Improving Active Learning by Avoiding Ambiguous Samples

Christian Limberg[1,2], Heiko Wersing[2], and Helge Ritter[1]

[1] Bielefeld University, CoR-Lab, Universitätsstraße 25, 33615 Bielefeld - Germany
{climberg,helge}@techfak.uni-bielefeld.de
[2] HONDA Research Institute Europe GmbH, Carl-Legien-Straße 30, 63073 Offenbach - Germany heiko.wersing@honda-ri.de

Abstract. If label information in a classification task is expensive, it can be beneficial to use active learning to get the most informative samples to label by a human. However, there can be samples which are meaningless to the human or recorded wrongly. If these samples are near the classifier's decision boundary, they are queried repeatedly for labeling. This is inefficient for training because the human can not label these samples correctly and this may lower human acceptance. We introduce an approach to compensate the problem of ambiguous samples by excluding clustered samples from labeling. We compare this approach to other state-of-the-art methods. We further show that we can improve the accuracy in active learning and reduce the number of ambiguous samples queried while training.

Keywords: active learning, ambiguous samples, certainty, rejection, clustering

1 Motivation

User-adaptable learning systems, who are post-trained by the user have the advantage, that they can adjust to new circumstances or improve towards a user-specific environment. In a classification system the samples can be trained incrementally and labeled by the user. Active learning [10] is an efficient training technique, where the samples which are predicted to deliver the highest improvement for the classifier are chosen for labeling by a human.

Whenever the user is involved, the system has to make sure that interaction and training is efficient. A user often feels bored with labeling tasks, therefore the learning system should limit the number of actions and they should be solvable for the human to not annoy him and instead make him feel comfortable and meaningful in his role as interaction partner. To know the time when the learning system needs advice, it is necessary to predict the competence of the learning system, which we demonstrated in our recent contribution [6] with respect to a classifier's accuracy in pool-based incremental active learning. However, on the other side the human teacher can also have limited competence to fulfill his task in an oracle role.

In most active learning approaches the oracle is expected to have perfect domain knowledge [11]. But in many real world applications a perfect oracle is not realistic because there can be samples resulting from noisy recordings like a dirty camera or bad light conditions. Also a specific oracle might not know the labels for specific samples because it can not identify them.

Our goal in this contribution is, that the learning system should adapt to the human weaknesses and adapt its strategy of interacting as a good cooperation partner. Related to active learning that means, rather than forcing the human to give uncertain answers, we want to give him the opportunity to reject the samples he is uncertain about.

There are diverse approaches in the literature for handling uncertainty in labeling. Much research was done on active learning with noisy labels or with labels from multiple oracles [15]. However in our task setting the robot is intended to have access to only one oracle. Käding et al. [5] proposed an approach for their Expected Model Output Change (EMOC) model that adds uncertain samples in one error class. However, their method only works with EMOC and is directly integrated into the classifier. A similar approach was done by Fang et al. [3]. They train a classifier that should distinguish certain and uncertain objects. However, in their evaluation they have clustered the data in three clusters and define two of them as ambiguous, which is too simplistic and does not model a real world task. The problem with classifier-based solutions for finding and rejecting ambiguous samples is that they, according to our experiments, can not generalize well in highly complex scenarios like the one we are facing. In our application scenario, a service robot acts in a garden environment [7], mows the lawn and records the garden and occurring objects by a camera. However, because occurring objects are diverse, there is no clear concept between recognizable and ambiguous samples in the feature space, making it hard to train i.e. a secondary classifier to separate them, as is shown in the experiment section.

We show that a more local method is better able to adapt to this distributed ambiguous samples and therefore we introduce Density-Based Querying Exclusion (DBQE), a lightweight clustering-based approach which finds ambiguous clusters and excludes them from querying in active learning. Our approach does not inhibit exploration of unknown classes, and can be stacked up to any existing active learning model and every querying technique. We evaluate it using a challenging outdoor data set (Fig. 1).

2 Active Learning

In pool-based active learning there is a labeled set $\mathcal{L}$ and an unlabeled set $\mathcal{U}$. The active training of a classifier C starts with an empty or small $\mathcal{L}$. The learner C can choose which samples from $\mathcal{U}$ should be labeled by a so-called oracle (which is often a human) and added to $\mathcal{L}$. This is called querying and there are a variety of approaches to find the best samples to query [11]. An often used querying technique is uncertainty sampling [1] which queries the samples with the least certainty for labeling. Other strategies select samples based on

the expected model output change [5], or they consider a committee of different classifiers [12] for choosing the samples to be queried. C is then trained in an incremental fashion or again from scratch on $\mathcal{L}$.

Fig. 1. Images from the outdoor object recognition benchmark [7,8]: The upper row images are labeled as recognizable and the bottom row as ambiguous. Objects like the basketball or the leaves are recognizable from every angle. The car is recorded in its canonical view, opposed to the blue duck which is ambiguous from this perspective. There are also views of different objects which are hardly distinguishable, like an apple (bottom center) and a tomato (bottom right).

3 Density-Based Querying Exclusion

We introduce Density-Based Querying Exclusion (DBQE) which clusters ambiguous samples and prevents them from querying by excluding them from $\mathcal{U}$. Our assumption is that ambiguous samples are located in clusters which can occur in a variety of places in the feature space. Density-based clustering approaches showed to be versatile and deliver good performance while at the same time are robust with handling outliers [2]. Another advantage is that the number of clusters does not have to be known in advance. This is important in particular because in our case we want to find only one cluster at a time, while there can be any number of clusters in the data set.

The training procedure of an active learning classifier using DBQE is illustrated in Algorithm 1.

Algorithm 1 Active learning with Density-Based Querying Exclusion (DBQE)

Require: $maxPts$ ▷ do clustering on $maxPts$ points nearby x_e
Require: $minPts$ ▷ minimum number of neighbors to be a core sample
Require: ϵ ▷ distance range describing a sample's neighborhood
$\mathcal{U} \leftarrow load\_data()$ ▷ unlabeled data
$\mathcal{L} \leftarrow \{\}$ ▷ labeled Set is empty
$C \leftarrow initialize\_classifier()$ ▷ initialize active classifier
while not $C.is\_trained()$ **do**
  $s \leftarrow C.query\_next\_sample(\mathcal{U})$ ▷ querying using uncertainty sampling
  $l \leftarrow ask\_for\_label(s)$ ▷ ask oracle for supervision
  if $l.is\_ambiguous()$ **then** ▷ oracle labeled s as ambiguous
    $c \leftarrow DBQE(s, minPts, maxPts, \epsilon)$ ▷ DBQE clustering is applied
    $\mathcal{U} \leftarrow \mathcal{U} \setminus c$ ▷ found cluster c is excluded from $\mathcal{U}$
  else ▷ s is not ambiguous and oracle labeled it
    $C.train(s, l)$ ▷ classifier C is trained with new sample s and label l
  end if
end while

function DBQE($x_e, minPts, maxPts, \epsilon$)
  $v \leftarrow \{\}$ ▷ visited samples
  $c \leftarrow \{x_e\}$ ▷ samples considered to be in cluster
  $t \leftarrow \{x_e\}$ ▷ samples to be processed
  $\mathcal{R} \leftarrow get\_samples\_nearby(\mathcal{U}, x_e, maxPts)$ ▷ get $maxPts$ nearest samples to x_e
  for $a \in t$ **do**
    if not $a \in v$ **then** ▷ if a was not visited before
      $v \leftarrow v \cup a$ ▷ mark a as visited
      $n \leftarrow region\_query(a, \epsilon)$ ▷ find neighborhood points
      if $n.size() > minPts$ **then** ▷ if a is a core sample
        $c \leftarrow c \cup a$ ▷ add a to cluster set c
        $t \leftarrow t \cup n$ ▷ add n to t
      end if
    end if
    $t \leftarrow t \setminus a$ ▷ remove a from queue t
  end for
  return c ▷ return ambiguous cluster c
end function

function $region\_query(s, \epsilon)$ ▷ returns samples from $\mathcal{R}$ within range ϵ to s
  $n \leftarrow \{\}$
  for $i \in \mathcal{R}$ **do**
    if $|i - s| < \epsilon$ **then** ▷ sample i is within ϵ range
      $n \leftarrow n \cup i$ ▷ i is added to set n
    end if
  end for
  return n ▷ samples in neighborhood are returned
end function

The active learning is applied as usual: First the query strategy selects a sample and the oracle is asked for a label. If it can provide it, the classifier is trained, otherwise our DBQE approach is applied which does a region growing to find the cluster containing the queried ambiguous sample, which we call x_e. In the clustering function we select a subset of samples $\mathcal{R} \subseteq \mathcal{U}$ which are the nearest samples to x_e for speed improvements and to limit the maximum number of excluded samples, denoting $maxPts$ as the number of points in $\mathcal{R}$. The region growing is applied similar to DBSCAN [2], also illustrated in Fig. 2. DBSCAN iteratively applies this region growing until the whole data set is clustered. There are two parameters involved: ϵ is a distance range describing an arbitrary sample's neighborhood points. The other parameter to choose is $minPts$ which is the minimum number of samples in a sample's neighborhood for the sample to be a so-called core sample, otherwise it is an outlier. The main idea is to expand a cluster c around the ambiguous sample x_e. The cluster samples in c are excluded from $\mathcal{U}$.

If there is no cluster containing x_e (so x_e itself is an outlier) DBQE is only excluding x_e from $\mathcal{U}$.

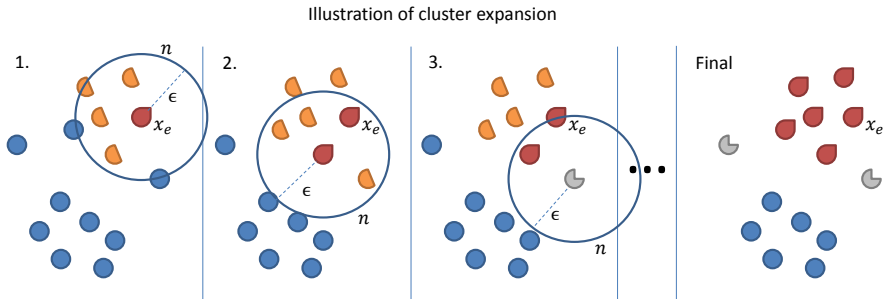

Fig. 2. Illustration of DBQE: the points represent samples from the unlabeled subset $\mathcal{R} \subseteq \mathcal{U}$ with the number of samples $maxPts = 14$. Blue points (circles) are samples not visited, visited points in v are displayed orange (half circles) and points determined as part of the ambiguous cluster c are in red (peaked circles), outliers in gray (pacman shape). The progress of the region growing is displayed with the minimum neighborhood size $minPts = 3$. The oracle defines x_e as ambiguous and in the first step x_e is determined as a core sample. The cluster is expanded, finding the second core sample in step 2. In step 3, an outlier is found, which is not included into the cluster. The final clustering result is displayed on the right.

4 Evaluation

We evaluated our method together with some baseline methods on our outdoor data set [7] because it provides a real application benchmark of high difficulty [6,7,8]. The data set is an image data set consisting of 50 object classes. The objects are laying on the lawn and were recorded by a mobile robot in a way

that the robot approaches the object and makes ten consecutive pictures each approach. In total each object has ten approaches with ten images each, summing up a total of 5000 images. Some objects can be hard to distinguish due to unfavorable viewing angle. Also there are some objects that look rather similar like an apple, onion, tomato, orange and ball or e.g. several rubber ducks. A feature representation of each image is extracted with the VGG16 deep convolutional net [13] trained on images from the imagenet competition. We removed the last softmax layer and using the outputs of the penultimate layer as a 4096 dimensional feature vector. There can be approaches or partial approaches of an object, from which the object images can be ambiguous for a human. We annotated this ambiguity property for our data set (compare to Fig. 1). In total we annotated 24% of the images as ambiguous, a selection of recognizable and ambiguous images can be seen in Fig. 1. For evaluation a 50/50 train-test split was done. The data was split by approaches, so that the images of a single approach are either completely in the train or in the test set. We repeated the experiment 15 times to average our results. As a classifier we chose Generalized Learning Vector Quantization (GLVQ). GLVQ has proved to be an accurate classifier in incremental learning [8] and is also suitable for active learning with uncertainty sampling [6].

DBQE needs the parameters $minPts$ and ϵ to be set to a suitable value. To have a better idea how the data is clustered, a look at unsupervised statistics related to the distances to neighboring samples can help. We achieved good results with many parameter combinations but we also applied a grid search where we defined ranges of $minPts$ and ϵ values and tested all combinations of those. There we found out $\epsilon = 35$, $minPts = 3$ and $maxPts = 20$ give best results for our evaluation on the outdoor data set. For training and evaluating an active learning classifier we developed the framework ALeFra[3] in context of this paper. By using it any offline and incremental classifier can be converted to an active classifier. There are also basic querying techniques implemented and the user can visualize the progress of the training with a few lines of code. There is a visualization of the feature space which uses a dimensional reduction like t-SNE [9] or MDS [14] and if the data consists of images, they are visualized in a collage which is created after each batch while training.

We investigate three approaches and compare them to simple baselines:

- **Classifier:** The problem can be represented as a binary classification task, predicting whether samples are recognizable or not [3]. The classifier is trained with all yet queried recognizable and ambiguous samples. We evaluated the classifiers GLVQ, kNN, logistic regression and SVM, where the kNN outperformed the others. This may occur because a local model like kNN can better adapt to the ambiguous samples, who may be diverse in feature space. Also we have observed, that if using classifier's confidence information of predicted samples can improve performance and exploring new classes in $\mathcal{U}$. Therefore we make use of a certainty value of the kNN-classifier, which uses distance information of the winning and loosing classes defined in [6].

[3] `https://github.com/limchr/ALeFra`

Only samples who are classified as ambiguous with a certainty value greater than a predefined threshold are avoided in querying. We tuned this threshold to the best performance for our evaluation on the outdoor data set.

- **Rejection:** The problem can be represented as a rejection task, where some samples are rejected from querying. Therefore we implemented a local rejection approach [4] for the GLVQ-classifier. Here every prototype has a rejection threshold which is set to zero at beginning. If an ambiguous sample is queried, the winning prototype's threshold is adjusted to $d * \alpha$, where d is the certainty of the ambiguous sample and α is a parameter that can be tuned. Only those samples are considered for querying, for which the distance d to their winning prototype is higher than the threshold of that particular prototype.
- **Clustering:** The problem can be represented as a clustering task. DBQE is using density-based clustering to represent ambiguous samples. We also tried to apply silhouette analysis, but density-based clustering results in higher accuracy in finding the ambiguous clusters and additionally it is very fast to expand a cluster and it can also detect outliers.

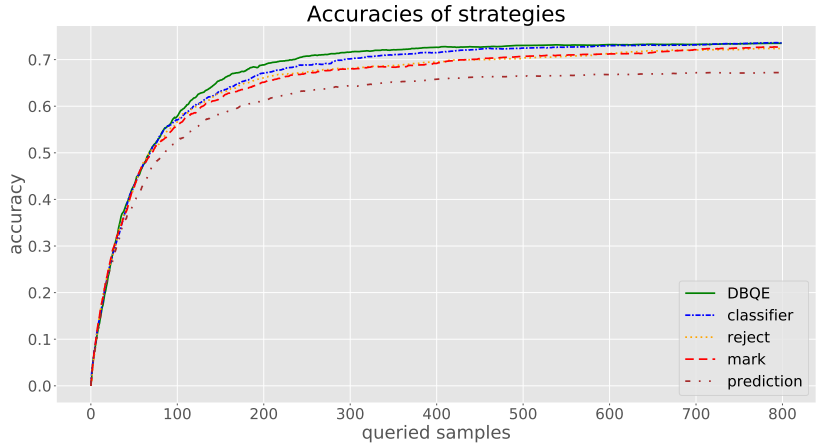

Fig. 3. Evaluation on the outdoor data set: test-accuracies (y-axis) of all approaches vs. number of queried samples (x-axis).

We also implemented the following two baseline strategies for comparison:

- **Mark:** If an ambiguous sample is queried, it is marked as ambiguous and is not considered in future queryings. This baseline strategy can be seen as a naive approach for handling ambiguous samples.
- **Prediction:** If an ambiguous sample is queried, the classifier predicts its label and uses this for training. With this baseline we want to determine if

the classifier itself is able to classify the samples that the human rejected as ambiguous.

Fig. 3 shows the test-accuracies of the strategies for active training. DBQE and *classifier* are the two strategies with the highest accuracy where DBQE is better in the middle stage of the training. *Reject* is slightly better than *mark*, where at the end of training, both are converging to DBQE and *classifier*. *Prediction* is significantly worse than the other approaches, indicating that the classifier is not accurate at predicting those labels that the human can not provide.

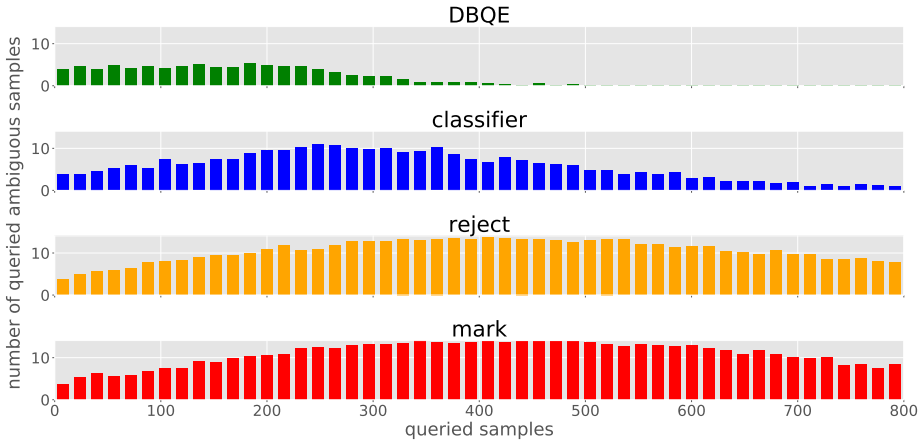

Fig. 4. Number of queried ambiguous samples during training. Each bin of the histograms represents the number of ambiguous samples queried, pooled in bins of 16 queryings giving a total of 50 bins. The number of ambiguous samples is displayed on the y-axis and the number of queries on the x-axis. Please note that the baseline strategy *prediction* is not represented here because it is using ambiguous samples for training.

DBQE is slightly better than *classifier* in terms of accuracy while training. However, another important objective was to minimize human frustration and to make him feel comfortable in his role. Therefore we visualized the number of ambiguous queried samples while training. In Fig. 4 it can be seen that significantly fewer samples are queried using DBQE. After 400 trained samples, ambiguous samples are queried only occasionally. The querying of ambiguous samples using *classifier* only drops slowly and especially in the earlier stage of training is significantly higher than DBQE. *Mark* is querying the most ambiguous samples compared to DBQE and *classifier*. To better visualize the total number of ambiguous queried samples, we plotted the cumulative sum of ambiguous queried samples in Fig. 5. DBQE is capable of querying approximately three times less ambiguous samples than *classifier* and five times less than *reject* and *mark*.

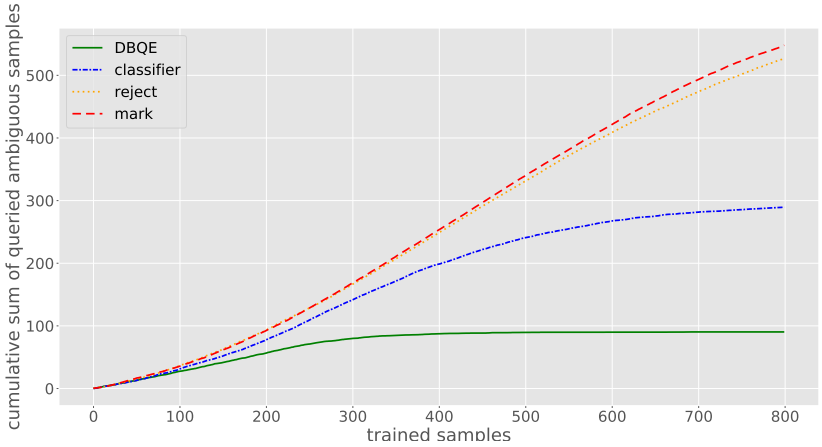

Fig. 5. Cumulative sum of queried ambiguous samples during training.

5 Conclusion

We showed that it is possible to efficiently exclude ambiguous samples from active learning. In our challenging outdoor object recognition setting, where ambiguous samples were distributed over the whole feature space, DBQE is able to improve the accuracy in active learning and further reduces the amount of meaningless queries significantly. We implemented and evaluated a variety of other approaches in depth and compared them to DBQE in a realistic setting.

We think that DBQE can be used to model human capabilities and significantly improve robot acceptance as a cooperation partner. To prove this as a next step we want to integrate DBQE in a robotic application and investigate a larger number of benchmarks.

References

1. Constantinopoulos, C., Likas, A.: Active learning with the probabilistic RBF classifier. In: International Conference on Artificial Neural Networks (ICANN). pp. 357–366 (2006)
2. Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). pp. 226–231 (1996)
3. Fang, M., Zhu, X.: I don't know the label: Active learning with blind knowledge. In: Proceedings of the 21st International Conference on Pattern Recognition (ICPR). pp. 2238–2241 (2012)
4. Fischer, L., Hammer, B., Wersing, H.: Optimal local rejection for classifiers. Neurocomputing **214**, 445–457 (2016)

5. Käding, C., Freytag, A., Rodner, E., Bodesheim, P., Denzler, J.: Active learning and discovery of object categories in the presence of unnameable instances. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4343–4352 (2015)
6. Limberg, C., Wersing, H., Ritter, H.: Efficient accuracy estimation for instance-based incremental active learning. In: European Symposium on Artificial Neural Networks (ESANN). pp. 171–176 (2018)
7. Losing, V., Hammer, B., Wersing, H.: Interactive online learning for obstacle classification on a mobile robot. In: International Joint Conference on Neural Networks (IJCNN). pp. 1–8 (2015)
8. Losing, V., Hammer, B., Wersing, H.: Incremental on-line learning: A review and comparison of state of the art algorithms. Neurocomputing **275**, 1261–1274 (2018)
9. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**, 2579–2605 (2008)
10. Ramirez-Loaiza, M.E., Sharma, M., Kumar, G., Bilgic, M.: Active learning: an empirical study of common baselines. Data Mining and Knowledge Discovery **31**(2), 287–313 (2017)
11. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1070–1079 (2008)
12. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: Conference on Computational Learning Theory (COLT). pp. 287–294 (1992)
13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
14. Strickert, M., Teichmann, S., Sreenivasulu, N., Seiffert, U.: High-throughput multi-dimensional scaling (hit-mds) for cdna-array expression data. In: International Conference on Artificial Neural Networks (ICANN). pp. 625–633 (2005)
15. Zhang, J., Wu, X., Sheng, V.S.: Learning from crowdsourced labeled data: a survey. Artif. Intell. Rev. **46**(4), 543–576 (2016)