Prog Artif Intell (2012) 1:291–301
DOI 10.1007/s13748-012-0023-9

REGULAR PAPER

“I can tell you what it’s not”: active learning from counterexamples

Nicolas Cebron · Fabian Richter · Rainer Lienhart

Received: 21 February 2012 / Accepted: 15 July 2012 / Published online: 4 August 2012

Abstract When dealing with feedback from a human expert in a classification process, we usually think of obtaining the correct class label for an example. However, in many real-world settings, it may be much easier for the human expert to tell us to which classes the example does *not* belong. We propose a framework for this very practical setting to incorporate this kind of feedback. We demonstrate empirically that stable classification models can be built even in the case of partial not-label information and introduce a method to select useful training examples.

Keywords Classification · Human feedback · Active learning

1 Introduction

The goal of supervised classification is to deduce a function from examples in a dataset that maps input objects to desired outputs. By using a set of labeled training examples, we can train a classifier that can be used to predict the nominal target variable for unseen test data. To achieve this, the learner has to generalize from the presented data to unseen situations.

While a plethora of algorithms for supervised classification has been developed, only a few works deviate from this classical setting. However, finding the correct class label for an example can be difficult—especially when there is a large number of classes. In the work of [15], it has been shown that the human expert error rate and the time needed to find the correct label grows with the number of classes; at the same time the user distress increases. In some situations, it might not even be possible for the human expert to determine the correct class label out of many possible class labels. In a normal classification setting, we would have to ignore this example.

As an example, consider the domain of image classification, where we have two common situations in which the human expert has problems providing the correct class label:

1. Ambiguous information: different class labels may be possible, but there is a lack of information to choose explicitly one of them. For example, it is unclear whether a person is playing baseball or softball when the ball itself is not present in the image.
2. Rare cases: the determination of the class label may be difficult because of missing expertise. For example, it may be difficult to classify a pangolin or a solenodon[1] in an image.

In this work, we consider a special setting in supervised classification: we do not obtain the label information itself, but the labels of the classes that this example does *not* belong to. We call the labels of those examples ‘¬labels’ and the tuple of example and ¬labels a ‘counterexample’. For the preceding examples, it can be very easy to specify the sports that are not present (not tennis, not soccer, etc.) or the animals that are not displayed in the image (not a cat, not a dog, etc.). We argue that in many real-world settings, it is much easier for the human expert to specify the classes to which

N. Cebron (✉) · F. Richter · R. Lienhart
Multimedia Computing Lab, University of Augsburg,
86162 Augsburg, Germany
e-mail: ncebron@gmail.com; cebron@informatik.uni-augsburg.de

F. Richter
e-mail: richter@uni-augsburg.de

R. Lienhart
e-mail: lienhart@uni-augsburg.de

[1] Both animals are very rare mammals.

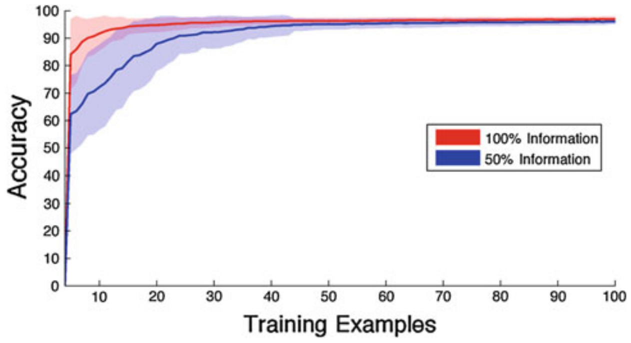


Fig. 1 Accuracy as a function of an increasing number of training examples. *Blue line* 50 % of ¬label information available, *red line* 100 % of ¬label information available (color figure online)

the example does not belong. Within the framework of classification with ¬labels, we enable the learning algorithm to gain information from examples in the dataset that are difficult to label. However, we keep our framework open and the information of labels and ¬labels is included seamlessly. In a classification problem with k disjoint classes, there is of course a loss of information induced from this setting, as we can expect to observe less than $k - 1$ ¬labels for each example.[2] How much does this loss of information influence the resulting classification model?

Figure 1 shows a toy example with the Iris dataset from the UCI machine learning repository [2]. There are three classes in this dataset. We have trained a classifier with 100 % of information (which corresponds to two ¬labels for each example) and with 50 % of information (corresponding to one randomly chosen ¬label for each example). We plot the accuracy of each training set with an increasing number of randomly chosen examples. The experiment has been repeated 100 times and we plot the mean value and the SD. We can gain two insights from this example:

1. We can obtain an almost equivalent model in terms of accuracy—even if we throw away a significant amount of information.
2. We need more training examples to reach the same level of accuracy.

The second point is quite obvious, because we cannot generate a good classification model in the first iterations with less information. However, in later iterations, a sparse classification model (that is used in this work) can outbalance the loss of information.[3]

Within this setting of a new form of feedback comes the question which examples contribute most to the classification model. Choosing randomly drawn examples from the dataset helps to improve classification accuracy, but may need a large number of iterations to converge. Instead of picking redundant examples, it would be better to pick those that can ‘help’ train the classifier. The concept of active learning tackles this problem by enabling a learner to pose specific queries that are chosen from an unlabeled dataset. In the active learning setting, we assume that we have access to a (noiseless) human expert who is able to predict the correct class label of a sample [7,12,17,22,24]. In this work, we will make use of some ideas in this field in order to select examples that contribute most within our new framework.

We will revise existing work in Sect. 2 and introduce the support vector domain description (SVDD) technique in Sect. 3, which forms the basis for our classification framework with ¬labels (Sect. 4). We will develop different strategies for selecting training examples in this framework in Sect. 5 and compare their performance in Sect. 6 before drawing conclusions in Sect. 7.

2 Related work

To the best of our knowledge, there is only one work that considers feedback in the form of ¬labels in a semi-supervised learning setting [14]. The authors propose a negative label propagation scheme based on the assumptions that nearby examples share the same class label. They leave the point of selecting negative examples open for future research. Some works have considered negative feedback in the image retrieval process [1,18]. As the retrieval process corresponds to a two-class problem, these works only share the general idea of a different form of feedback with this work. At first sight, our work seems to be related to the domain of multi-label classification [25], where a mapping from an example to a set of class labels is sought. In contrast to this work, our set of ¬labels inhibits a specific structure and our final goal is to predict *one* class label from the set of ¬labels. There is also a relation to the topic of learning from partial class labels [9,19], where the algorithm is given a candidate set of labels, only one of which is correct. There is a direct relation between mapping from a candidate set of labels to a set of ¬labels. However, our type of feedback is fundamentally different in an interactive setting and it results in a completely new learning framework and a new selection strategy to query new examples for labeling.

The field of active learning has dealt with the selection strategy for over two decades. An active learner is described

[2] If we have $k - 1$ ¬labels, we can deduce the corresponding label directly.

[3] Sparsity is a very useful property of some Machine Learning algorithms. Such an algorithm yields a sparse result when, among all the coefficients that describe the model, only a small number are non-zero.

Footnote 3 continued
Within our framework, this means that only a small subset of examples is used for classification.

by its underlying classifier and its query function. The classifier is trained on the labeled data. The query function makes a decision based on the current model as to which samples from the unlabeled data pool should be chosen for labeling. In each iteration, samples are chosen from the unlabeled pool and get labeled by the human expert. The classifier is trained on the current labeled dataset. This process continues until some stopping criterion has been met. We can categorize the existing active learning approaches by their selection strategy:

Optimization of a target function Based on the minimization of the expected error function (or maximization of a likelihood function), examples can be selected by their contribution to this function. Popular approaches in this field are the works of [8,16,17,21]. From a theoretical point of view, the explicit definition of a target function that should be minimized makes it easy to analyze the selection strategy. However, these approaches make several assumptions (e.g., that a stable model built with randomly chosen examples already exists or that the learner does not have a bias [8]); therefore, the outcome of the selection strategy depends on how much these assumptions apply.

Reduction of version space The goal of this approach is to reduce the version space with a selected sample as much as possible. One of the most popular approaches is the Query by Committee algorithm [12], which uses a committee of diverse but consistent hypotheses and queries, examples for which the disagreement is maximal. Another approach imitates the most general and most specific hypothesis with a neural network and queries examples at the region of uncertainty between those two hypotheses [7]. In the work of [24], the parameter space of a support vector machine (SVM) is related to the version space in order to derive several strategies to query new examples.

Uncertainty sampling This heuristic approach focuses on selecting examples at the classification boundary. The most popular approaches use an SVM and query examples at the decision hyperplane in the kernel-induced space [5,22] similar to one of the version space reduction approaches described by Tong and Koller [24]. Uncertainty sampling is prone to selecting outliers. Like all other approaches, it relies on a stable classification model that has been initialized with some randomly chosen examples.

In more recent works in the field of active learning, one can observe the trends toward making these schemes more robust by using meta techniques to balance strategies for exploration and exploitation [4,6,20] and focusing on the theoretical aspects and benefits [3,10] of active learning.

3 Support vector domain description

In this section, we introduce the SVDD from [23] that is usually used for anomaly detection. We will use this technique to build a committee of classifiers in the next section. The training dataset comprises N input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$. Let $\phi(\mathbf{x})$ denote a fixed nonlinear feature-space transformation. The goal of SVDD is to find the smallest enclosing hypersphere with center $\mathbf{a}$ and radius R in the mapped feature space for the given data points:

$$\begin{aligned} &\min R^2 + C\sum_{n=1}^{N}\xi_n \\ &\text{s.t.}||\phi(\mathbf{x}_n) - \mathbf{a}^2|| \leq R^2 + \xi_n \\ &\quad \xi_n \geq 0, \quad \text{for } n = 1, \ldots N \end{aligned} \tag{1}$$

where ξ_n are the slack variables for all data points. The goal is to minimize the radius while softly penalizing data points that do not lie inside the hypersphere. The parameter C controls the trade-off between the slack variable penalty and the radius. We note that when the sphere is mapped back to the original feature space, it may represent several clusters of data points. Figure 2 provides an overview of the SVDD technique.

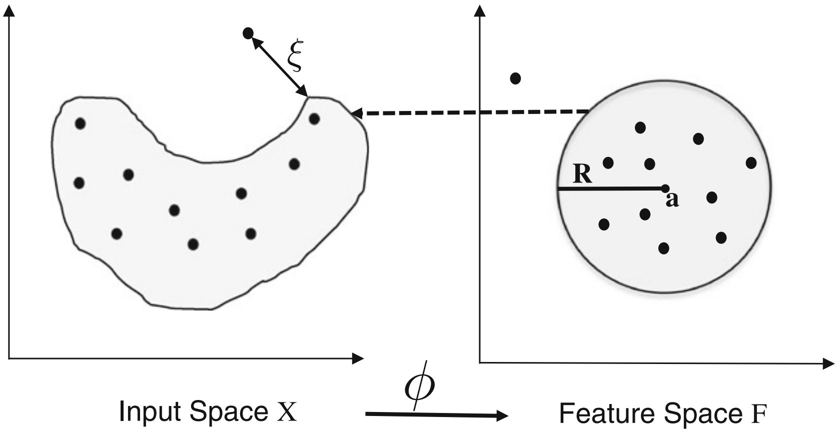


Fig. 2 SVDD: the dataset is mapped via ϕ to a feature space F, where we seek the smallest enclosing hypersphere with center a and radius R

The corresponding Lagrangian to this quadratic problem is given by:

$$L(R, \mathbf{a}, \xi) = R^2 - \sum_{n=1}^{N}(R^2 + \xi_n - ||\phi(\mathbf{x}_n) - a||^2)\beta_n - \sum_{n=1}^{N}\xi_n\mu_n + C\sum_{n=1}^{N}\xi_n \tag{2}$$

Setting the derivatives of L with respect to R and $\mathbf{a}$ to zero, we obtain the following relations:

$$\sum_{n=1}^{N}\beta_n = 1 \qquad \mathbf{a} = \sum_{n=1}^{N}\beta_n\phi(\mathbf{x}_n) \tag{3}$$

These relations can be used to reformulate the problem in terms of the variables β_n and therewith the solution of this dual problem is given by:

$$\begin{aligned} \max \tilde{L}(\boldsymbol{\beta}) &= \sum_{n=1}^{N} K(\mathbf{x}_n, \mathbf{x}_n)\beta_n - \sum_{m=1}^{N}\sum_{n=1}^{N}\beta_m\beta_n K(\mathbf{x}_m, \mathbf{x}_n) \\ &\text{s.t. } 0 \leq \beta_n \leq C \\ &\sum_{n=1}^{N}\beta_n = 1, \quad n = 1, \ldots, N \end{aligned} \tag{4}$$

with the kernel function $K(\mathbf{x}_m, \mathbf{x}_n) = \phi(\mathbf{x}_m) \cdot \phi(\mathbf{x}_n)$.[4] The points with $0 < \beta_n < C$ define the boundary of the hypersphere and are called the support vectors. The distance of the image of $\mathbf{x}$ from the sphere center is given by:

$$R^2(\mathbf{x}) = ||\phi(\mathbf{x}) - \mathbf{a}||^2 = K(\mathbf{x}, \mathbf{x}) - 2\sum_{n=1}^{N}\beta_n K(\mathbf{x}_n, \mathbf{x}) + \sum_{m=1}^{N}\sum_{n=1}^{N}\beta_m\beta_n K(\mathbf{x}_m, \mathbf{x}_n) \tag{5}$$

The domain that describes the support of the data points is then given by $\{\mathbf{x} : R^2(\mathbf{x}) = R^2(\mathbf{x}_i)\}$ for any support vector $\mathbf{x}_i$. In our work, we generate the output of a classifier as $f(\mathbf{x}) = R^2(\mathbf{x}) - R^2(\mathbf{x}_i)$. If a data point is outside the domain, we get a positive value, and if it is inside the domain we obtain a negative value.

4 Not-label classification framework

We now introduce the framework for the classification with ¬labels. In this setting, we assume that each of the N input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is associated with a vector $\mathbf{y} = (y^1, \ldots, y^k)$, where each entry $y^j \in \{0, 1\}$ indicates whether we know that this input vector does *not* belong to class j (1) or that we do not have any information concerning class j for this input vector (0). Our goal is to obtain a classifier that is able to map new examples to the correct class. However, our training dataset examples only contain partial information in the form of ¬labels, which renders a learning process of a direct function from examples to classes impossible. Instead, we propose to model each ¬label from the training examples separately with an SVDD classifier. The result is a committee of k SVDD classifiers, whose outputs need to be combined to deduce the correct label in the testing phase. The detailed procedure of our proposed method is as follows:

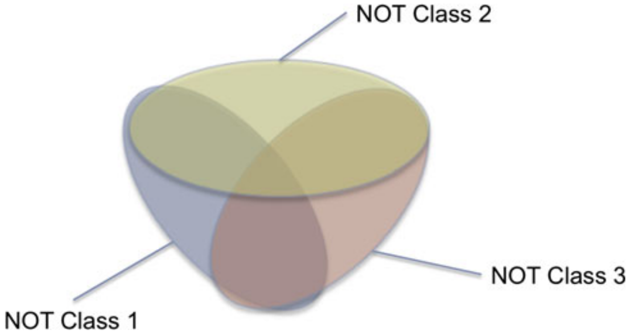


Fig. 3 Overlapping ¬labels implicate the class label

Step 1: Data partitioning: we first partition the training data into k subsets $\{D_c\}_{c=1}^{k}$ according to their ¬labels. The cth dataset D_c contains N_c elements from input vectors that do *not* belong to class c.

Step 2: SVDD committee: for each training set D_c, we train an SVDD according to the description in Sect. 3.

Step 3: Final classification: let $\mathbf{f} = (f^1(\mathbf{x}), \ldots, f^k(\mathbf{x}))$ denote the output vector that contains the outputs of the k One-Class SVM's for an input vector $\mathbf{x}$. The final classification decision from the ¬label outputs is obtained by choosing $c = \arg\min_c f^c(\mathbf{x})$.

Figure 3 illustrates the framework for classification with ¬labels. Our goal is to define regions in the data space for each ¬label with the use of SVDDs. These regions should have maximal overlap of (ideally) $k - 1$ ¬labels so that we can deduce the corresponding class label in this region with high confidence. If we have an overlap of k ¬labels or an overlap of less than $k - 1$ ¬labels in a region of the data space, the correct class label can still be deduced correctly as long as the distances (or the outputs of the SVDDs) to the other ¬labels are bigger than the distance to the correct label.

We can easily integrate normal class labels in this framework by adding the same example multiple times to the

[4] In this work, we make use of the Gaussian kernel function which is given by: $K(\mathbf{x}_m, \mathbf{x}_n) = \exp(-||\mathbf{x}_m - \mathbf{x}_n||^2/2\sigma^2)$.

training dataset with all values of the labels it does *not* belong to. However, this results in a linear increase of the training set size. If we let $l \in [0, 1]$ denote the percentage of examples with normal class labels in the training dataset, the increase is $l \cdot n \cdot (k-1)$.

5 Active selection of training examples

In a normal classification setting, we can select an example and obtain a class label. The difficulty in our new setting lies in the fact that we do not know which ¬labels we are going to obtain from the human expert. Any number of ¬labels between one and $k-1$ contains information for the classification model. We assume that each ¬label is equally likely.

We propose three different selection strategies in this framework. The first two are based on the active learning literature and the third one is based on our own observational research.

5.1 Random sampling

We use random sampling as a baseline for the other methods. It is independent of the underlying classification algorithm. By randomly choosing an example in each iteration, each example in the dataset has an equal chance of being selected, which makes random sampling usually a very good base line.

5.2 Uncertainty sampling

Uncertainty sampling in active learning with SVMs usually refers to the process of selecting the example that is closest to the decision boundary [22,24]. For multiclass problems, we need the class probabilities in order to measure the uncertainty in the classification decision. As the estimation of probabilities for one class SVMs is still an open research issue, we use the outputs of the committee and make sure that they sum to one. There are two measures commonly used to measure the degree of uncertainty in multiclass problems:

Entropy is a well-known measure from the domain of information theory to determine the disorder of a system. Let $p_c(\mathbf{x})$ denote the probability for class c for an example $\mathbf{x}$. Entropy is defined as $-\sum_{c=1}^{k} p_c(\mathbf{x}) \cdot \log(p_c(\mathbf{x}))$. It generalizes the disagreement as defined for binary classification to the multiclass case. However, we cannot use entropy in our setting as we do not have the class probabilities/class outputs. We could use the outputs of the committee as the ¬label probabilities instead, but then the entropy measure would no longer capture the uncertainty in the model. This can be easily seen if we imagine a clear classification decision from the point of ¬label probabilities: we would expect $k-1$ high probability values and one low probability value to deduce the correct class label. However, the entropy measure does not reflect the large difference between k high probability values and $k-1$ high probability values.

Margin is calculated as the difference between the first and second highest class probability. The margin thus evaluates the competitiveness of the most likely class labels. However, it does not consider any information about the remaining class probabilities. In our setting, where we have ¬label probabilities, we can calculate the difference between the first and second lowest ¬label probability. The lower ¬label probabilities correspond to the more likely classes; therefore, a small margin reflects a high classifier uncertainty.

5.3 MaxDistance sampling

We have seen that the entropy measure cannot be used in our framework and that the margin measure does not contain information about the other ¬label probabilities. Our new MaxDistance Sampling includes all outputs and is easily applicable in our setting. Each SVDD in the committee describes a region in the data space. The outputs of each SVDD reflect whether the current example belongs to this ¬label (positive output) or not (negative output). In our framework, we expect that the outputs from the SVDDs are positive for a high number of ¬labels (ideally $k-1$), indicating that this example does not belong to the corresponding classes.

We try to select examples that reduce the error of the classifier. As we typically do not have a labeled ground truth in an active learning setting, we compute the expected error $\widehat{E}(\mathbf{x})$ of the classifier instead. We do this by computing the deviation from our expectation of the output. Let $(f^1(\mathbf{x}), \ldots, f^k(\mathbf{x}))$ denote the output vector of the SVDD committee. Our expectation is a vector of length k with $k-1$ positive entries and one negative entry at an arbitrary position: $(+, +, \ldots, +, -, +, \ldots, +)$. Our expected error is defined as:

$$\widehat{E}(\mathbf{x}) = \frac{1}{k-1} \sum_{c \in \{1,\ldots,k\}, c \neq \min(f^c(\mathbf{x}))} \delta(f^c(\mathbf{x})) \cdot |f^c(\mathbf{x})|$$

$$\delta(f^c(\mathbf{x})) = \begin{cases} 1, & \text{if } f^c(\mathbf{x}) \le 0 \\ 0 & \text{else} \end{cases} \qquad (6)$$

In this way, we focus on examples that are far away from all decision regions of the One-Class SVMs. In each active learning iteration, we select the example $\mathbf{x}$ with maximum error: $\max \widehat{E}(\mathbf{x})$.

There is also an intuitive way to describe this selection strategy: if an example is classified with a large negative value from the SVDD, but actually does belong to the

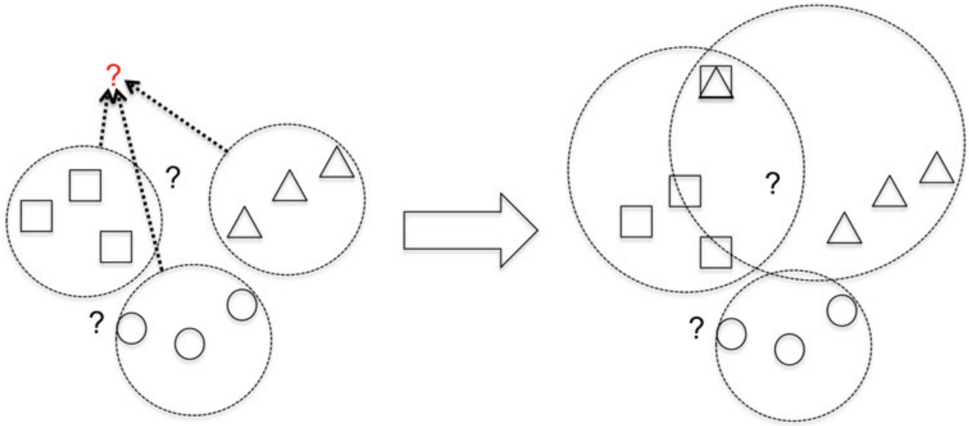
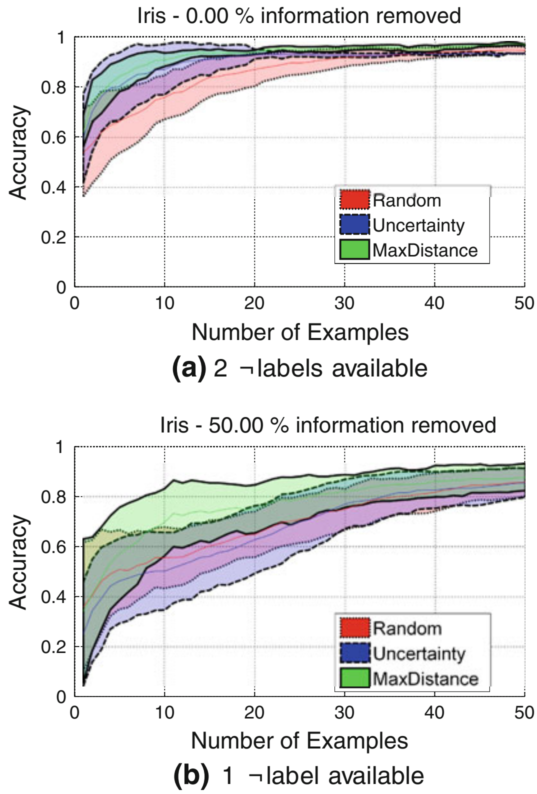


Fig. 4 Choosing an example with maximum distance from the current SVDD regions maximizes the effect on the committee in the next iteration

Table 1 Datasets, parameters and classification accuracy

Dataset	Samples	Classes	C	σ	Accuracy (%)
Iris	150	3	2	38.01	95.66
Multiclass	500	8	0.1	32	78.33
Libras	360	15	4	1.41	78.70
Landsat	4,435	7	2	2.23	83.23
Segmentation	2,310	7	0.2	13.45	87.95
RobotNavi	5,456	4	2	3.36	82.41

corresponding ¬label, the effect on the SVDD is very high as it needs to be retrained to include this example. With our approach, we try to maximize this effect on the committee of SVDDs. This idea is illustrated in Fig. 4. Each SVDD is described as a hypersphere. The output of the SVDD is positive if the example is within the radius of the hypersphere and negative if it is outside. When we need to choose an example to be labeled (question marks), we need to choose one that is far away from the current decision regions in order to maximize their growth after the labeling step.



(a) 2 ¬labels available

(b) 1 ¬label available

Fig. 5 Iris dataset (3 classes): comparison of random, uncertainty and MaxDistance sampling

6 Experiments

Before we go into the detailed descriptions of the experiments, we state our experimental methodology. The datasets have been chosen from the UCI machine learning repository [2]. We have filtered all datasets for classification tasks with multiple classes and numerical attributes given in matrix format, which leaves us with ten datasets from which we have chosen a subset from the most diverse classification domains. In each iteration, we either use the given split for training and testing or use 40 % for training and 60 % for testing (randomly chosen). All training instances are first assumed to be unlabeled. We select one example in each iteration (plotted on the x-axis) and plot the classification accuracy given the ground truth in the testing data on the y-axis. We compare three selection strategies: random sampling, uncertainty sampling (based on the margin) and our MaxDistance sampling.

As the original training sets contain the true class label, we have transformed them by inferring the corresponding $k-1$ ¬labels from the original class label. We are especially interested how the selection strategies perform with different numbers of ¬labels. Therefore, we have employed the following scheme: for a classification problem with k classes, we make individual experiments within the range of $k-1$

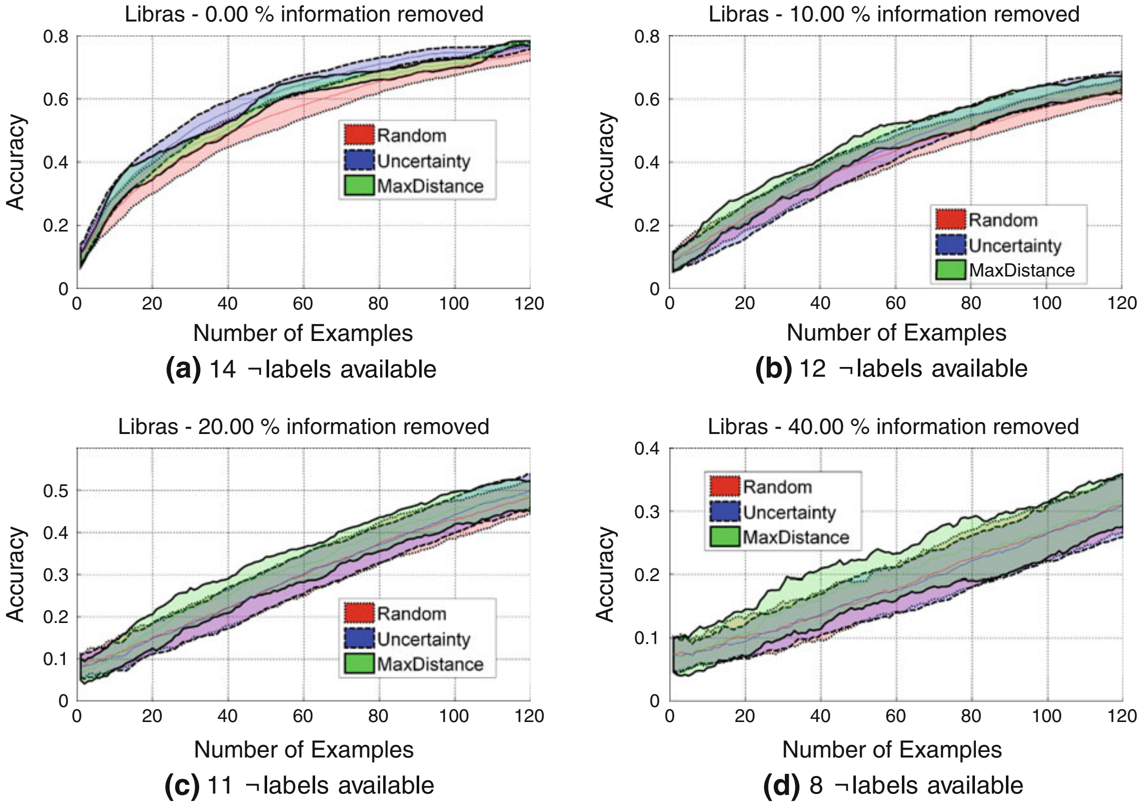


Fig. 6 Libras dataset (15 classes): comparison of random, uncertainty and MaxDistance sampling

to 1 ¬labels. In each iteration, we choose the ¬labels randomly as we have assumed that each ¬label is equally likely. We have repeated each experiment 100 times and plotted the mean accuracy along with the SD. We present selected individual experiments from the range of $k - 1$ to 1 ¬labels, to demonstrate certain effects. We have determined the parameters for the SVDD committee with fivefold cross validation on the training set. Table 1 gives an overview of the datasets, the C-parameter for the SVDD and the σ-parameter for the RBF kernel. We also list the classification accuracy based on the complete set of training examples with $k - 1$ ¬labels, which corresponds to a fully labeled dataset.

6.1 Iris

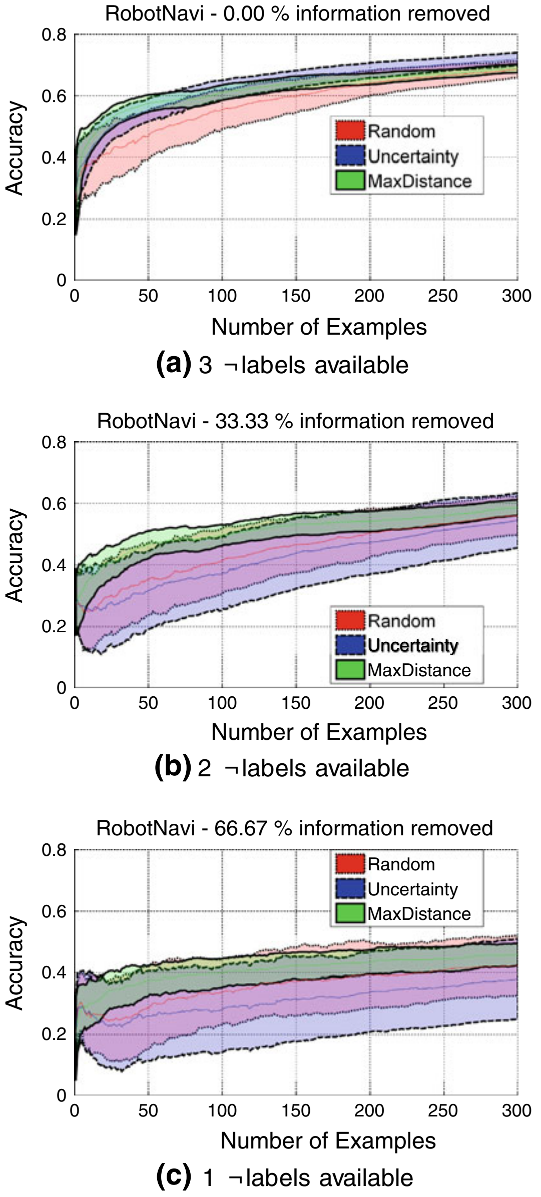
We start our experiments with the Iris dataset from the introductory example from Sect. 1. It contains three classes of 50 instances each, where each class refers to a type of iris plant. Figure 5 shows a comparison of the three sampling methods for a different amount of ¬labels. When the full information is available (2 ¬labels, Fig. 5a), the two active learning strategies Uncertainty and MaxDistance sampling perform better than random selection. This changes significantly when we remove information and only observe one ¬label (Fig. 5b) for each example: the performance of the uncertainty sampling strategy decreases to values that are mostly below the strategy of random sampling. The loss of information has an effect on the overall performance of all strategies. In relation to the other sampling strategies, our MaxDistance strategy performs significantly better.

6.2 Libras

The Libras dataset from the UCI machine learning repository [2] contains 15 classes of 24 instances each. Each class references to a hand movement type in Libras, a Brazilian sign language [11]. Figure 6 shows a comparison of the three sampling strategies for a different amount of ¬labels. We observe that the Uncertainty strategy performs best when all ¬labels are available, closely followed by the MaxDistance strategy. But as soon as there are less ¬labels available, its performance degrades and the MaxDistance strategy performs best.

6.3 RobotNavi

The Wall-Following Robot Navigation dataset from the UCI machine learning repository [2] contains measurements of 24 ultrasound sensors from a robot that navigates through the room following the wall in a clockwise direction. This dataset has a very small number of classes, which correspond



(a) 3 ¬labels available

(b) 2 ¬labels available

(c) 1 ¬labels available

Fig. 7 Robot navigation dataset (4 classes): comparison of random, uncertainty and MaxDistance sampling

to four different directions for the robot. Figure 7 demonstrates the performance of the three different selection strategies.

The MaxDistance strategy performs best in the first few iterations, but the performance degrades in later iterations. This might be due to the fact that the MaxDistance strategy tends to explore the dataspace more, whereas uncertainty sampling focuses on class boundaries that are difficult to separate. For some datasets, this strategy might be more beneficial. We note that when we observe less ¬labels in the classification process, MaxDistance performs best and uncertainty sampling again performs worse than random sampling.

6.4 Landsat

The Landsat data from the UCI machine learning repository [2] consists of 4,435 training instances which describe satellite image data by their multispectral values. There are seven different classes in this dataset. Figure 8 compares the performance of the selection schemes on different levels of supervision. Again, the uncertainty strategy performs better than the other strategies when all ¬labels are available. When information is removed from the ¬labels, the MaxDistance strategy gains performance in comparison to the other strategies.

6.5 Segment

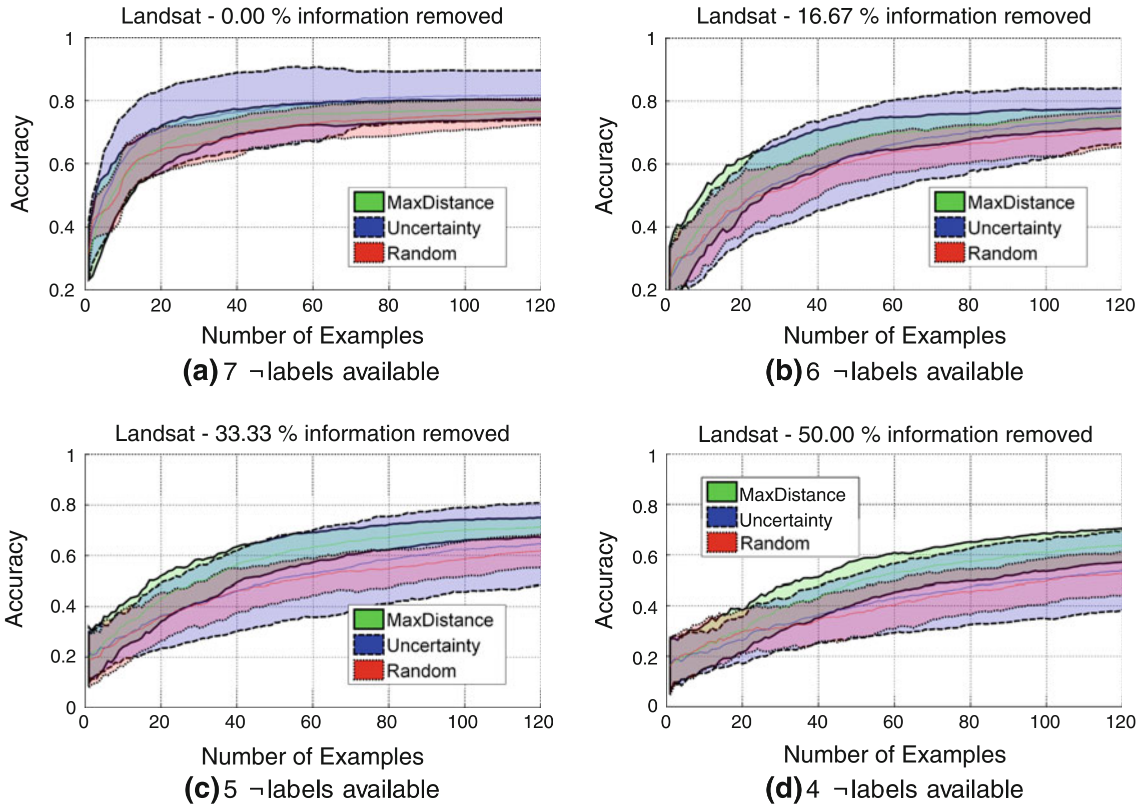
The segment data from the UCI machine learning repository [2] consists of 19 numerical features of different outdoor images. The images were hand-segmented to create a classification for every pixel. There are eight classes in this dataset. Figure 9 compares the performance of the selection schemes on different levels of supervision. Again, we can observe that the MaxDistance strategy performs well with full supervision and better than the other methods if there is less supervision.

6.6 Multiclass

The multiclass dataset is an artificially generated two-dimensional dataset of eight Gaussian distributed classes from different shapes (i.e. banana-shapes, spheres). We have used the implementation in the PRTools software [13]. All classes have equal prior probabilities and we have used a training size of 200 examples and a test set size of 300 examples. Figure 10 shows a comparison of the different selection strategies for a different number of ¬labels. Overall, we can observe that the MaxDistance strategy performs best, especially in the first iterations. Its SD is much smaller compared to the other methods, showing that it is more stable. Uncertainty sampling has the highest SD; as soon as we do not have the full set of ¬labels available, its performance is inferior to random sampling. We believe that this is due to the fact that uncertainty sampling only considers two ¬classes at a time, whereas the MaxDistance strategy tries to maximize the impact on all members of the classifier committee. The MaxDistance strategy seems to be well suited for problems where we only have a small set of ¬labels available, as we would expect it in a real-world setting.

7 Conclusions

In this work, we have introduced a new form of feedback in classification that is easier to obtain than a class label. Not only is the assignment of ¬labels to examples easier,



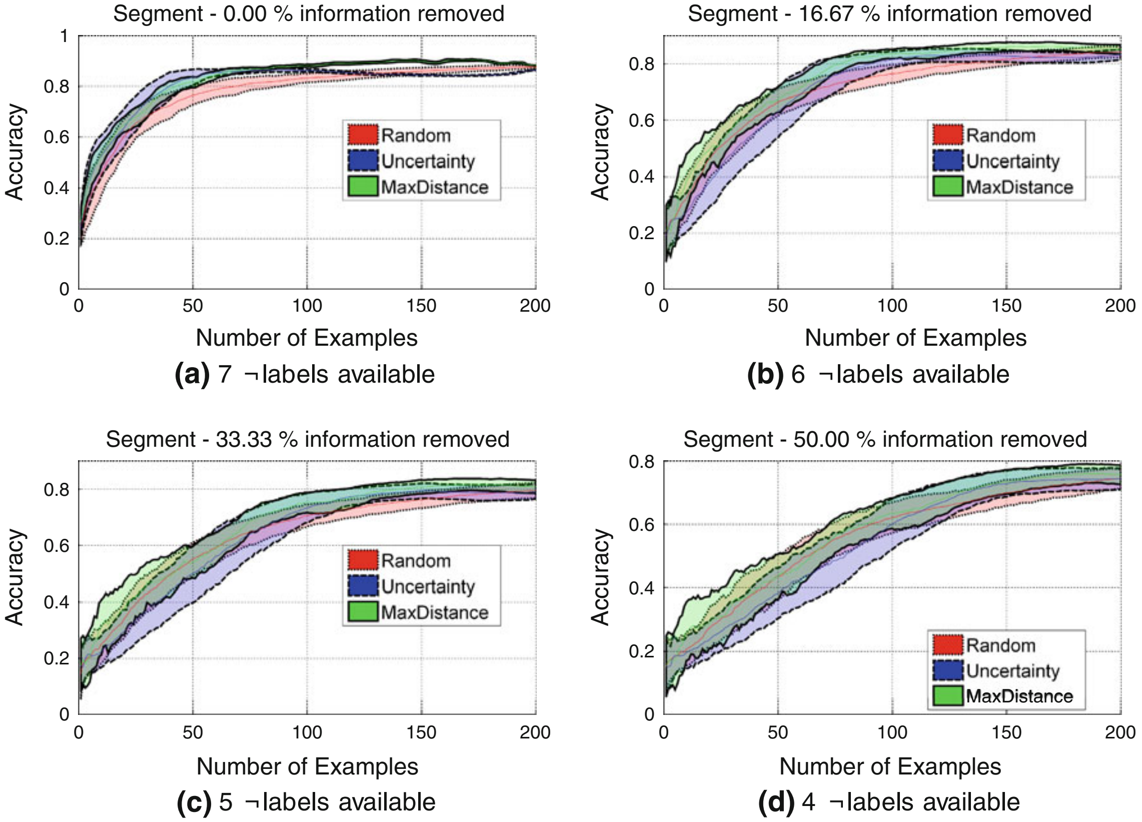
(a) 7 ¬labels available

(b) 6 ¬labels available

(c) 5 ¬labels available

(d) 4 ¬labels available

Fig. 8 Landsat dataset (7 classes): comparison of random, uncertainty and MaxDistance sampling



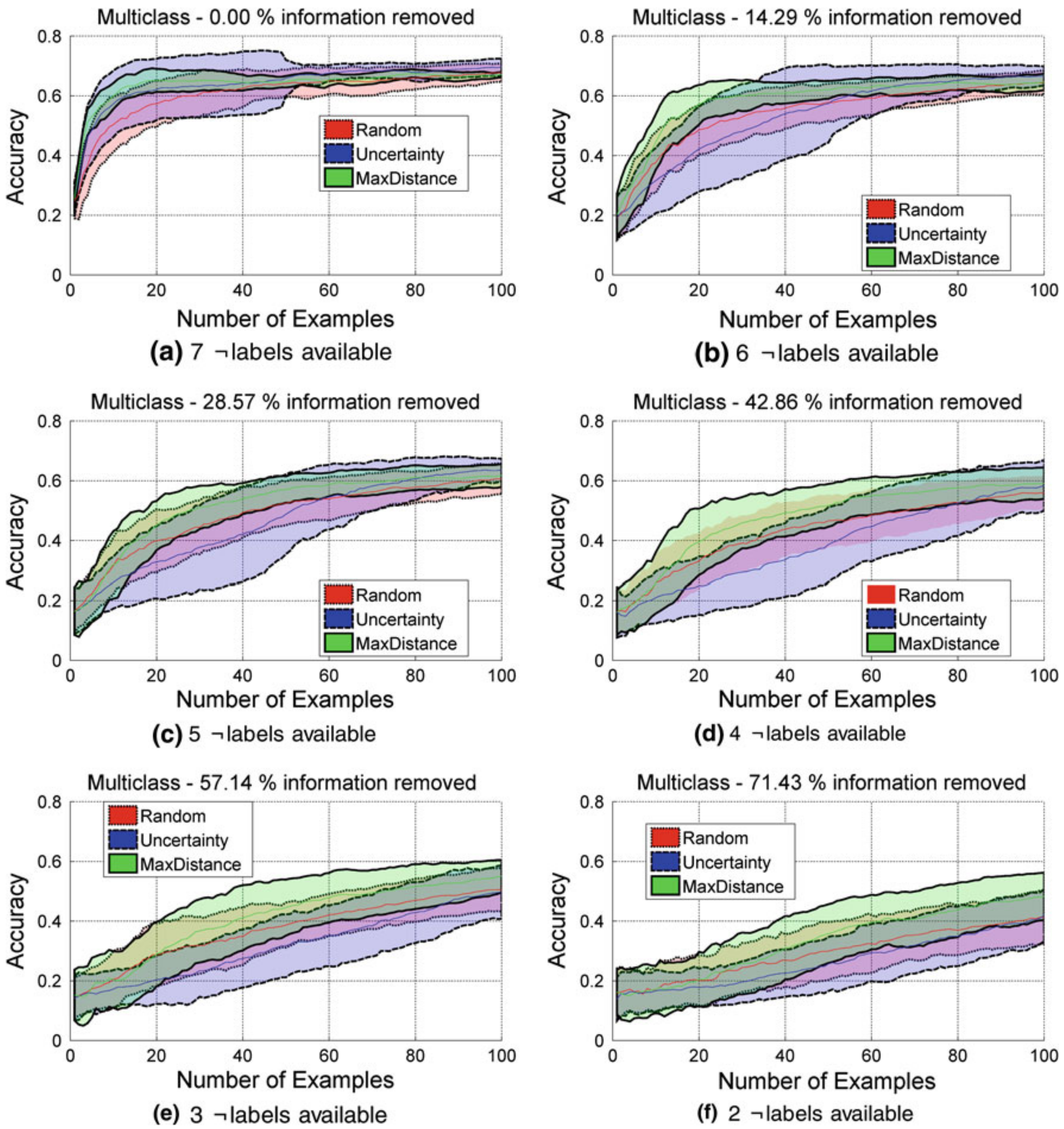
(a) 7 ¬labels available

(b) 6 ¬labels available

(c) 5 ¬labels available

(d) 4 ¬labels available

Fig. 9 Segmentation dataset (8 classes): comparison of random, uncertainty and MaxDistance sampling



(a) 7 ¬labels available

(b) 6 ¬labels available

(c) 5 ¬labels available

(d) 4 ¬labels available

(e) 3 ¬labels available

(f) 2 ¬labels available

Fig. 10 Multiclass dataset (8 classes): comparison of random, uncertainty and MaxDistance sampling

but it is also faster for the human expert and allows to gain information for almost every example.

We have introduced a new framework for the classification with ¬labels. A committee of SVDDs is built—each one defining a nonlinear region in the data space for the corresponding ¬label. In this framework, a large overlap of these regions is sought in order to deduce the correct class label. We have developed a selection strategy that tries to maximize the impact on (and with it the overlap of) the committee of SVDDs.

We have seen that classical strategies in active learning can fail when less information is available. In almost all cases, random sampling performed better than uncertainty sampling. Experiments have shown that our MaxDistance strategy works best in this case. Whenever less than $k - 1$ ¬labels are given from the human expert—a setting that we would consider as typical in a real-world environment—MaxDistance sampling outperforms the other strategies.

In this work, we have assumed that each ¬label is equally likely. In the future, it would be interesting to explore whether certain preferences for ¬labels can be deduced from the human expert behavior and how these preferences could be included in the selection process.

We hope that this work does inspire future work in the community on different forms of feedback in active learning and their impact on the selection strategy.

References

1. Ashwin, T., Jain, N., Ghosal, S.: Improving image retrieval performance with negative relevance feedback. IEEE Int. Conf. Acoust. Speech Signal Process. **3**, 1637–1640 (2001)
2. Asuncion, A., Newman, D.: UCI machine learning repository. http://mlearn.ics.uci.edu/MLRepository.html (2010)
3. Balcan, M.F., Beygelzimer, A., Langford, J.: Agnostic active learning. In: ICML '06: Proceedings of the 23rd International Conference on Machine Learning, pp. 65–72. ACM, New York (2006). doi:10.1145/1143844.1143853
4. Baram, Y., El-Yaniv, R., Luz, K.: Online choice of active learning algorithms. J. Mach. Learn. Res. **5**, 255–291 (2004)
5. Campbell, C., Cristianini, N., Smola, A.J.: Query learning with large margin classifiers. In: ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 111–118. Morgan Kaufmann, San Francisco (2000)
6. Cebron, N., Berthold, M.R.: Active learning for object classification: from exploration to exploitation. Data Min. Knowl. Discov. **18**(2), 283–299 (2009)
7. Cohn, D.A., Atlas, L., Ladner, R.E.: Improving generalization with active learning. Mach. Learn. **15**(2), 201–221 (1994)
8. Cohn, D.A., Ghahramani, Z., Jordan, M.I.: Active learning with statistical models. In: Tesauro, G., Touretzky, D.S., Leen, T.K. (eds.) NIPS, pp. 705–712. MIT Press, Cambridge (1994)
9. Cour, T., Sapp, B., Taskar, B.: Learning from partial labels. J. Mach. Learn. Res. **12**, 1501–1536 (2011)
10. Dasgupta, S., Kalai, A.T., Monteleoni, C.: Analysis of perceptron-based active learning. J. Mach. Learn. Res. **10**, 281–299 (2009)
11. Dias, D.B., Madeo, R.C.B., Rocha, T., Bíscaro, H.H., Peres, S.M.: Hand movement recognition for Brazilian sign language: a study using distance-based neural networks. In: Proceedings of the 2009 International Joint Conference on Neural Networks, IJCNN'09, pp. 2355–2362. IEEE Press, Piscataway (2009). http://portal.acm.org/citation.cfm?id=1704555.1704610
12. Freund, Y., Seung, H.S., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. Mach. Learn. **28**(2–3), 133–168 (1997)
13. van der Heijden, F., Duin, R., de Ridder, D., Tax, D.: Classification, Parameter Estimation and State Estimation: An Engineering Approach Using Matlab. Wiley, New York (2004)
14. Hou, C., Nie, F., Wang, F., Zhang, C., Wu, Y.: Semisupervised learning using negative labels. IEEE Trans. Neural Netw. **22**(3), 420–432 (2011)
15. Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Breaking the interactive bottleneck in multi-class classification with active selection and binary feedback. In: CVPR, pp. 2995–3002. IEEE, New York (2010)
16. Lindenbaum, M., Markovitch, S., Rusakov, D.: Selective sampling for nearest neighbor classifiers. Mach. Learn. **54**(2), 125–152 (2004)
17. MacKay, D.J.C.: Information-based objective functions for active data selection. Neural Comput. **4**(4), 590–604 (1992)
18. Mueller, H., Mueller, W., Squire, D.M., Marchand-Maillet, S., Pun, T.: Strategies for positive and negative relevance feedback in image retrieval. In: Proceedings of the International Conference on Pattern Recognition, vol. 1, pp. 1043–1046. IEEE Computer Society, Washington, DC (2000). http://portal.acm.org/citation.cfm?id=876866.877254
19. Nguyen, N., Caruana, R.: Classification with partial labels. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, pp. 551–559. ACM, New York (2008)
20. Osugi, T., Kun, D., Scott, S.: Balancing exploration and exploitation: a new algorithm for active machine learning. In: ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining, pp. 330–337. IEEE Computer Society, Washington, DC (2005). doi:10.1109/ICDM.2005.33
21. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: Brodley, C.E., Danyluk, A.P. (eds.) International Conference on Machine Learning (ICML), pp. 441–448. Morgan Kaufmann, Menlo Park (2001)
22. Schohn, G., Cohn, D.: Less is more: active learning with support vector machines. In: Langley, P. (ed.) ICML, pp. 839–846. Morgan Kaufmann, Menlo Park (2000)
23. Tax, D.M.J., Duin, R.P.W.: Support vector data description. Mach. Learn. **54**(1), 45–66 (2004)
24. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. J. Mach. Learn. Res. **2**, 45–66 (2001)
25. Tsoumakas, G., Katakis, I.: Multi label classification: an overview. Int. J. Data Wareh. Min. **3**(3), 1–13 (2007)