Active Learning with Mixed Query Types in Learning from Demonstration

Maya Cakmak and Andrea L. Thomaz MAYA,ATHOMAZ@CC.GATECH.EDU
Georgia Institute of Technology, 801 Atlantic Dr. Atlanta, GA 30332

Abstract

Active Learning (AL) has recently drawn a lot of attention within the Learning from Demonstration (LfD) community. We are particularly interested in the potential of AL to significantly improve the efficiency of learning new skills from human demonstrators. In this paper we review the different types of queries proposed in the AL literature and exemplify how they can be applied to LfD problems. We also discuss the factors that affect query selection in a mixed query-type scenario.

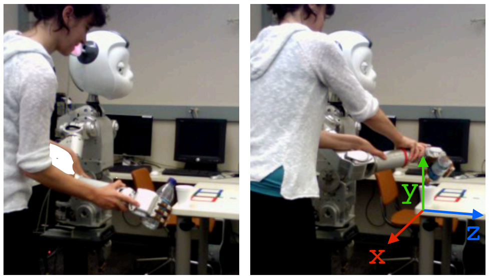


Figure 1. Start and end snapshots of a human giving a kinesthetic demonstration to a robot for teaching the skill of pouring water from a bottle into a cup.

1. Introduction

Learning from Demonstration (LfD) is a compelling approach for programming new skills on a robot. However providing a large number of demonstrations of the same skill can become cumbersome for the teacher, when it is necessary for achieving robustness and generalizability. Thus the a robot needs to use a teacher's time efficiently.

Active Learning (AL) addresses this very issue by letting the learner choose the examples from which it is going to learn (Angluin, 1988; Cohn et al., 1995). The learner picks such *queries* from viable unlabeled samples based on a strategy such as reducing uncertainty or maximizing information gain (Settles, 2010). AL has been shown to improve sample efficiency in a large range of applications in fields such as Computer Vision and Text Classification.

This potential of AL has also been noticed by the Robotics community (Lopes & Oudeyer., 2010; Martinez-Cantin et al., 2010). Active learning has been applied to a number of machine learning problems in LfD. For example, *confidence based autonomy* uses uncertainty sampling to select states in which the the learning agent requests a demonstration while learning a policy (Chernova & Veloso, 2007). Lopes *et al.* (2009) use AL to select the states in which an expert human should be queried for an appropriate action. Similarly, in (Gribovskaya et al., 2010), the robot actively selects points outside the region of stability of a learned policy, and requests demonstrations from these states.

AL in the context of LfD is particularly interesting from a human-robot interaction (HRI) perspective. In the traditional LfD setting the learner is passive and the interaction is fully controlled by the teacher. Making the learner active gives more control of the interaction to the robot. User studies by Cakmak *et al.* (2010) show that a robot that constantly makes queries results in an undesirable interaction. They propose *teacher-triggered* queries as a mechanism better suited for HRI scenarios. Rosenthal *et al.* (2009) investigate how the accuracy of a human teacher's answers to a robot's questions could be improved by augmenting queries with context information or uncertainty.

While these works have demonstrated some of the benefits and challenges in using AL methods in Robotics, we believe that there is still a large gap between the

Appearing in *Proceedings of the ICML Workshop on New Developments in Imitation Learning*, Bellevue, WA, USA, 2011. Copyright 2011 by the author(s)/owner(s).

two fields. In particular we believe that LfD could benefit much more from the modern techniques developed by the AL community and go beyond simple label queries. Secondly, we believe that this gap is partly due to the fundamental differences between the learning problems in Robotics and the fields that motivate most of the methods developed in AL. Therefore, we think that robotics could pose problems for which new AL methods will need to be developed.

In this paper we review some of the less explored methods in AL and discuss ways they can be used in learning skills from human demonstrations. In particular we focus on the different types of queries and what it means for a robot to make these queries. We exemplify these different types of queries based on a concrete skill learning method with a sample data for teaching a particular skill. Finally we discuss factors that influence query selection given the choice between different query types. We hope that our paper draws the attention of the LfD community to the large range of methods in AL, points out their potential benefits, and lays out several new research problems.

2. Approach

In this section we first describe a skill learning framework and then exemplify the different query types on a concrete example in this framework.

2.1. Skill Representation

We consider a general skill representation that is equivalent to *options* in Hierarchical Reinforcement Learning. A skill is represented by a tuple $\langle \mathcal{I}, \pi, \beta \rangle$; an initiation set $\mathcal{I}$, a policy π and a termination set β. Assume x indicates the robot's state and S is the set of all possible states. The skill is available in state x if and only if $x \in \mathcal{I} \subseteq S$. If the skill is executed, then actions are selected according to π until the skill terminates according to $\beta : S \rightarrow [0, 1]$.

The robot state can be represented in a number of different ways. Among the commonly used in LfD are joints of the robot, end-effector configuration (position and orientation) relative to the robot, or relative to a reference/goal coordinate frame in the world.

In this paper we consider the example skill of pouring water into a cup (Fig. 1). We represent the state x of the robot as the end-effector configuration relative to the cup coordinate frame (which is axis-parallel to the robot's coordinate frame, but shifted in position). For the policy we consider a dynamical system model, as in (Gribovskaya et al., 2011), that maps a state x to a change in the state $\dot{x}$; *i.e.* $\pi(x) = \dot{x}$.

2.2. Skill Learning

A demonstration consists of a sequence of state-action pairs; $\mathcal{D}_i = \{(x_{i0}, \dot{x}_{i0}), ..., (x_{iN_i}, \dot{x}_{iN_i})\}$. A skill is learned from a set of m demonstrations $\{\mathcal{D}_i\}_{i=1}^m$ as follows. The initiation set $\mathcal{I}$ and termination set β are modeled with multivariate Gaussian distributions, $\mathcal{N}_\mathcal{I}$ and $\mathcal{N}_\beta$, fit to $\{x_{i0}\}_{i=1}^m$ and $\{x_{iN_i}\}_{i=1}^m$ respectively. The probability $\mathcal{P}(x; \mathcal{N}_\mathcal{I})$ is thresholded to determine whether $x \in \mathcal{I}$. Similarly, the skill terminates if $\beta(x) = \mathcal{P}(x; \mathcal{N}_\beta)$ is above a threshold.

The policy is modeled using Kernel regression with the Nadaraya-Watson estimator. In any state x, the velocity $\dot{x}$ is estimated as a locally-weighted average of the velocities in the provided demonstrations. The policy is written as $\pi(x) = \sum_{i=1}^m \sum_{j=1}^{N_i} K(x, x_{ij})\dot{x}_{ij}$ where K is a kernel serving as a weighting function.

For learning the pouring skill we use a Gaussian kernel which produces exponentially decreasing weights as the regression point moves away from the demonstration points. Regression uses five demonstrations, of different number of frames, collected through kinesthetic interactions. Fig. 2 visualizes the five demonstrations, the learned $\mathcal{N}_\mathcal{I}$ and $\mathcal{N}_\beta$ and some trajectories produced by the policy for fifty starting points randomly sampled from $\mathcal{N}_\mathcal{I}$.

2.3. Active Skill Learning

The unit of data transfer between the teacher and the learner in the LfD setting is *demonstrations*. The LfD problem considered in this paper involves three separate learning problems for modeling $\mathcal{I}$, π and β. A passive learner learns all three from demonstrations. Thus it is natural that an active LfD process involves queries that lets the learner receive demonstrations that it chooses.

On the other hand, the input to the three sub-problems are at a smaller scale than whole demonstrations. A demonstration provides, a single data point for learning $\mathcal{I}$ and β each (x_{i0} and x_{iN_i}, both implicitly labelled as positive) and N_i-1 data points $(x_{ij}, \dot{x}_{ij})$ for learning $\pi(x) = \dot{x}$. Thus in our approach we want to allow possibly lower-cost queries that directly address these sub-problems.

The active LfD process is further shaped by the human-robot interaction through which data is transferred from the teacher to the learner. It is imperative that the queries are physically possible and interpretable by a human teacher. For instance it might not be possible to provide isolated $(x_{ij}, \dot{x}_{ij})$ pairs through kinesthetic demonstrations. Similarly, a question by the learner that involves naming features, requires that

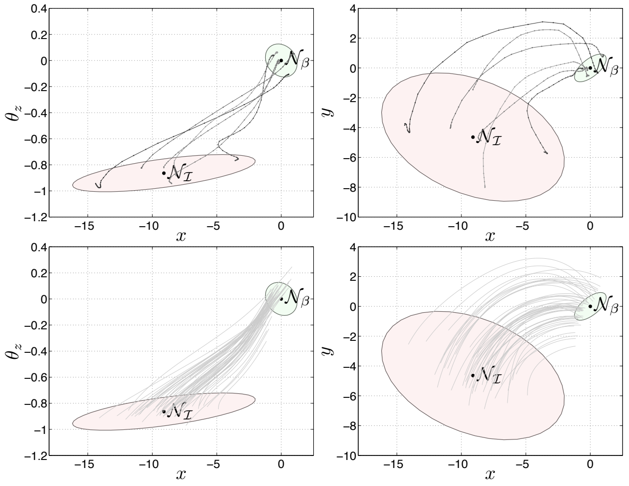


Figure 2. (Top) The model for the pouring skill learned from five demonstrations, projected onto two 2-dimensional sub-spaces (x-y and x-θ_z). (Bottom) Fifty sample trajectories produced by the model in the same sub-spaces.

the feature is observable and intuitive for the human.

Accordingly, we identify four types of queries for active LfD: label, instance, partial instance, and feature queries. In the remainder of this section we detail each of these query types.

2.3.1. Label queries

The common type of query in active learning is a label query (also called a membership query), where the learner requests a label for an instance. The instance can be chosen from a pool of unlabeled instances or instantiated by the learner in some way. In our LfD setting we have three types of label queries. The first two are individual label queries for learning $\mathcal{I}$ and β, while the third is a label query for a whole demonstration.

Label queries for $\mathcal{I}$ involve the robot learner moving to a particular configuration $\tilde{x}_\mathcal{I}$ and asking whether the skill can be initiated in this configuration. If the answer is "yes", $\mathcal{N}_\mathcal{I}$ is updated to account for $\tilde{x}_\mathcal{I}$.

Label queries for β involve the robot learner moving to a particular configuration $\tilde{x}_\beta$ and asking whether the skill can end in this configuration. If the answer is "yes", $\mathcal{N}_\beta$ is updated to account for $\tilde{x}_\beta$.

Demonstration-label queries involve the robot performing the skill from start to end and asking whether it was successful. If the answer is "yes", the performance is added as a new demonstration and the skill is updated. Note that in robotics it is important to have the ability to verify the safety of such queries.

The choice of label queries for $\mathcal{I}$ and β can be guided by different objectives such as maximizing or minimizing variance some state variables. Similarly, queries for learning the policy can be chosen to minimize variance in the estimated policy. Cohn *et al.* (1995) describe methods for choosing queries among a candidate set such that variance of the learned model (mixture of Gaussians or locally-weighted regression) is minimized. Several other methods for choosing queries in regression problems are described in (Castro et al., 2005).

In our scenario the relationship between the three subproblems can also guide the choice of queries. For instance, in the example shown in Fig. 2, we observe that some trajectories produced by the learned policy do not terminate according to β even though they are initiated from $\mathcal{I}$. A label query for β can be made at the end points of these trajectories, in an attempt to preserve the policy that produces these trajectories. If these end points are indeed valid termination points, the learner can update β and not need any modifications for π. A label query for $\mathcal{I}$ can follow this to make sure that the start point of the trajectory is in $\mathcal{I}$. Then the robot can reproduce the trajectory to make a demonstration-label query. This example is illustrated in Fig. 3 (Example 1).

Negative Examples. In the traditional LfD setting, negative examples are not common. A skill is modeled after demonstrations of successful skill performances, and it is considered unnatural to demonstrate "what not to do". However negative examples naturally arise in an active LfD setting. These examples can be exploited in two ways:

1. The model can be updated from negative examples, such that the probability of the positive data being generated from the model is maximized and the probability of the negative data being generated by the model is minimized.
2. Even if the model is not affected by the negative examples, the active learning strategy should be adapted as to reduce the probability of the same part of the space being queried again. This will facilitate the model progress in unexplored directions.

For demonstration-label queries, negative examples are particularly interesting since they raise a credit assignment problem. For instance, if a performance by the robot is deemed unsuccessful, this does not necessarily imply that that its starting point should not belong to $\mathcal{I}$. Therefore, these queries can focus on learning π by choosing the start and end of queries conservatively from $\mathcal{I}$ and β.

2.3.2. Instance queries

Instance queries (known as active class selection (Lomasky et al., 2007)) consist of requesting an example from a certain class. While this query leaves a lot of flexibility to the teacher, it can provide some control to the learner. In our LfD setting we consider the following queries.

Demonstration queries given $\tilde{x}_{\mathcal{I}}$ are requests for a demonstration starting from a configuration that the learner chooses. This type of query can naturally follow a label query on $\tilde{x}_{\mathcal{I}}$, given that it turns out to be $\in \mathcal{I}$. Note that the policy estimation method will allow the learner to hypothesize a trajectory to perform the skill starting from $\tilde{x}_{\mathcal{I}}$ as well as the new points that end up in $\mathcal{I}$ after being updated with $\tilde{x}_{\mathcal{I}}$. Thus a useful criteria for deciding whether to make this type of query is whether the hypothesized trajectory terminates according to β and the confidence over the trajectory. If it does, the robot can perform this trajectory to make a demonstration-label query (ask whether the skill was performed successfully) as in Example 1 in Fig. 3.

In other cases a hypothesized trajectory that starts at a point known to be $\in \mathcal{I}$ might not terminate according to β. This means that the policy fails to produce a valid trajectory starting at this point in order to reproduce the skill. This is a condition well suited for a demonstration request given the start point. Example 2 in Fig. 3 illustrates one such situation.

Demonstration queries given $\tilde{x}_{\beta}$ are requests for a demonstration that ends at a configuration chosen by the learner. Note that with this type of query the teacher will need to first take the robot to a starting configuration then provide a demonstration that terminates in the requested configuration. As a result the end configuration might not precisely match the requested configuration.

Partial demonstration queries are requests for demonstration sub-sequences where the rest of the sequence is performed by the robot. For instance the robot can start and perform a skill up to a point and request that the teacher completes the demonstration. Similarly the teacher can start a demonstration and the robot can take over after a certain point.

Instance queries for $\mathcal{I}$ or β individually would be requesting "another" configuration from which the skill can be initiated, or at which the skill can terminate. These are rather unnatural and provide little control to the learner. As a result it is difficult to estimate the information gain from such queries.

2.3.3. Partial Instance queries

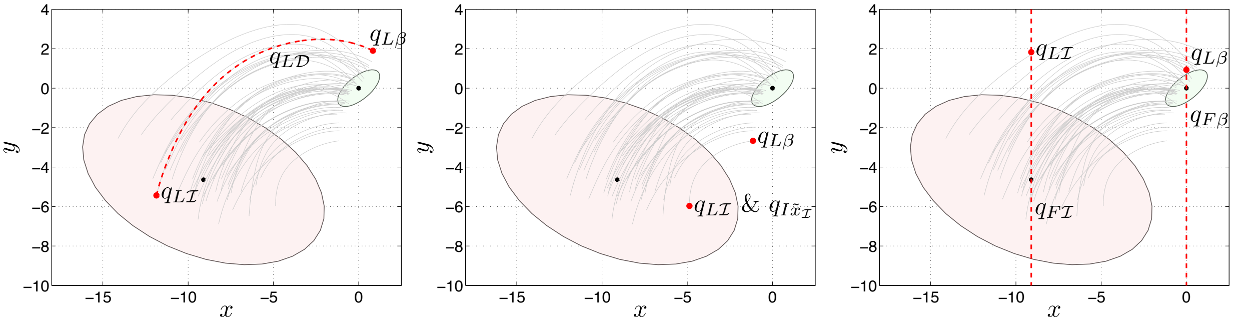
Instance and label queries are both fundamentally requests for missing information, but they are at two extremes where the missing information is *a label for a given state* or *a whole state for given a label*. From this perspective, the learner can make queries that lie along a spectrum of specificity, for instance requesting the rest of the state given a partial state and a label.

In LfD a particular type of kinesthetic demonstrations proposed in (Calinon & Billard, 2009) resemble such partial queries. In these demonstrations the person controls some of the joints while the robot controls the others in a full-length demonstration. This interaction is made possible by the skills being learned in the joint space of the robot. In general, providing such interactions in any state space is non-trivial. For instance it might be difficult for the robot to control the end-effector position while allowing the teacher to control its orientation, since both are determined by all joints of the robot.

One possible method that could allow partial instance queries is to get full instance queries but ignore or fix the irrelevant part. This requires the ability to indicate subspaces in a way that is intuitive to human teachers. One example is splitting the end-effector configuration as *position* and *orientation*. In the example skill from Sec. 2.2 the robot could go to a certain position within or close to β and request that the teacher shows the *orientation* of the arm at the current *position* as an end configuration for pouring. The control of the full configuration can still be given to the teacher hoping the position will be kept close to the one specified by the robot.

2.3.4. Feature queries

Feature queries (Raghavan et al., 2006; Druck et al., 2009) (also referred to as feature relevance queries) ask whether a feature is important or relevant for the learning problem at hand. In our LfD setting they correspond to asking whether a state variable is relevant for a skill. Such information can be directly used in scenarios where the skills are learned in sub-spaces of the original state, *i.e.* when a feature selection process preceds learning. For instance in (Muhlig et al., 2009) the robot tries to choose among a large number of state variables such as end-effector configurations relative to salient objects in the environment, or relative to the start configuration of the demonstration trajectory.



Examples		Query Type	Symbol	Question	Response
1 (left)	1	Label query for β	$q_{L\beta}$	*Can pouring end in this configuration?*	Yes
	2	Label query for $\mathcal{I}$	$q_{L\mathcal{I}}$	*Can pouring start in this configuration?*	Yes
	3	Demo. label query	$q_{L\mathcal{D}}$	*Can I pour like this?*	Yes
2 (middle)	1	Label query for β	$q_{L\beta}$	*Can pouring end in this configuration?*	No
	2	Label query for $\mathcal{I}$	$q_{L\mathcal{I}}$	*Can pouring start in this configuration?*	Yes
	3	Demo. query given $\tilde{x}_\mathcal{I}$	$q_{I\tilde{x}_\mathcal{I}}$	*Can you show pouring from here?*	$\mathcal{D}_{m+1}$
3 (right)	1	Feature query for $\mathcal{I}$ (y)	$q_{F\mathcal{I}}$	*Does height matter at the start of pouring?*	No
	2	Label query for $\mathcal{I}$	$q_{L\mathcal{I}}$	*Can I start pouring in this configuration?*	Yes
	3	Feature query for β (y)	$q_{F\beta}$	*Does height matter at the end of pouring?*	Yes
	4	Label query for β	$q_{L\beta}$	*Can I end pouring in this configuration?*	No

Figure 3. Three example query sequences and illustration of queries on the two dimensional subspace x-y.

A critical issue for feature queries is the communication about state variables with human teachers. Seemingly intuitive state variables, such as "orientation around the y-axis relative to an external coordinate frame" might be difficult to interpret for an inexperienced teacher. Communicative behaviors such as moving back and forth between two values on the referred variable while keeping all the other variables constant might be useful.

In our LfD setting, we consider three types of feature queries analogous to the label queries in Sec. 2.3.1.

Feature queries for $\mathcal{I}$ involve the learner asking whether a state variable matters in the starting configuration. If a state variable is irrelevant it is expected that $\mathcal{N}_\mathcal{I}$ will have a large variance on along that variable. This information could be incorporated in the model for $\mathcal{I}$ by increasing the variance $\mathcal{N}_\mathcal{I}$ along the state variable known to be irrelevant. In cases where the model cannot be directly updated the information can still be used to guide the choice of label queries. For instance a label query $\tilde{x}_\mathcal{I}$ can be varied from $\mathcal{N}_\mathcal{I}$ more in the variables known to be irrelevant and less in the ones known to be relevant.

An example from the pouring skill is asking whether the starting height (*i.e.* the y coordinate of the end effector position relative to the cup) matters in pouring. If the answer is "no" the robot can make label queries for $\mathcal{I}$ that is different from the mean of $\mathcal{N}_\mathcal{I}$ by a large difference in the y coordinate. Given the outcome of the feature query, the expectation will be that the point will still be part of $\mathcal{I}$. This lets the robot quickly expand $\mathcal{I}$ and increase the applicability of the skill. This is illustrated in Fig. 3 (Example 3).

Features queries for β involve the learner presenting asking whether a state variable matters in the ending configuration of a skill. Information about relevance of features for β can be incorporated in learning in a similar way as the information obtained from feature queries for $\mathcal{I}$.

For the pouring skill we can assume that the answer will be "yes" if the robot asks whether the height of the end effector matters at the end configuration of pouring. In that case, the robot can make label queries for β at points that are not too far from the mean of $\mathcal{N}_\beta$ in the y coordinate. Given that height is important at the end configuration, we can expect that such queries will quickly reveal the boundary between valid and invalid termination points. This is also illustrated in Fig. 3 (Example 3).

Features queries for π involve asking about the relevance of state variables during the performance of the skill. As in the other feature queries, directly using this information might not be straight forward, however can greatly benefit the active learning process with other types of queries. For example, if a state variable is barely changed by the learned policy, the learner can explicitly ask whether it is important to maintain this state variable at the corresponding value. If the answer is "yes", the learner can refrain from varying this state variable when making label queries.

2.4. Combining Different Query Types

The types of queries outlined above provide a rich repertoire for a robot learner to choose from. In order to use these effectively the learner needs a *query selection strategy* that chooses the best query. One such strategy in the literature is cost-sensitive active learning (Vijayanarasimhan & Grauman, 2009). This involves balancing the *effort* required to provide a certain type of annotation (*e.g.* answering questions about image content versus segmenting items in an image) with the *informativeness* of the annotation. This can be framed as a decision theory problem, in which the learner tries to decide which query has the highest utility. There are several factors that affect the utility of queries as discussed in the following.

Informativeness. A learner first needs to be able to quantify the benefit of alternative queries of the same type, for each type of query. Several such metrics of usefulness for queries can be found in the literature that introduced or studied the different types of queries. Secondly, given the best queries for each type, the learner needs to be able to compare the usefulness of different types of queries.

Cost. The cost of making a query also needs to be considered in comparing alternatives. This can involve several measures of cost such as the total time taken by the robot to make the query, the time taken by the human to respond to the query, effort or energy required to make a query or respond to it and the safety/risk of exploring a state that has not been visited before. These different measures of cost can be combined to asses the overall cost of making a query, possibly with predefined priorities for each measure. For example if more priority is given to reduce the cost in terms of effort spent by the teacher, a demonstration label query (robot performs the skill and asks if it was successful) might be a better choice as compared to requesting a demonstration from the human. Similarly, if time is the main measure of cost, short queries with "yes" or "no" answers should have higher chance of being selected.

Human interaction constraints. As illustrated by the examples in Fig. 3 certain orders of queries feel more natural from the human teacher's perspective. This can be captured by increasing the utility of potential "follow-up" queries after a query is made. We can expect that this will reduce the "context switching" required by the human, and that they will more readily respond to the followup query. Similarly, switching back and forth between questions about $\mathcal{I}$ and β can be more costly than first asking all questions about $\mathcal{I}$ and then all questions about β. Such preferences that are invariant across humans can be discovered or validated through user-studies.

In addition, certain types of queries that are valid alternatives for the learner seem impractical (*e.g.* requesting isolated $(x, \dot{x})$ pairs) or unnatural (*e.g.* asking for a trajectory that ends a desired point) for human teachers.

Teacher differences. The query selection mechanism can also be adapted to individual users depending on their recent history (*performance* or *compliance*) or known preferences. For example, an experienced teacher who provides smooth trajectories efficiently could be given more demonstration queries. Similarly, if a teacher is known to enjoy or get frustrated by certain types of queries (*e.g.* the robot controlling a subset of the joints) the utility should be increased or reduced for these queries.

3. Future Work

We are currently working on implementing all the different types of queries explained in this paper and comparing alternative approaches for query selection. As part of this work, we will perform user-studies to evaluate the feasibility and and measure the cost of different types of queries. After several iterations of the query selection mechanism, we plan to evaluate our system by comparing it to purely passive learning and active learning with a single type of query.

4. Conclusion

In this paper we review the different types of queries studied in the Active Learning literature and propose ways that they could be used in a Learning by Demonstration (LfD) context. We tried to demonstrate potential benefits and challenges involved in using these queries in LfD through a concrete example. We hope that our paper can serve as a list of tools that LfD researchers could choose from in order to improve the efficiency of their learners and provide guidelines on how to adapt these tools to their specific problems.

References

Angluin, D. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

Cakmak, M., Chao, C., and Thomaz, A.L. Designing interactions for robot active learners. *IEEE Transactions on Autonomous Mental Development*, 2(2): 108–118, 2010.

Calinon, S. and Billard, A. Statistical learning by imitation of competing constraints in joint space and task space. *Advanced Robotics*, 23(15):2059–2076, 2009.

Castro, R., Willett, R., and Nowak, R. Faster rates in regression via active learning. In *In Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2005.

Chernova, S. and Veloso, M. Confidence-based policy learning from demonstration using gaussian mixture models. In *Proc. of Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2007.

Cohn, D., Ghahramani, Z., and Jordan., M. Active learning with statistical models. In Tesauro, G., Touretzky, D., and Alspector, J. (eds.), *Advances in Neural Information Processing*, volume 7. Morgan Kaufmann, 1995.

Druck, G., Settles, B., and McCallum, A. Active learning by labeling features. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 81–90, 2009.

Gribovskaya, E., d'Halluin, F., and Billard, A. An active learning interface for bootstrapping robots generalization abilities in learning from demonstration. In *RSS Workshop Towards Closing the Loop: Active Learning for Robotics*, 2010.

Gribovskaya, E., Khansari-Zadeh, S.M., and Billard, A. Learning nonlinear multi-variate motion dynamics for real-time position and orientation control of robotic manipulators. *International Journal of Robotics Research*, 30:80–117, 2011.

Lomasky, R., Brodley, C., Aernecke, M., Walt, D., and Friedl, M. Active class selection. In *Machine Learning: ECML 2007*, volume 4701 of *Lecture Notes in Computer Science*, pp. 640–647. Springer-Verlag, 2007.

Lopes, M. and Oudeyer., P. Active learning and intrinsically motivated exploration in robots: Advances and challenges (guest editorial). *IEEE Transactions on Autonomous Mental Development*, 2, June 2010.

Lopes, M., Melo, F., and Montesano, L. Active learning for reward estimation in inverse reinforcement learning. In *European Conference on Machine Learning (ECML/PKDD)*, 2009.

Martinez-Cantin, R., Peters, J., and Krause, A. (eds.). *RSS Workshop Towards Closing the Loop: Active Learning for Robotics*. 2010.

Muhlig, M., Gienger, M., Hellbach, S., Steil, J.J., and Goerick, C. Automatic selection of task spaces for imitation learning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4996–5002, 2009.

Raghavan, H., Madani, O., and Jones, R. Active learning with feedback on features and instances. *Journal of Machine Learning Research*, 7:1655–1686, 2006.

Rosenthal, S., Dey, A.K., and Veloso, M. How robots' questions affect the accuracy of the human responses. In *In Proceedings of the IEEE Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2009.

Settles, Burr. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2010.

Vijayanarasimhan, S. and Grauman, K. Whats it going to cost you? predicting effort vs. informativeness for multi-label image annotations. In *In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.