Automated Active Learning with a Robot

Kristina Scharei, Marek Herde, Maarten Bieshaar, Adrian Calma, Daniel Kottke and Bernhard Sick

Abstract In the field of automated processes in industry, a major goal is for robots to solve new tasks without costly adaptions. Therefore, it is of advantage if the robot can perform new tasks independently while the learning process is intuitively understandable for humans. In this article, we present a highly automated and intuitive active learning algorithm for robots. It learns new classification tasks by asking questions to a human teacher and automatically decides when to stop the learning process by self-assessing its confidence. This so-called stopping criterion is required to guarantee a fully automated procedure. Our approach is highly interactive as we use speech for communication and a graphical visualization tool. The latter provides information about the learning progress and the stopping criterion, which helps the human teacher in understanding the training process better. The applicability of our approach is shown and evaluated on a real Baxter robot.

Kristina Scharei · Marek Herde · Maarten Bieshaar · Adrian Calma · Daniel Kottke · Bernhard Sick
Intelligent Embedded Systems, University of Kassel
Wilhelmshöher Allee 71–73, 34121 Kassel, Germany
✉ kristina.scharei@uni-kassel.de
✉ marek.herde@uni-kassel.de
✉ mbieshaar@uni-kassel.de
✉ adrian.calma@vencortex.com (research done while working at the University of Kassel)
✉ daniel.kottke@uni-kassel.de
✉ bsick@uni-kassel.de

Archives of Data Science, Series A
(Online First)
KIT Scientific Publishing
Vol. 5, No. 1, 2018

DOI: 10.5445/KSP/1000087327/16
ISSN 2363-9881

1 Introduction

Ever since the existence of robot manipulators, robotics has been a key technology for industrial automation and mass production. These robots are mostly pre-programmed to perform a task repeatedly, over and over again. To cope with the challenges of a globalized world, industrial companies must be able to quickly adapt to new circumstances, shorter product life cycles, or the regionalization and customization of products. Here, machine learning is considered as an enabling technology for a paradigm shift from programming the robot to teaching it by the operator on-site. However, teaching a robot is mostly a time-consuming and therefore costly procedure. Active learning (AL; Settles, 2010) is a machine learning paradigm in which the learning algorithm actively selects data from an unlabeled pool to be labeled by an expert.

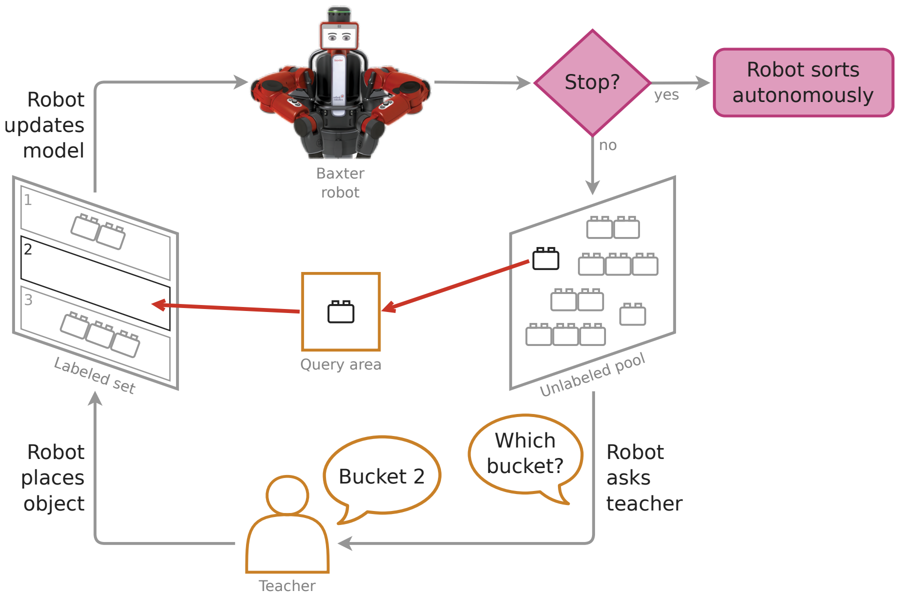

Figure 1: Our automated active learning application running on a Baxter robot to sort objects into buckets. Key elements addressed in this article are the stopping criterion (upper right), the sorting procedure (center) and the intuitive interaction with the teacher (orange).

Figure 1 shows our transductive AL application where the robot aims to correctly sort a selection of a-priori known items. In this AL example, the objects in the unlabeled pool must be sorted into three buckets according to an arbitrary decision rule defined by the human teacher. The robot selects an object according to a selection strategy, which minimizes the total number of queries selecting only objects with the highest information content. Hence, the human teacher is being actively queried by the robot to provide class labels such that the robot can verify or specify its current classification hypothesis. The robot keeps on querying samples until it is certain about the decision rule and starts sorting autonomously without the teacher's help. Especially, the robot's ability to decide when to stop the AL process adds an extra degree of autonomy, making it an automated AL procedure.

Our vision is to create a fully automated and highly interactive as well as intuitive AL process for sorting robots. In a previous article (Herde et al, 2018), we presented an efficient training procedure for sorting robots and showed that probabilistic AL can be applied to robotics. However, this approach is still lacking a stopping criterion (SC) for full automation. To maintain clarity of the learning progress, it is further essential to improve the interaction between robot and teacher. Therefore, our extended AL approach is:

Highly automated We develop a fully automated AL algorithm for a real robot, which learns new classification tasks by interacting with a human teacher and stops learning by itself when it is confident about the task. In particular, we apply different SCs together with probabilistic AL and evaluate their usefulness in the application of an automated sorting robot.

Highly interactive The robot is trained in a question-answer loop with the teacher and provides feedback by performing and visualizing the new information.

Intuitive We use speech to guide the teacher when training the robot. This procedure is suitable even for non-computer scientists.

Applicable We demonstrate the applicability of our AL approach by using a real Baxter robot.

Open-source We provide an easy-to-use implementation[1].

[1] Software available at University of Kassel: `https://git.ies.uni-kassel.de/aalwar/automated_active_learning_with_a_robot`

The remainder of this article is structured as follows: Section 2 summarizes related work in the field of AL including sorting with robots and several SCs. Section 3 describes fundamentals regarding the chosen classifier and selection strategy. It is followed by Section 4, which comprises our AL method for a sorting robot, starting with the AL environment up to the implemented SC. Several SC are evaluated in Section 5, whereas Section 6 includes a more detailed discussion.

2 Related Work

In the field of *pool-based AL*, we assume that a set of unlabeled samples is available at low or no cost but the corresponding labels are expensive, time-consuming, or difficult to acquire. Therefore, AL methods optimize the labeling process by selecting those samples that are expected to improve the classification performance the most (Settles, 2010). In the context of this article, we consider the *transductive AL* scenario (Vapnik, 2000), where the robot aims to correctly sort a selection of a-priori known objects. This is contrary to inductive learning, where the aim is to create a general model for unknown test data.

The most widely used *selection strategy* for AL is called uncertainty sampling (Lewis and Gale, 1994). The idea is to acquire labels from samples near the decision boundary (i. e., in uncertain regions) to refine the decision rule. Methods that use the expected error reduction approach (Roy et al, 2001) overcome this problem by simulating the acquisition of every sample but increase the computational complexity. Seung et al (1992) had the idea to use an ensemble of different classifiers (e.g., by bootstrapping the available training data) and select instances, where the members of the ensemble disagree the most. More recently, Kottke et al (2016) combined the idea of expected error reduction and uncertainty sampling in probabilistic AL: By introducing a prior belief on the posterior probabilities, they were able to estimate the expected error locally, which saves time and achieves superior performance.

Chao et al (2010) *applied AL* to examine the interaction between a human teacher and a robot. The robot's task was to assign objects to one out of four classes, which differed in color, shape, and size. By pointing the finger at an object, the robot asks for labels. Moreover, the human teacher can test the robot's learning progress. Cakmak et al (2010) showed that AL achieves better results compared to passive supervised learning in experiments with a robot. Subsequently, Cakmak and Thomaz (2012) investigated different types of questions regarding labeling, demonstration, and features. However, these approaches do not include some kind of autonomous sorting with a robot.

Gupta and Sukhatme (2012) used a clustering algorithm to let the *robot sort* cluttered bricks according to color and size. Abdo et al (2015) utilized robots to sort objects into buckets according to user preferences. These robots have been learning on crowd-sourced data using collaborative filtering. Then, they used spectral clustering to sort objects into buckets. In contrast to the previous section, these approaches do not use AL for training.

Herde et al (2018) presented an efficient training procedure for sorting robots and showed that probabilistic AL can be applied to this use-case. The robot actively learns to sort objects by asking a human teacher. They provide a visualization of the robot's confidence to let the teacher decide when to start autonomous sorting. The considered approaches do not investigate an automatic SC to create a fully automated AL robot for sorting tasks. Therefore, we focus on implementing a SC.

Laws and Schütze (2008) propose a *performance-based SC* by estimating the f-score of a classifier in an AL process. The score is calculated using the true positive rate, true negative rate, and false negative rate extending the idea from the generalization error used in expected error reduction. Predicting the f-score in each training iteration, they stop acquiring labels as soon as a desired f-score value is reached. Zhu et al (2010) proposed the minimum expected error method (MEE), which uses the estimation of the current classifier's expected error on all future unlabeled objects similar as before. A meta-learning approach is used by Naik et al (2013). They create features from the classification model at certain time points and train a regression algorithm to learn a model-performance-mapping. Using this mapping, they predict the performance.

Wang et al (2014) use a *gradient-based SC*. They compute the gradient of the empirical error for a given algorithm and stop if all remaining samples in the pool will barely change the model. They applied their method to Logistic Regression and Support Vector Machines (SVM). Laws and Schütze (2008) compute the gradient of either performance (based on the generalization error) or uncertainty (based on class posteriors) to determine if the pool has become uninformative such that an additional sample will not change the model significantly.

Zhu et al (2010) give an introduction to *confidence-based SC*. Their maximum uncertainty method (MU) stops if all samples from the unlabeled pool have an uncertainty value below a user-defined threshold. Their overall uncertainty method (OU) uses the average uncertainty of the pool. Vlachos (2008) proposed a method for SVM, maximum entropy models, and Bayesian logistic regression. After each acquisition, he computes the confidence (i.e., the average margin of a SVM or the entropy in Bayesian logistic regression). Once the confidence starts dropping, the AL process is stopped because no informative samples are left in the pool.

3 Fundamentals

The main idea of AL is that a selection strategy chooses an instance $\mathbf{x}$ out of an unlabeled pool $\mathcal{U}$. Then, an oracle (the human teacher) assigns a class label $y \in \{1, \ldots, C\}$, where each label corresponds to one of C different buckets. By subsequently gathering new information, the classifier improves its decision rule. We collect this information in the labeled set $\mathcal{L} = (\mathcal{L}_1, \ldots, \mathcal{L}_C)$, which consists of several subsets, each representing one class (resp. bucket).

We use *Parzen windows* for classification as we need a generative classifier to apply probabilistic AL. A kernel function K serves as a similarity score between two instances $\mathbf{x}$ and $\mathbf{x}'$ to provide class probabilities and kernel frequency estimates $k_{\mathbf{x},y}$.

$$P(y \mid \mathbf{x}) = \frac{k_{\mathbf{x},y}}{\sum_{y'} k_{\mathbf{x},y'}} \qquad \text{with} \quad k_{\mathbf{x},y} = \sum_{\mathbf{x}' \in \mathcal{L}_y} K(\mathbf{x}, \mathbf{x}') \tag{1}$$

To select the most informative instances from the unlabeled pool $\mathcal{U}$, we use *multi-class probabilistic AL* (McPAL) as proposed by Kottke et al (2017). McPAL calculates the usefulness of a sample by requiring solely the above-mentioned label frequencies $\mathbf{k_x} = (k_{\mathbf{x},1}, \ldots, k_{\mathbf{x},C})^T$ of the currently considered labeling candidate $\mathbf{x}$. In general, the approach follows a decision-theoretic idea: Considering only the region close to the candidate $\mathbf{x}$, we determine the expected value of what an additional label would change in terms of classification accuracy. The parameter M is the number of labels that can maximally be acquired. Investigating the influence of m label acquisitions in the neighborhood of $\mathbf{x}$, we expect a performance of $\text{expPerf}(\mathbf{k_x}, m)$. Accordingly, the expected performance without new information is $\text{expPerf}(\mathbf{k_x}, 0)$. As our algorithm is restricted to selecting one instance at a time, we divide the expected gain by the number of simulated label acquisitions m:

$$\mathrm{perfGain}(\mathbf{k_x}, \mathrm{M}) = \max_{m \leq M} \left(\frac{1}{m} \left(\mathrm{expPerf}(\mathbf{k_x}, m) - \mathrm{expPerf}(\mathbf{k_x}, 0) \right) \right) \tag{2}$$

$$\mathrm{expPerf}(\mathbf{k_x}, m) = \mathop{\mathbb{E}}_{\mathbf{l}} \left[\mathop{\mathbb{E}}_{\mathbf{p}} \left[\mathrm{perf}(\mathbf{k} + \mathbf{l} | \mathbf{p}) \right] \right] \tag{3}$$

To determine the expPerf function, we calculate the expected value over all possible posterior vectors **p** which follow a Dirichlet distribution. Moreover, we simulate over all possible labeling situations **l** to determine the expected gain in performance.

To consider the influence of an instance on the whole dataset, we weight the expected performance gain $\mathrm{perfGain}(\mathbf{k_x}, \mathrm{M})$ with its density $P(\mathbf{x})$. Thereby, we achieve that instances in dense regions are preferred over outliers. The final usefulness score is determined as follows.

$$\mathrm{McPalScore}(\mathbf{x}) = P(\mathbf{x}) \cdot \mathrm{perfGain}(\mathbf{k_x}, \mathrm{M}) \tag{4}$$

4 Our Method

In this section, we describe our applied AL sorting application in more detail. First, we characterize our setup and the preliminary work, followed by a description of the AL cycle and our adapted selection strategy. Finally, our SC is defined. A video showing the robot's platform together with the described AL approach applied to our Baxter robot is available (see `https://www.youtube.com/watch?v=ncZvi5-NTSI`).

4.1 Interaction Setup and Preliminary Work

For our automated AL technique, we propose to use a two-armed robot such as the Baxter robot by Rethink Robotics. Each arm is equipped with a camera and a movable gripper. The *work environment* consists of two areas as depicted in Figure 1. We place all objects which are to be classified into the area of the unlabeled pool. The area of the labeled set contains C class buckets. Between these two areas, we define a query area, which can be reached by both arms. The robot's task is to sort objects from the unlabeled pool into the buckets according to the teacher's decision rule.

To make the interactions between the robot and a human teacher convenient and intuitive, we use *communication with natural language*. Therefore, Google's speech recognition and Google text to speech are used for natural language processing. Speech recognition is required for the robot to understand human answers. Text to speech is essential for the robot to ask questions, i.e., querying the class label.

Our *object detection*, *segmentation*, and *feature extraction* techniques are based on Herde et al (2018). In the original work, every object needed to be scanned separately, which was time-consuming. In our implementation, all objects are scanned at once. As a consequence, the teacher does not need to wait for the robot and training is started immediately.

The object's *features* are color, shape, and size. To describe the shape of an object, we use the Hu moment invariants. The object's size is determined by calculating the area of the object's contour. A hue-saturation-histogram is used to characterize the color of an object.

4.2 Active Learning Cycle and Selection Strategy

A robot R and a human teacher T are active participants during the AL training phase, where objects must be sorted from the unlabeled pool $\mathcal{U}$ into buckets based on their extracted features (cf. Algorithm 1). The robot selects one of the available objects in the unlabeled pool using McPAL (cf. Section 3), whereupon it is moved into the query area QA. The object gets labeled by the human teacher using the desired bucket number (class y). Then, it is moved to the corresponding bucket ($\mathcal{L}_y$) by the robot. After adding this new information to the Parzen window classifier (PWC, cf. Section 3), which is used as a classifier in our AL approach, this procedure continues until the SC is reached (cf. Section 4.3). This finishes the training phase and the robot starts sorting the remaining objects autonomously according to predictions of the PWC.

Our selection strategy is based on McPAL. In order to compare quantities of the $McPalScore$, we normalize it with the maximally reachable gain, which is calculated with an empty labeled set. Hence, $\mathbf{k}$ is a vector consisting of zeros. This is necessary because the score depends on the number of classes. The resulting score for selecting samples is determined as follows:

$$\text{McPalScore}_{\text{norm}}(\mathbf{x}) = P(\mathbf{x}) \cdot \text{perfGain}_{\text{norm}}(\mathbf{k}_{\mathbf{x}}, M) \tag{5}$$

$$\text{perfGain}_{\text{norm}}(\mathbf{k}_{\mathbf{x}}, M) = \frac{\text{perfGain}(\mathbf{k}_{\mathbf{x}}, \text{M})}{\text{perfGain}(\mathbf{0}, 1)} \tag{6}$$

Algorithm 1: Active learning cycle for a sorting robot.

Input: Robot R, Teacher T, Number of buckets C, Labeled set $\mathcal{L} = (\mathcal{L}_1, \ldots, \mathcal{L}_C)$, Query area QA, Unlabeled pool $\mathcal{U}$, Classification model M, Stopping criterion SC_X, User-predefined stopping threshold θ_X

repeat
    R selects object $\mathbf{x}$ from $\mathcal{U}$ using the $McPALScore_{norm}$
    R puts $\mathbf{x}$ into QA and asks T for label y
    T provides y
    R places $\mathbf{x}$ into the labeled set $\mathcal{L}_y$
    $\mathcal{L}_y \leftarrow \mathcal{L}_y \cup \mathbf{x}$
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathbf{x}$
    R updates model M by passing new labeled set $\mathcal{L}$
until $SC_X(\mathcal{U}) \leq \theta_X$
for all $\mathbf{x} \in \mathcal{U}$ **do**
    M provides label y for $\mathbf{x}$
    R places $\mathbf{x}$ into labeled set $\mathcal{L}_y$
end for

4.3 Stopping Criterion

One of the many challenges regarding automated AL approaches is to find an appropriate SC. It is important to determine the optimal point in time when the learning process can be stopped: The robot should be able to sort the remaining samples autonomously, at the same time the number of teacher queries should be as low as possible. We propose to use a confidence-based SC based on McPAL by *extending MU*. Instead of using the entropy to calculate the classifier's uncertainty, we use the normalized probabilistic gain as discussed before:

$$SC_{\text{MG}}(\mathcal{U}) := \max_{\mathbf{x} \in \mathcal{U}} \left(\text{perfGain}_{\text{norm}}(\mathbf{x}) \right) \tag{7}$$

The idea behind this is as follows: The probabilistic gain describes the gain in performance of the candidate **x**. If all instances improve the classification results by less then a pre-defined threshold θ_{MG}, we assume that all interesting instances have already been selected or that additional instances will not provide further significant information. The choice of a proper threshold θ_{MG} is discussed in Section 5 and Section 6.

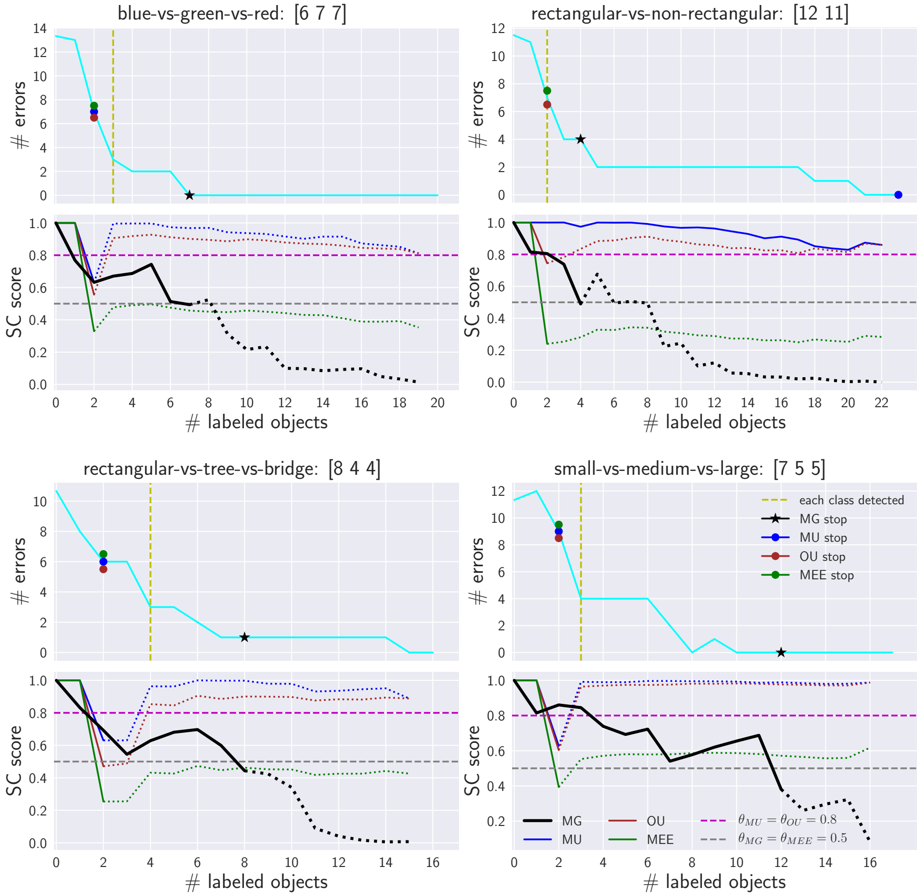

Figure 2: Experimental results on four datasets: The upper plot of each dataset illustrates the number of classification errors on the unlabeled objects along the number of labeled objects as the cyan line. Moreover, the vertical yellow line indicates the number of labeled objects until at least one object of each class has been labeled, whereas a star/circle marks the point of stopping recommended by the respective SC. The lower plot illustrates the scores computed with the different SC along the number of labeled objects as solid lines. The corresponding thresholds are represented by the dashed lines.

To assist the human teacher during training, we use a *visualization tool* to display the certainty of the remaining unlabeled objects. Herde et al (2018) already provided a visualization tool for the Baxter robot, that shows each object in its predicted bucket. The certainty of the prediction is shown by the position within the bucket. We extended this idea by visualizing the threshold of the SC using a green line. Thereby, the teacher is able to supervise the learning progress. If all objects are below the drawn threshold the SC is reached.

5 Experiments and Results

In this section, we evaluate the effectiveness of our SC MG on four different *datasets* with varying tasks. The objects of the datasets have been building blocks and wooden toys, which were scanned by the robot. We provide the dataset names including their class distributions in the headings of each plot in Figure 2. The names indicate the classification problems to be solved as well. For example, an object of the dataset *rectangular-vs-non-rectangular* has to be classified according to its shape, which can be either rectangular or non-rectangular.

Since we assume a *transductive learning setting*, the goal is to classify the objects in the unlabeled pool correctly by either asking the teacher or using the predictions of the classifier. Thereby, we aim to minimize the labeling effort. In contrast to inductive learning, we do not have a hold-out evaluation set. We evaluate the performance of the classifier by counting the number of wrong predictions for the unlabeled objects. Hence, this is a one-shot scenario and as our methods are deterministic, there is no need to repeat the experiments.

At the start of each *experiment*, all objects are unlabeled and added to the labeled pool one after the other (cf. Algorithm 1). We employ the PWC classifier, which is retrained on the labeled objects in each AL cycle. As kernel, we utilize the similarity measure proposed in Herde et al (2018). The object selection is executed according to the strategy McPAL, which is superior to expected error reduction and uncertainty sampling as shown in Herde et al (2018). After each cycle, we determine the number of classification errors on the remaining unlabeled objects and the scores of all SC. As *baseline algorithms* for SC, we use MU, OU, and MEE proposed by Zhu et al (2010) and defined as follows:

$$SC_{\mathrm{MU}}(\mathcal{U}) := \max_{\mathbf{x}\in\mathcal{U}} \left(-\sum_{y} P(y|\mathbf{x}) \log P(y|\mathbf{x}) \right) \tag{8}$$

$$SC_{\mathrm{OU}}(\mathcal{U}) := \frac{\sum_{\forall \mathbf{x}\in\mathcal{U}} - \sum_{y} P(y|\mathbf{x}) \log P(y|\mathbf{x})}{|\mathcal{U}|} \tag{9}$$

$$SC_{\mathrm{MEE}}(\mathcal{U}) := \frac{1}{|\mathcal{U}|} \sum_{\mathbf{x}\in\mathcal{U}} (1 - \max_{y\in Y} P(y|\mathbf{x})) \tag{10}$$

All SC are based on user-defined thresholds. For our MG criterion, we fixed the threshold θ_{MG} at 0.5 by experimental evaluation. The authors of MU, OU, and MEE recommended to rely on the threshold $\theta_{\mathrm{MU}} = \theta_{\mathrm{OU}} = \theta_{\mathrm{MEE}} = 0.1$ (Zhu et al, 2010). However, none of the SC was reached using the recommended threshold. Therefore, we adapted the threshold to $\theta_{\mathrm{MEE}} = 0.5$ and $\theta_{\mathrm{MU}} = \theta_{\mathrm{OU}} = 0.8$. For evaluation purposes, we did not stop learning but repeated the AL cycle until all objects were labeled.

The corresponding results are summarized in Figure 2. We observe a correlation between the MG curve and the curve of classification errors. The point of stopping is reasonable on all four datasets although it is not optimal. Decreasing the threshold could lead to better results for some datasets but this increases the risk of stopping too late. In this application, the most important point is that we were able to achieve good results with much fewer label acquisitions compared to a fully supervised approach. The results of the other three criteria, namely, MU, OU, and MEE, are not suitable to define a well-performing SC. For each of them, no global threshold leads to reasonable results in terms of stopping points across all four datasets. Moreover, none of them shows a trend that follows the curve of the classification error.

6 Discussion

By introducing a SC for a real robot, we have come a good step closer to our vision of autonomous learning on robots. But more challenges need to be worked on. In our AL scenario, speech recognition is limited to numbers, e.g., *three*, and *yes* and *no*. The challenge in using speech for communication is to cope with incomprehensible keywords provided by the teacher. In such a case, we implemented the robot to ask again.

The influence of possible environmental changes on the robot has to be considered as well. The incidence of light strongly contributes to how well the images and thus the objects are recognized and therefore segmented. In order to cope with possible lighting changes, we offer an exposure adaptation before training the robot. Furthermore, it must be taken into account that the arms of the robot could collide during the sorting process. To prevent this, we defined waiting areas, where the arms stay as long as the query area is occupied.

However, some aspects cannot be improved any further. For example, the Baxter robot sometimes grasps the object incorrectly because the movements of the robot are affected by trembling. Besides, errors can occur during gripping, e.g., when the object slips out of the gripper.

Furthermore, the question arises, whether the Baxter robot in our scenario can be replaced by any other robot. Essentially, the robot in use requires two arms with grippers and cameras for the AL cycle. A display for the visualization tool as well as a microphone and loudspeaker for communication are optional since we offer communication via the console. It is of advantage if the Robot Operating System (ROS) is used as we launch our application using ROS.

For future applications, we consider the adaption of the automated AL approach to other robotics domains, e.g., quality inspection. In particular, the extension of two-dimensional data to three dimensions is required. From this, further challenges will arise, e.g., calibration of the three-dimensional sensors. Moreover, we need to investigate more advanced features, e.g., derived from deep learning methods, and the interplay of the high-dimensional feature space as well as the limited amount of labeled training samples with AL and SC.

Apart from that, the intuitiveness of our approach can still be increased. For example, it should be feasible for the robot to ask questions as well. The robot could use this add-on to expand his knowledge by questioning it.

7 Conclusion

Our vision is to create a fully automated and highly interactive as well as intuitive AL process for sorting robots. In this article, we presented an extension of the AL sorting process for robots by Herde et al (2018) by proposing a SC to fully automate the AL process. We showed that our implemented SC_{MG} achieved better results in the case of AL compared to other SCs. Furthermore, we utilized speech as a communication medium to enable a more natural interaction between human teacher and robot. Although we made progress towards our vision, improvement and enhancement is still possible.

Acknowledgements This work results from the project DeCoInt2, supported by the German Research Foundation (DFG) within the priority program SPP 1835: "Kooperativ interagierende Automobile" grant number SI 674/11-1. We would like to gratefully acknowledge the work of Fabien Chalas and Alexander Kühn helping us in the realization of the AL software for the Baxter robot.

References

Abdo N, Stachniss C, Spinello L, Burgard W (2015) Robot, Organize my sSelves! Tidying Up Objects by Predicting User Preferences. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), Institute of Electrical and Electronics Engineers (IEEE), pp. 1557–1564. DOI: 10.1109/ICRA.2015.7139396.

Cakmak M, Thomaz AL (2012) Designing Robot Learners that Ask Good Questions. In: 2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Institute of Electrical and Electronics Engineers (IEEE), pp. 17–24. DOI: 10.1145/2157689.2157693.

Cakmak M, Chao C, Thomaz AL (2010) Designing Interactions for Robot Active Learners. IEEE Transactions on Autonomous Mental Development 2(2):108–118, Institute of Electrical and Electronics Engineers (IEEE). DOI: 10.1109/TAMD.2010.2051030.

Chao C, Cakmak M, Thomaz AL (2010) Transparent Active Learning for Robots. In: 2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Institute of Electrical and Electronics Engineers (IEEE), pp. 317–324. DOI: 10.1109/HRI.2010.5453178.

Gupta M, Sukhatme GS (2012) Using Manipulation Primitives for Brick Sorting in Clutter. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), Institute of Electrical and Electronics Engineers (IEEE), pp. 3883–3889. DOI: 10.1109/ICRA.2012.6224787.

Herde M, Kottke D, Calma A, Bieshaar M, Deist S, Sick B (2018) Active Sorting – An Efficient Training of a Sorting Robot with Active Learning Techniques. In: 2018 International Joint Conference on Neural Networks (IJCNN), Institute of Electrical and Electronics Engineers (IEEE), pp. 1–8. DOI: 10.1109/IJCNN.2018.8489161.

Kottke D, Krempl G, Lang D, Teschner J, Spiliopoulou M (2016) Multi-Class Probabilistic Active Learning. Frontiers in Artificial Intelligence and Applications 285:586–594. DOI: 10.3233/978-1-61499-672-9-586.

Kottke D, Calma A, Huseljic D, Krempl G, Sick B (2017) Challenges of Reliable, Realistic and Comparable Active Learning Evaluation. In: Proceedings of the Workshop and Tutorial on Interactive Adaptive Learning co-located with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2017), Vol. 1924, pp. 1–13.

Laws F, Schütze H (2008) Stopping Criteria for Active Learning of Named Entity Recognition. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Vol. 1, p. 8. URL: `https://www.aclweb.org/anthology/C08-1059`.

Lewis DD, Gale WA (1994) A Sequential Algorithm for Training Text Classifiers. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94), Croft B, van Rijsbergen C (eds), Springer, London (United Kingdom), Vol. 29, pp. 3–12. DOI: 10.1007/978-1-4471-2099-5_1.

Naik AW, Kangas JD, Langmead CJ, Murphy RF (2013) Efficient Modeling and Active Learning Discovery of Biological Responses. PLoS ONE 8(12):1–13. ISSN: 1932-6203, DOI: 10.1371/journal.pone.0083996.

Roy N, Mccallum A, Com MW (2001) Toward Optimal Active Learning Through Monte Carlo Estimation of Error Reduction. In: ICML, San Francisco (USA), pp. 1–8.

Settles B (2010) Active Learning Literature Survey. Tech. Rep., University of Wisconsin–Madison. URL: `http://burrsettles.com/pub/settles.activelearning.pdf`. Computer Sciences Technical Report 1648.

Seung HS, Opper M, Sompolinsky H (1992) Query by Committee. In: Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT '92), pp. 287–294. DOI: 10.1145/130385.130417.

Vapnik V (2000) The Nature of Statistical Learning Theory, 2nd edn. Information Science and Statistics, Springer, New York (USA). ISBN: 978-1-475732-64-1, DOI: 10.1007/978-1-4757-3264-1.

Vlachos A (2008) A Stopping Criterion for Active Learning. Computer Speech and Language 22(3):295–312. DOI: 10.1016/j.csl.2007.12.001.

Wang W, Cai W, Zhang Y (2014) Stability-based Stopping Criterion for Active Learning. In: 2014 IEEE International Conference on Data Mining (ICDM), Institute of Electrical and Electronics Engineers (IEEE), pp. 1–6. DOI: 10.1109/ICDM.2014.99.

Zhu J, Wang H, Hovy E, Ma M (2010) Confidence-based Stopping Criteria for Active Learning for Data Annotation. ACM Transactions on Speech and Language Processing (TSLP) 6(3):1–24. DOI: 10.1145/1753783.1753784.