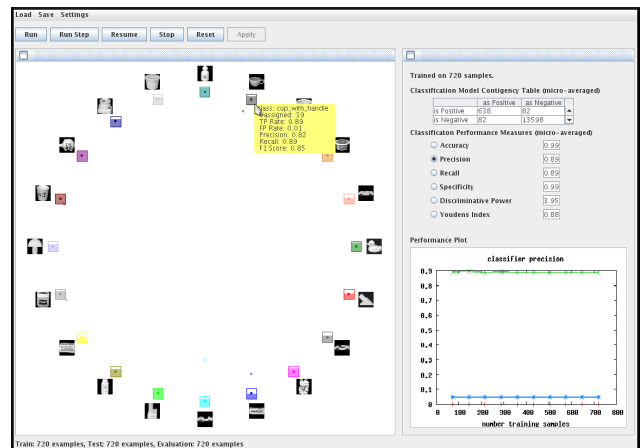


(b) PCA features + Naive Bayes classifier

Figure 4. Classifier visualization on PCA features for the COIL-20 database



(a) LDA + K-Nearest Neighbor classifier



(b) LDA + Naive Bayes classifier

Figure 5. Classifier visualization on LDA features for the COIL-20 database

To sum up, the simulations revealed that the visualization clearly reflects the difference between PCA and LDA features for the used dataset. The visualization also clarified that the KNN and the NB classifier perform quite equally on the LDA feature set. Furthermore, from the visualization we were able to conclude that the NB needs less training items than KNN.

5. Conclusion and Future Work

In conclusion, with our classification process visualization users can observe the classification process in detail. We state that our system can be used to assess a classifier's performance. Our simulations showed that to correctly assess the performance three levels need to be considered: the classifier level, the class level, and the test item level. The

classifier level gives an overall impression of the classifier. Test items placed close to their classes indicate a good classifier. A color mix in the center of the circle indicates a bad performance. However, our simulations showed that in some cases the visual information on classifier level can be insufficient to correctly assess the overall classifier performance, and more details are necessary. Therefore a user can also assess the classifier on class level and test item level. On class level, a user can derive how many items were assigned to a class and how well the classifier performs especially for this class. On test item level, the spatial position indicates how well the classifier could predict the class label. The prediction is determined by the a-posteriori probability that an item belongs to a particular class. The final prediction is denoted by the color of the test item. Also, on interaction, test items feature lines with varying thickness to

all class items, whereas the thickness is proportional to the particular class probability. Summing up, the test item level enables the user to better understand a classifier's decision.

We state that our visualization seems intuitive yet it is not clear how well the visualization reflects classifier performance gains or losses. We plan to conduct a user study to investigate how perceived performance correlates with the actual technical measurements. Currently, our system provides a set of standard performance measures and a performance plot to compare the visual output of the classification process with numerical values and contingency tables. Besides, our classification process visualization can handle multiple classes, nominal and numeric attributes and supports classifiers, whose output can be mapped to a probability distribution. As discussed, the output of the visualization can be ambiguous in some cases. We partly solve the ambiguity by drawing lines showing the origin of the largest spring forces acting on the item under mouse. Also, we will develop heuristics to change the sequence of the classes to minimize the ambiguity and integrate context and detail views by using magic lenses. Currently, the visualization is applicable to a limited number of classes and items. These limitations are imposed by color discrimination abilities of human vision and the limited space of the display. Further investigations are needed for larger data sets. In addition, we want to integrate user feedback into the classification model to enable non-expert users to tune classifiers. Appropriate mechanisms for feedback integration have to be developed, on the visualization side as well as on the classifier side.

6. Acknowledgment

The Know-Center is funded within the Austrian COMET Program under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

- [1] M. Ankerst, M. Ester, and H.-P. Kriegel. Towards an effective cooperation of the user and the computer for classification. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining (KDD)*, pages 179–188, New York, NY, USA, 2000. ACM.
- [2] B. Becker, R. Kohavi, and D. Sommerfield. Visualizing the simple bayesian classifier. In *KDD Workshop Issues in the Integration of Data Mining and Data Visualization*, 1997.
- [3] B. G. Becker. Research report: Visualizing decision table classifiers. In *Information Visualization*, Los Alamitos CA, 1998. IEEE Computer Society Press.
- [4] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, 1997.
- [5] C. M. Bishop. *Pattern Recognition and Machine Learning*. Number 978-0387310732. Springer-Verlag New York Inc., February 2008.
- [6] D. Chodos and O. Zaiane. ARC-UI: Visualization Tool for Associative Classifiers. In *Proc. Information Visualisation*, pages 296–301, July 2008.
- [7] D. Cook, D. Caragea, and V. Honavar. Visualization in classification problems. In *Proceedings of the COMPSTAT Symposium*, 2004.
- [8] P. Hoffman. *Table Visualizations: A Formal Model and its Applications*. PhD thesis, University of Massachusetts Lowell, 1999.
- [9] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley. DNA visual and analytic data mining. In *Proc. Conference on Visualization (VIS97)*, pages 437–441, 572, Los Alamitos, CA, USA, Oct 1997. IEEE Computer Society Press.
- [10] P. Hoffman, G. Grinstein, and D. Pinkney. Dimensional anchors: a graphic primitive for multidimensional multivariate information visualizations. In *Proc. Workshop on new Paradigms in Information Visualization and Manipulation at the ACM International Conference on Information and Knowledge Management (NPIVM99)*, pages 9–16, New York, NY, USA, 1999. ACM.
- [11] R. Kohavi. Data mining and visualization. Invited talk at the National Academy of Engineering US Frontiers of Engineers (NAE), September 2000.
- [12] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. *Informatica*, 31(3):249–268, 2007.
- [13] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-20). Technical Report CUCS-005-96, Department of Computer Science, Columbia University, February 1996.
- [14] B. Poulin, R. Eisner, D. Szafron, P. Lu, R. Greiner, D. S. Wishart, A. Fyshe, B. Pearcy, C. MacDonell, and J. Anvik. Visual explanation of evidence in additive classifiers. In *Proc. Conf. on Innovative Applications of Artificial Intelligence (IAAI06)*, pages 1822–1829, Boston, MA, July 2006.
- [15] P. Rheingans and M. desJardins. Visualizing high-dimensional predictive model quality. In *In Proceedings of IEEE Visualization*, pages 493–496, 2000.
- [16] M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond Accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. In *Advances in Artificial Intelligence (AI2006)*, LNAI 4304, pages 1015–1021, Berlin/Heidelberg, 2006. Springer.
- [17] K. Thearling, B. Becker, D. DeCoste, W. Mawby, M. Pilote, and D. Sommerfield. *Information Visualization in Data Mining and Knowledge Discovery*, chapter Visualizing data mining models, pages 205–222. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.
- [18] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, 1991.
- [19] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition edition, 2005.