Active Learning For Outdoor Obstacle Detection

Cristian Dima and Martial Hebert
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Email: [cdima,hebert]@ri.cmu.edu

***Abstract*— Real-world applications of mobile robotics call for increased autonomy, requiring reliable perception systems. Since manually tuned perception algorithms are difficult to adapt to new operating environments, systems based on supervised learning are necessary for future progress in autonomous navigation.**

Data labeling is a major concern when supervised learning is applied to the large-scale problems occuring in realistic robotics applications. We believe that algorithms for automatically selecting important data for labeling are necessary, and propose to employ active learning techniques to reduce the amount of labeling required to learn from a data set.

In this paper we show that several standard active learning algorithms can be adapted to meet specific constraints characteristic to our domain, such as the need to learn from data with severely unbalanced class priors. We validate the solutions we propose by extensive experimentation on multiple realistic data sets captured with a robotic vehicle. Based on our results for the task of obstacle detection, we conclude that active learning techniques are applicable to our domain, and they can lead to significant reductions in the labeling effort required to use supervised learning in outdoor perception.

I. Introduction

Outdoor mobile robotics has made remarkable progress towards becoming the preferred solution for many tasks considered too dangerous or too repetitive for humans. The successful transfer of certain robotic technologies from research laboratories to real-world applications has generated a strong interest in developing more advanced robots, that are robust and provide increased autonomy and intelligence.

Reliable perception capabilities are a key requirement for achieving these ambitious goals. While several groups have demonstrated autonomous navigation and perception capabilities using hand-tuned systems (see [1]–[5]), we believe that this approach has a fundamental limitation: since truly general perception systems are yet to be developed, these systems need to be tuned to their operating environment in order to achieve good performance. Since they tend to be complex, a large number of parameters need to be tuned *manually* each time the perception systems need to be adapted to a new environment. This is a slow process, and key application domains for mobile robotics require frequent such reconfigurations.

Learning techniques hold the promise of a practical solution to the tuning problem: if training data consisting of desired input/output pairs is available, many standard learning algorithms can be used to automatically tune the parameters of a model so that it agrees with the training data. Thus, the problem of manual tuning can be transformed in the problem of labeling data, i.e. providing the desired outputs for training purposes.

Unfortunately, labeling data is itself challenging: the data sets used in realistic outdoor perception scenarios are often large, and manual labeling is a slow and expensive process. Before supervised learning becomes practical for our domain, effective methods must be conceived in order to reduce the effort involved in labeling large data sets.

One can think of two such methods: making labeling easy or trying to label only a subset of the available data. As an example of the first approach, the authors of [6] described a learning system for ground height estimation, in which large amounts of labeled data were generated by simply driving a vehicle over interesting terrain. Their solution is highly practical because driving a vehicle represents an easy method of labeling data, and because the system can benefit from adapting its parameters online. Pomerleau [7] proposed a different approach in the context of road following: learn a mapping from images to steering angles by observing a human expert drive a vehicle. This type of approaches work well, but do not generalize to many of the classification problems in our domain.

A more general idea, proposed and demonstrated by Ollis [8], is to use a Bayes classifier and to estimate the posterior of the "non-obstacle" class by using training data to model $P(\text{data}|\text{non-obstacle})$ and $P(\text{data})$. This approach can avoid the challenging task of explicitly modeling the "obstacle" class, but is sensitive to how $P(\text{data})$ is estimated, and is likely to fail without warning in case there is significant class overlap. The idea of using contact sensors to automatically detect when a collision occured (and maybe use that signal for reinforcement learning) is also not general, since collisions with obstacles are only tolerable in a reduced set of situations.

Since there is no generally applicable method to make labeling easy, the remaining alternative is to only label a subset of the data, selected either by random sampling or by identifying the important examples to label based on the distribution of the unlabeled data.

Although very attractive because of its simplicity, using random sampling to reduce the amount of data labeled is dangerous in our domain. Several of the classification problems of interest (such as obstacle detection) are characterized by an extreme unbalance between the priors for the different classes. For example, in many data sets collected by driving a vehicle over interesting terrain, the vast majority of the

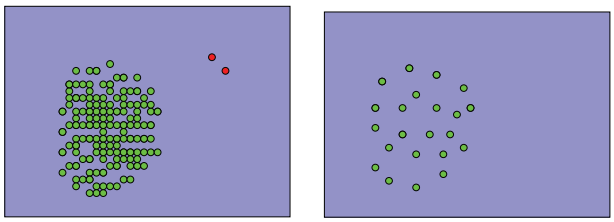

Fig. 1. When learning from data sets with extremely unbalanced class priors, reducing the amount of data by random sampling can result in failing to get any examples from the rare class.

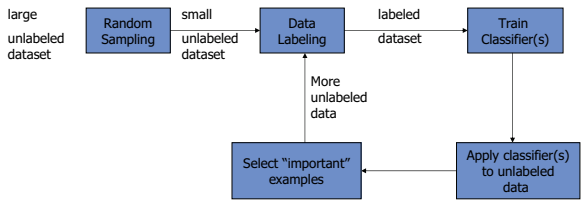

Fig. 2. Typical active learning algorithms are iterative, alternating between training classifier(s) on the available data, identifying "interesting" point(s) and labeling them.

data contains relatively non-interesting examples from the "traversable" class (i.e. different types of roads, grass, etc.), and only few interesting patterns corresponding to obstacles. Applying random sampling in such cases can result in the complete elimination of the rare examples corresponding to obstacles, which can have catastrophic consequences. Such a situation is depicted in Figure 1.

As a solution to these problems, this paper focuses on reducing the amount of labeling needed to achieve good generalization performance by taking into account the distribution of the unlabeled data. We show that several standard active learning algorithms can be adapted to meet specific constraints that are characteristic to our domain, such as the need to learn from data with severely unbalanced class priors. The solutions we propose are validated by experimentation on several data sets captured with a robotic vehicle in representative environments.

II. A Brief Introduction to Active Learning

Active learning techniques are designed for the situation in which some amount of unlabeled data is available, and we are interested in obtaining good performance while labeling only the minimal amount of data necessary. The algorithms typically have access to the unlabeled data, and have the freedom of choosing which queries to make to an expert who can label data.

Active learning presents three main paradigms: *constructive query*, *query filtering* and *pool-based* approaches. In the constructive approach (see [9], [10]) algorithms can generate new query points that are most beneficial to the learner. In contrast, query filtering and the pool-based approaches can only select a query point from an already existent set. While the filtering approach assumes that new examples arrive as a random stream and the decision whether to query or not is made for each point individually, in the pool-based methods the entire data set can be analyzed before choosing the next query point. Since the pool-based methods can use more information, they are generally expected to lead to faster learning than the filtering approaches.

The constructive approach cannot be applied in domains where it is hard to develop a good model for the generative process of the data: the active learning algorithm can generate queries that do not make sense to the expert (see [11]). Given the nature of our data, we cannot use the constructive approach and thus we are limited to using pool-based active learning.

As shown in Figure 2, most active learning techniques are iterative, alternating between choosing a new query point based on a measure of interest and training one or more classifiers on the new data set obtained after completing the query. In cases where the computational complexity of training the classifier(s) is high and a large number of query points is needed, it is common to select a *batch* of query points in a single step, based on the same classifier parameters. In Section III we will discuss an interesting constraint specific to our application domain: we need to select batches of query points not due to computational constraints, but because in our domain it is only practical to label entire *blocks* of data points at each iteration.

The main difference between different active learning algorithms consists in how the importance of an unlabeled example is estimated. In the remaining part of this section, we introduce and discuss several methods to compute interest scores for unlabeled data.

A. Query-by-Committee

One of the most influential results in active learning was the development of the Query-By-Committee (QBC) algorithm [12], which inspired many of the algorithms we discuss in this paper. Query-by-Bagging is centered around the idea of reducing the size of the *version space* [13], the set of all concepts that are consistent with the labeled data available. The version space is a representation of all the information contained in the examples observed by the learning algorithm, and the rate at which the size of the version space decreases is considered a good measure of the progress of the learning process. Observing additional training data can only reduce the size of the set of concepts that agree with the data, and the goal of the QBC algorithm is to identify those unlabeled data points that, when labeled, have a high probability of eliminating a large portion of the version space.

In [12], the authors demonstrate that for a two-class problem, the reduction in the size of the version space can be maximized by making queries for those unlabeled examples whose predicted class is least constrained given the current set of hypotheses that are consistent with the training data. More precisely, if an unlabeled example x has probabilities p and $1-p$ of belonging to the two classes, where p is estimated over *all* the hypotheses consistent with the training data, it can be shown that the information gain –the expected reduction in the size of the version space– is given by

$$\mathcal{H}(p) = -p \log p - (1-p) \log(1-p)$$

which is the Shannon information content (the entropy) of a binary random variable whose probability of being 1 is p. The entropy is not estimated over the entire version space, but by sampling hypotheses from it. These hypotheses form a committee which predicts the label of unlabeled data points, and the QBC algorithm attempts to maximize the information gain by selecting for labeling those points where there is disagreement in the committee.

The authors are able to prove that for certain classes of learning problems, QBC guarantees an exponential decrease of the prediction error as a function of the number of queries. The proof is relatively complex, but at a high level it involves two steps: proving that having a lower bound on the information gain of the queries does guarantee a fast decrease in the prediction error, and proving that for a restricted family of parametrized concept classes, the queries made by QBC have an expected information gain that is guaranteed to be higher than a constant.

While the guarantees of QBC are interesting in themselves, the algorithm has few practical uses in the exact form in which it is presented. For most learning problems in the real world it is impossible to find a hypothesis that is consistent with all the labeled data due to noise; as a result, the version space would be empty. Sampling from the version space is also an issue. In certain cases (see [14]) researchers use generative models and are able to sample from the distributions over their parameters, but this is not a generally applicable solution.

In a slightly more general setting, one could use QBC with randomized algorithms that can reach different hypotheses even when presented with the same training data (e.g. multilayer neural networks initialized randomly). For those who intend to use a deterministic algorithms such as logistic regression, the solution is to randomize the training process using a method such as the Query-by-Bagging algorithm, proposed by Abe and Mamitsuka [15].

B. Query-by-Bagging

Query-by-Bagging (QBBAG) is a combination of QBC with the *bagging* algorithm [16] based on committees of learners trained on resampled training data. Each random subsample used to train committee members has a distribution similar to the initial training set and, after training the bagging committee, its members can be viewed as hypotheses "sampled" from the version space. These hypotheses can be used just like in QBC to identify unlabeled example with high expected information gain by measuring the disagreement between the committee members. Since this time randomization is introduced directly in the training set, QBBAG can be used with both deterministic and randomized classifiers.

A difference between the original formulation of Query-by-Committee and Query-by-Bagging is that the first algorithm was introduced using the query filtering paradigm, while QBBAG is using the more common pool-based approach. Thus, QBBAG is trying to identify the *maximally* informative unlabeled example, which requires a measure of the disagreement among the committee members.

Abe and Mamitsuka [15] use as disagreement metric the difference between the number of committee members that vote for each one of the two labels, which is equivalent to scoring based on the number of committee members that are in minority. A slightly more general version of this metric, which is applicable to classification problems with more than two classes, is the *vote entropy* method (see [14], [17]).

Given that the classifier used for our experiments, logistic regression, provides posterior estimates instead of just class labels, it would be desirable to exploit this additional information. As a result, we have chosen to use Query-by-Bagging with a different committee disagreement metric, the *Kullback-Leibler-divergence-to-the-mean* (see [14], [18]). Its main advantage is that it takes into account the confidence of the classifications made by the committee members when estimating their disagreement.

The KL-divergence-to-the-mean is defined as the average of the KL-divergences between the posterior class distribution of each committee member and the average posterior class distribution of the entire committee[1]. If k is the number of members in the QBBAG committee and x is an unlabeled example, the KL-divergence-to-the-mean is defined as

$$\frac{1}{k} \sum_{m=1}^{k} D(\mathrm{P}_m(C|x) || \mathrm{P}_{avg}(C|x)),$$

where C is a random variable over the classes, $\mathrm{P}_m(C|x)$ is the posterior class distribution of committee member m, and $\mathrm{P}_{avg}(C|x)$ is the average posterior class distribution of the committee. The KL divergence between two distributions $\mathrm{P}_1(C)$ and $\mathrm{P}_2(C)$ is given by

$$D(\mathrm{P}_1(C) || \mathrm{P}_2(C) = \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}_1(c_j) \log \frac{\mathrm{P}_1(c_j)}{\mathrm{P}_2(c_j)}$$

where $\mathcal{C}$ is the set of classes, and $\mathrm{P}(c_j)$ is the probability the class label is j.

To better illustrate the difference between the KL-divergence-to-the-mean and the vote entropy, let us consider the example of a binary $(+, -)$ classification problem in which the output of the classifiers represent the probabilities of the "$+$" label. The vote entropy of an example on which a three-member committee produces outputs $(0.51, 0.51, 0.49)$ is the same as if the outputs were $(0.99, 0.98, 0.01)$, because vote entropy ignores classification confidence information. KL-divergence-to-the-mean would score the second example much higher than the first one, which is intuitively desired from a committee disagreement measure. Preliminary experiments in which we compared the two disagreement metrics confirmed that better performance can be achieved using the metric that takes class posteriors into account.

[1]We will follow here the development from McCallum and Nigam [14]

C. Uncertainty Sampling

Uncertainty Sampling (US) [19] is a heuristic alternative to QBC, and is one of the simplest active learning algorithms we are aware of. The algorithm requires a classifier that can produce reasonable estimates of its prediction confidence. After the classifier is trained on all the labeled data available, Uncertainty Sampling applies it to all the unlabeled examples and selects for querying the one on which the current classifier is least confident.

The motivation for Uncertainty Sampling comes from the desire to avoid having to sample classifiers from the version space. Lewis and Gale propose to approximate the *classifier uncertainty* (the confidence that one of the labels occurs given the current version space) by the *label uncertainty* as estimated by the unique classifier trained on all labeled data.

To understand the difference and why this approximation can be poor, consider a hypothetical classifier that is presented with a binary classification problem and indicates that, given its parameters, the probability of an example x of belonging to the "$+$" class is 0.99. Uncertainty Sampling would consider such an example extremely non-interesting. In reality, it can be the case that the training data does not constrain the parameters of the model very well, meaning that several decision boundaries would agree equally well with the training data. In this case, the true uncertainty in the classification of the example is much higher. If a QBC/QBBAG committee analyzes the same problem with the same base learner, the high uncertainty in the classification boundary makes it likely that some of the hypotheses sampled from the version space disagree on the label of the example, resulting in a query. The problem of having over-confident classifiers is acknowledged by Lewis and Gale in their original paper [19].

Despite the fact that Uncertainty Sampling is probably the best known active learning algorithm, our experiments have clearly shown that its performance is inferior to that of more principled algorithms, such as Query-by-Bagging. As a result, we will not include results obtained with Uncertainty Sampling in our experimental evaluation.

D. Co-Testing

A slightly more interesting algorithm is Co-Testing [20], which borrows the idea of using *redundant views* from Co-Training (a semi-supervised learning algorithm proposed by Blum and Mitchell in [21]) and adapts it to the active learning domain. According to the authors, "a domain has redundant views if there are at least two mutually exclusive sets of features that can be used to learn the target concept". Since in our domain one often uses different sensing modalities for perception, we have considered the features extracted from the different sensors to be our views. Thus, in our experiments we will classify obstacles based on "color", "texture" or "laser" views, for example.

The algorithm works by training classifiers on the different views of the available training data, and classifying the unlabeled data independently in all the views. The set of examples on which the views disagree represents the pool of potential labeling candidates, and different flavors of Co-Testing exist for selecting one query out of this pool. The authors claim that the most efficient gains can be obtained by querying contention points on which the views disagree most strongly, but the experimental results presented were generated by simply choosing randomly from the contention points.

Comparing Co-Testing to Uncertainty Sampling, Muslea et al. [20] claim that by using multiple views their algorithm can be more aggressive: querying contention points on which the algorithms are most confident increases the probability of a large effect in at least one of the views. For comparison to QBC, the authors construct a learning problem which is PAC-learnable but on which QBC fails with high probability while Co-Testing succeeds in very few steps. In the experimental results presented in [20], Co-Testing seems to generally perform quite similarly to Query-by-Bagging.

In our experiments, we used Co-Testing with the selection scheme in which the most contentious unlabeled data points are queried. Similarly to the Query-by-Bagging algorithm, we have chosen the KL-divergence-to-the-mean as a disagreement measure, where the members of the committee are the classifiers trained on the different views of the data.

To conclude this section, we acknowledge the fact that this list of algorithms is by no means exhaustive. Several other principled active learning algorithms (such as the ones proposed by Roy and McCallum in [22] or by Tong and Koller in [23]) were not included in our experimentation, mostly because of their computational requirements. Nevertheless, we would like to remind the reader that the goal of our research is to determine *if* active learning is applicable in our domain, and not to identify the most effective active learning method.

III. Active Learning for Obstacle Detection

A. The Initialization Problem

In the previous section we presented a number of active learning algorithms with the potential of reducing the amount of data labeling required for large-scale applications of supervised learning. Until now, we have glanced over an important aspect: the initialization step required by all the algorithms we presented.

In Figure 2, we indicated that the "seed" labeled data set required to start the active learning iterations is obtained by random sampling. This is indeed the typical initialization method used in the active learning literature, while in other cases data is selected manually. Unfortunately, neither of these options is practical for our domain: manual selection is not applicable to large data sets, and random selection can fail as described in the introduction: when a data set is severely unbalanced, we can obtain initialization sets that contain examples from a single class (see Figure 1).

To address this important problem, we have developed the Unlabeled Data Filtering (UDF) algorithm [24]. The intuition behind UDF is simple: we are interested in reducing the size of an unlabeled data set by discarding redundant examples, while keeping most of the rare patterns. We define "rare" patterns as those data points that correspond to sparse regions of the

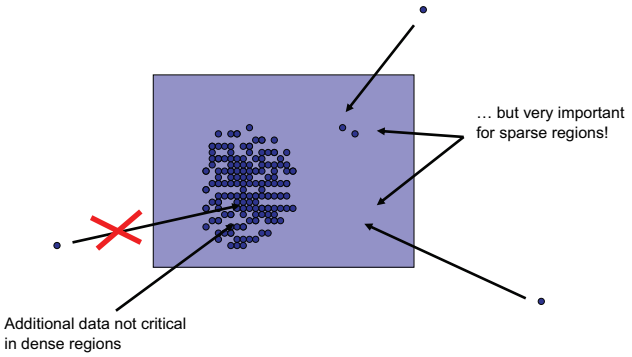

Fig. 3. The essence of the Unlabeled Data Filtering algorithm: when new data is observed, discard examples corresponding to densely populated regions of the feature space while keeping the ones from sparsely populated regions.

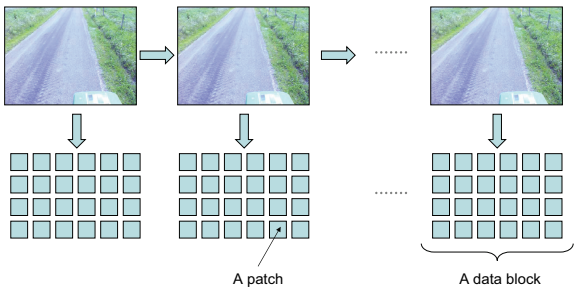

Fig. 4. The Data Block Constraint: our classifiers and active learning algorithms work at the image patch level, while data labeling takes place at the image level. The interest scores assigned to individual patches need to be aggregated to obtain image interest scores.

feature space, while the "redundant" examples are the ones from regions of the feature space that are densely populated.

The essence of the UDF algorithm is illustrated in Figure 3: starting with any random data point, construct a set S of data to be labeled by iteratively selecting data points from the most sparse regions of the feature space. Sparsity is defined using some form of probability density estimation, based on the current set S. At a high level, UDF can be viewed as an algorithm that attempts to select for labeling data points that cover as uniformly as possible the feature space of the classification problem of interest.

The Unlabeled Data Filtering algorithm is closely related to the anomaly detection research area, in which estimating the probability density function (PDF) over some feature space is the method of choice for identifying anomalous patterns. The subtle but essential difference is that UDF does *not* select for labeling only anomalous patterns: since the PDF it uses for estimating sparseness is based on the set S of data points selected for labeling, in the initial stages of the algorithm any data point is equally likely to be selected for labeling, even if it lies at the center of a compact cluster of points in the original unlabeled data set. The difference in the data sets over which the sparseness of the feature space is estimated is what clearly distinguishes the UDF algorithm from the algorithm proposed by Pelleg and Moore in [25], which is the most closely related work we are aware of. For a more detailed discussion, we refer the reader to [26].

The interest measure used by UDF is the negative log-likelihood given by the PDF estimated over S at each unlabeled data point. For the experiments presented in this paper, we estimate the density over the feature space using kernel density estimation with a Gaussian kernel, whose bandwidth is chosen using cross-validation. Since our initial feature space is often high-dimensional, we project our data to a lower dimensional space using Principal Component Analysis.

B. The Data Block Constraint

An additional aspect preventing the direct application of active learning in our domain is what we call the *data block constraint*, illustrated in Figure 4.

This constraint appears naturally in perception problems because we often try to classify as small a region as possible to improve the localization of our detections, while a much larger region needs to be analyzed by a human expert before data can be labeled. For example, we are interested in classifying individual rectangular image patches as obstacles or non-obstacles, but a human needs to be presented with an entire image before being able to indicate the correct label for an image patch. The same is true for 3-D voxels of data, if the classification process takes place in the 3-D space. An additional benefit of presenting experts with entire images is that they can label a much larger number of data points at the same time, for example by drawing contours that enclose all the obstacles in an image.

The data block constraint can be summarized by the fact that our classifiers and active learning algorithms work at the image patch level, while we need interest scores for entire images in order to select the informative ones for labeling. To address this constraint, we need to aggregate the interest scores assigned to individual image patches (the data points) into interest scores assigned to images (blocks of data points).

In [24] we argued that aggregating interest measures over all the patches within an image is undesirable because high interest patches can be overwhelmed by large amounts of uninteresting patches. The other extreme of scoring an image only based on its most interesting patch is also ineffective, because the most unusual patterns in an image will often be outliers corresponding to various artifacts. This problem is also reported in [14], where the authors address the outlier problem by weighting the contributions of the different data points based on local data density.

The solution we chose for balancing between these two undesired behaviors was to average the interest measures of the k highest scoring patches in each image. Experiments we presented in [26] demonstrate that this simple accumulation scheme is robust to the specific value k that is chosen: while our intuition that accumulating over too few or too many patches is detrimental has been been confirmed, there is a wide interval for k (roughly between 8 and 50) for which the performance of the algorithms is almost unchanged.

TABLE I

SOME OF THE DATA SETS USED IN OUR EXPERIMENTS.

Name	# Images	Distance traveled (m)
FmObsCourseNonPoles_01	436	372
FmObsCourseNonPoles_02	558	488
FmObsCourseAll_01	1018	749
FmObsCourseAll_02	673	481
ApMdSpiral	1393	768
FmDriveDown_01	2030	1627
FmDriveDown_02	1191	938
FmVariousObs	1096	543

IV. EXPERIMENTAL EVALUATION

A. Data sets, Feature Sets and Learners

The experiments presented in this paper are based on data collected with a robotic vehicle on a farm and in a meadow with tall vegetation. The data sets on which we performed most of our experiments are listed in Table I, along with the number of images recorded and the distance traveled. To be able to perform the randomized runs necessary to obtain confidence bounds for the various performance estimates, we have *exhaustively* labeled all the data sets in Table I.

The obstacles in our data sets are either natural features such as trees, fences and buildings, or objects that we placed in the environment, such as green and gray flower pots hidden in vegetation of various heights, thin and thick pipes, several pieces of wood and a small cart. The pairs of data sets whose names are not separated by a horizontal line in the table contain data that is similar in nature, and thus can be used as train/test sets. For two of our data sets we did not have additional test sets. This is however not crucial for the kind of experiments we perform: we measure learning efficiency and not generalization performance.

Our vehicle is equipped with visible and infrared cameras whose positions relative to our laser range finder units is precisely known. As a result, for each image patch we can extract a multi-modal feature vector containing color (COL, 5 features), texture (TEX, 24 features), range (LASER, 4 features) and infrared (IR, 2 features) information (see [26] for more details).

While one would normally use the all the available sensing modalities for the classification problem, for these experiments we chose to test our algorithms on different subsets of the feature vector. The various sensing modalities have different characteristics, which means that for each data set we can test the active learning algorithms on a variety learning problems and better assess the generality of our conclusions.

For all the experiments we present, we used logistic regression as a base classifier. Logistic regression has many desirable properties, such as fast training and testing, and guaranteed convergence to a global minimum [27].

B. Performance Metric

In the active learning literature performance is typically measured by comparing the performance achieved after different numbers of queries with the maximum information performance, obtained when the learner has access to all the labeled data in the training set.

We used the Area Under the ROC Curve (AUC) as our performance metric. Many properties of the AUC make it preferable to test error rates or precision/recall break-even points (see [28] for an excellent discussion): it eliminates the need to set a threshold on the class posteriors, it is robust to unbalanced class priors and has a precise probabilistic interpretation.

Exploiting our exhaustively labeled data sets, we performed at least 10 randomized runs for each experiment, and we are displaying point-wise confidence intervals of $+/-1.64\sigma$ for all plots. According to bar-overlaps-bar test described in [29], this width of the confidence intervals guarantees the 95% significance of the ordering of *individual runs*, provided that the intervals do not overlap.

C. Experiments and Results

Our experiments were designed to answer two main questions: we wanted to know if active learning can reduce labeling requirements compared to random sampling, and to verify that UDF is an effective initialization method.

Since the solutions we propose are meant for use in real-world systems, adjusting the parameters of our algorithms for the different data or feature sets is unacceptable. All the experiments we present in this paper and in [26] are performed without *any* parameter re-tuning. The initial settings were chosen based on experiments presented in [26], analyzing their influence on our algorithms. In particular, we used QBBAG committees of 15 classifiers, we compressed our initial data to a two-dimensional space using PCA for UDF, and we accumulated the interest scores over the 8 highest scoring patches in each image. These parameters have relatively wide ranges of reasonable settings.

To answer the first question concerning the benefits of using active learning methods in our domain, we have compared the efficiency of QBBAG and UDF to manual and random selection. To perform randomized manual selection experiments, a human expert created for each data set a list of images with close-up views of all the obstacles. At run-time, the MANUAL selection scheme could sample without replacement from these lists of "interesting" images. In all of our plots we included for reference a horizontal line corresponding to the performance achieved by the MAX INFO classifier, which is trained using the labels of all the data in the training set. Ideally, we would want our automatic data selection schemes to get close to the MAX INFO performance after very few image queries.

To study the effect of the initialization schemes on active learning, we compared the performance of the QBBAG algorithm using RANDOM, MANUAL, and UDF initialization. For each initialization method, we selected 5 images which were then used as "seed" data set for QBBAG.

Due to space constraints, the results presented in Figure 5 are only a small subset of all the experiments we performed; additional results are presented in more detail in [26]. Nevertheless, Figure 5 is representative for the type of

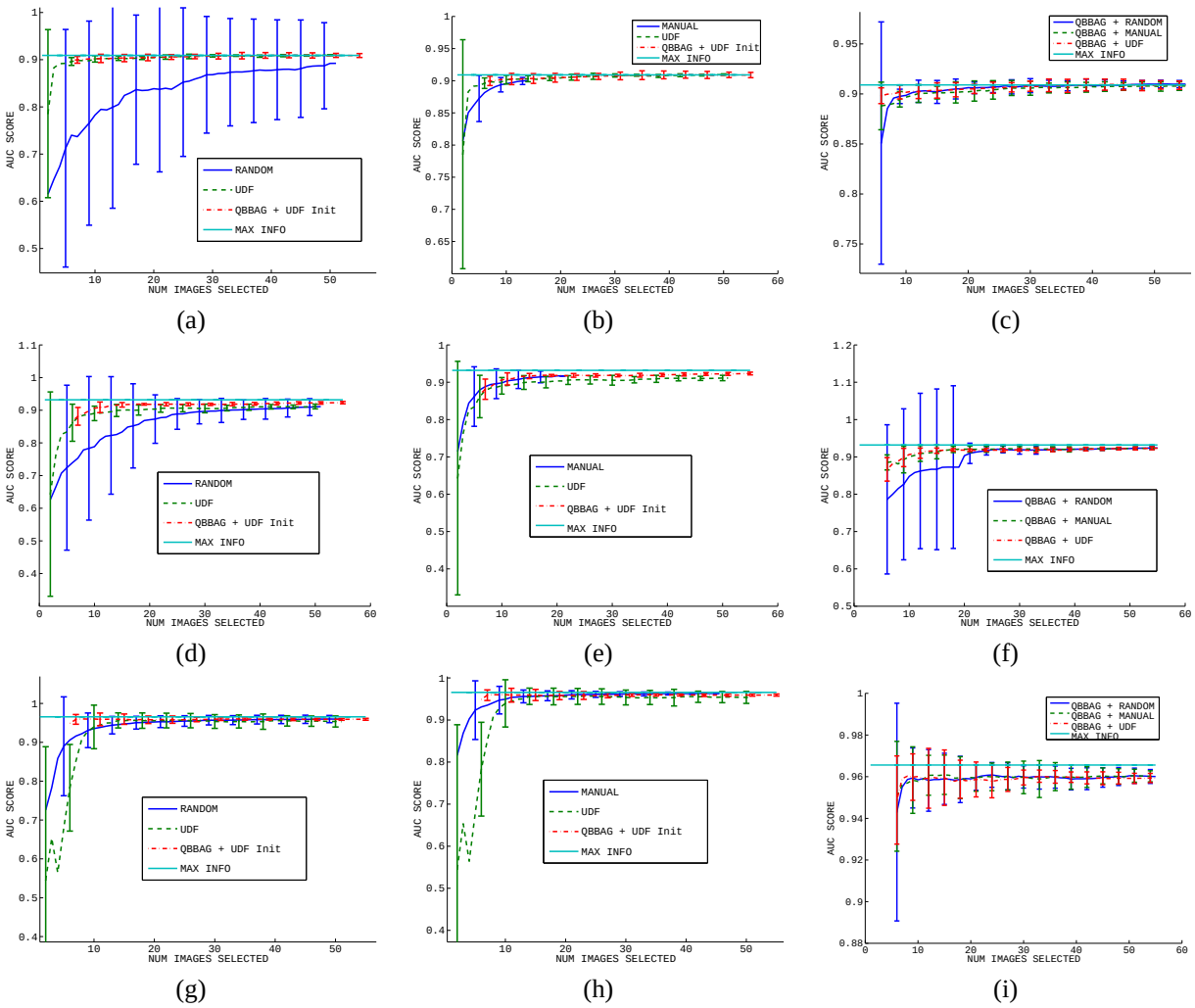

Fig. 5. Results on FmObsCourseNonPoles COL+TEX (a-c) , ApMdSpiral COL+TEX+LASER (d-f), and FmDriveDown COL+TEX+LASER (g-i). We are comparing the performance of RANDOM vs. Active Learning (first column), MANUAL vs. Active Learning (second column) and the RANDOM, MANUAL and UDF initialization methods (third column).

results we obtained. The plots in the left column compare the performance of RANDOM selection to QBBAG with UDF initialization and to the UDF algorithm used as a stand-alone data selection method. The middle column compares MANUAL selection to the same active learning methods. The right column displays the performance achieved by QBBAG when initialized randomly, manually, and using UDF.

The results obtained on the data sets represented in the first two rows are very good. The data presents strongly skewed class priors, and the phenomenon we described in Figure 1 occurs naturally: RANDOM selection is likely to query for the labels of several non-obstacle images before an obstacle is observed. As a result, the average performance is significantly lower than that of both QBBAG and UDF, which quickly get close to the MAX INFO performance. The standard deviation for RANDOM selection is also much higher for these data sets, because in many runs no obstacles are observed during the first selection iterations, resulting in low test AUC scores. Both active learning algorithms perform similarly to the MANUAL selection method: their average performance is essentially the same, while the standard deviations are slightly smaller. The results in the third column show that UDF is indeed effective at initializing active learning methods: for the first few iterations QBBAG has roughly the same scores and confidence bounds when initialized with either UDF or MANUAL, and both versions are better than QBBAG with RANDOM initialization. The difference is most significant for the ApMdSpiral data set, which is the most skewed.

The performance of active learning on the more balanced FmDriveDown data set was quite different: QBBAG resulted in an improvement over RANDOM selection, but UDF performed slightly worse. This is not unexpected: this data set presents some class overlap, and given the nature of the obstacles, a random sample of the data has a good chance to contain most of the relevant data for obstacle detection. This is confirmed by the relatively small performance difference between MANUAL and RANDOM selection on this data set. When used for QBBAG initialization, UDF resulted in a very

small improvement over RANDOM selection.

Although we are not presenting results here, our experiments with Co-Testing indicate that it performs comparably to QBBAG when the color, texture and laser features are used as three "redundant" views of the data. This seems to confirm the results presented by Muslea et al. in [20].

V. Discussion

Our experiments indicate that active learning has an important role to play in making supervised learning applicable to large-scale robotics applications. Especially when class priors are unbalanced, which happens naturally in our domain, active learning techniques can lead to significantly reduced labeling requirements compared to random data selection. The learning efficiency of our algorithms is comparable to that of a human expert, and they can be applied to large data sets where manual data selection is impractical.

Automatically selecting data for labeling has an additional benefit: it makes training a perception system accessible to end-users with minimal expertise, which opens the door to a new set of robotics applications.

The fact that we obtained good experimental results on several combinations of data sets and feature sets without retuning *any* parameters indicates that our algorithms are robust and practical. Furthermore, in [26] we present qualitative results showing that they scale up to data sets of close to $50,000$ images.

As future work, we intend to perform additional large-scale experiments, and to explore robust algorithms for imposing spatial coherency constraints among the patches used to estimate the interest score of an image.

Acknowledgment

This paper was prepared through collaborative participation in the Robotics Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0012. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

References

[1] Mark Ollis and Todd Jochem, "Structural method for obstacle detection and terrain classification," 2003, vol. 5083, pp. 1–12, SPIE.

[2] A. Stentz, A. Kelly, P. Rander, H. Herman, O. Amidi, R. Mandelbaum, G. Salgian, and J. Pedersen, "Real-time, multi-perspective perception for unmanned ground vehicles," in *AUVSI*, 2003.

[3] Mark Rosenblum and Benny Gothard, "A high fidelity multi-sensor scene understanding system for autonomous navigation," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, October 2000, pp. 637–643.

[4] Carl D. Crane III, David G. Armstrong II, Maryum Ahmed, Sanjay Solanki, Donald MacArthur, Erica Zawodny, Sarah Gray, Thomas Petroff, Mike Grifis, and Carl Evans, "Development of an integrated sensor system for obstacle detection and terrain evaluation for application to unmanned ground vehicles," 2005, vol. 5804, pp. 156–165, SPIE.

[5] Brian Yamauchi, "The Wayfarer modular navigation payload for intelligent robot infrastructure," 2005, vol. 5804, pp. 85–96, SPIE.

[6] Carl Wellington and Anthony Stentz, "Online adaptive rough-terrain navigation in vegetation," in *Proceedings of the Int. Conf. on Robotics and Automation (ICRA 04)*, April 2004.

[7] Dean Pomerleau, "Progress in neural network-based vision for autonoumous robot driving," in *Proceedings of the Intelligent Vehicles '92 Symposium*, 1992, pp. 391–396.

[8] Mark Ollis, "Bayesian learning with ladar and radar sensors," Presented at the DARPA PerceptOR Workshop in Arlington, VA, January 2004.

[9] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan, "Active learning with statistical models," in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, Eds. 1995, vol. 7, pp. 705–712, The MIT Press.

[10] E.B. Baum, "Neural net algorithms that learn in polynomial time from examples and queries," in *IEEE Transactions on Neural Networks*, January 1991, vol. 2, pp. 5–19.

[11] E.B. Baum and K. Lang, "Query learning can work poorly when human oracle is used," in *International Joint Conference in Neural Networks*, 1992.

[12] Yoav Freund, H.Sebastian Seung, Eli Shamir, and Naftali Tishby, "Selective sampling using the query by committee algorithm," *Machine Learning*, vol. 28, pp. 133–168, 1997.

[13] Tom M. Mitchell, "Generalization as search," *Artificial Intelligence*, vol. 18, no. 2, 1982.

[14] Andrew McCallum and Kamal Nigam, "Employing EM and pool-based active learning for text classification," in *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp. 359–367.

[15] Naoki Abe and Hiroshi Mamitsuka, "Query learning strategies using boosting and bagging," in *Proceedings of the Fifteenth International Conference on Machine Learning*. July 24-27 1998, pp. 1–10, Morgan Kaufmann Publishers.

[16] Leo Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[17] Ido Dagan and Sean P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *International Conference on Machine Learning*, 1995, pp. 150–157.

[18] Fernando C. N. Pereira, Naftali Tishby, and Lillian Lee, "Distributional clustering of English words," in *Meeting of the Association for Computational Linguistics*, 1993, pp. 183–190.

[19] David D. Lewis and William A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual ACM SIGIR conference on Research and development in information retrieval*. 1994, pp. 3–12, Springer-Verlag New York, Inc.

[20] Ion Muslea, Steven Minton, and Craig A. Knoblock, "Selective sampling with redundant views," in *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI)*, Austin, Texas, July 30 - August 3 2000, pp. 621–626.

[21] Avrim Blum and Tom Mitchell, "Combining labeled and unlabeled data with co-training," in *COLT: Proceedings of the Workshop on Computational Learning Theory*. 1998, pp. 92–100, Morgan Kaufmann, San Francisco, CA.

[22] Nicholas Roy and Andrew McCallum, "Toward optimal active learning through Monte Carlo estimation of error reduction," in *Proceedings of the International Conference on Machine Learning*, June 2001.

[23] Simon Tong and Daphne Koller, "Support vector machine active learning with applications to text classification," in *Proceedings of the Seventeenth International Conference on Machine Learning*, July 2000.

[24] Cristian Dima, Martial Hebert, and Anthony Stentz, "Enabling learning from large datasets: Applying active learning to mobile robotics," in *Proceedings of the International Conference on Robotics and Automation*. April 26 - May 1 2004, vol. 1, pp. 108 – 114, IEEE.

[25] Dan Pelleg and Andrew Moore, "Active learning for anomaly and rare-category detection," in *Advances in Neural Information Processing Systems 18*, December 2004.

[26] Cristian Dima, *Active Learning for Outdoor Perception*, Ph.D. thesis, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, June 2005.

[27] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics. Springer-Verlag, 2001.

[28] Tom Fawcett, "ROC graphs: Notes and practical considerations for researchers," Tech. Rep., HP Laboratories, Palo Alto, CA, March 2004.

[29] Tom Minka, "Judging significance from error bars," http://research.microsoft.com/ minka/papers/, November 2002.