Domain Adaptation For Mobile Robot Navigation

David M. Bradley, J. Andrew Bagnell
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15217
dbradley, dbagnell@rec.ri.cmu.edu

1 Introduction

An important challenge in outdoor mobile robotic perception is maintaining terrain classification performance throughout the extremely variable conditions that we may wish a robot to operate under. Outdoor robots operate in a series of "environments" that consist of diverse terrain, vegetation, weather, and lighting conditions. A physical robot does not randomly jump between environments; typically it will operate for long stretches of time in one particular environment, making it advantageous to adapt the robot's performance to its current environment.

We apply and adapt recent advances in learning from multiple sources [1, 2] to the mobile robot navigation problem. Specifically, we examine a terrain classification problem that is an important stage in many mobile robot navigation systems, including the UPI/Crusher autonomy system we have developed. Our extensive field experiments demonstrate large differences in the appearance of terrain to cameras and Lidar between environments, making it necessary to treat each environment as a separate data source. We demonstrate significant improvements in classification and the resulting navigation performance by compensating for *covariate shift* between the environments where training data was collected and the current test environment using unlabeled data from the robot's current locale.

2 Terrain Classification Application

The "Crusher" robot is a capable platform for research in off-road autonomous driving. The high-level task that the robot is designed to perform is driving from its current location to an arbitrary goal location defined by GPS coordinates (Figure 2). The task of the perception system is to predict a safe and efficient path (shown as the green dotted line) from the current location of the robot to the goal, using the data from a suite of onboard cameras and laser scanners to avoid obstacles. This task is accomplished by first classifying each 3-D voxel of space around the robot into "ground", "vegetation", or "obstacle" terrain classes, and then computing a *traversal cost* for each 2-D cell as a function of the classification of the voxels contained in each column of the map as well as the local shape of the underlying ground supporting surface.

The "Crusher" robot has operated in many different terrain types and seasonal conditions. In each environment, sensor data has been collected and labeled to make sure that the perception system is operating correctly. As shown by the randomly selected training set images in Figure 1, over the life of the program this has resulted in data from a diverse set of terrain, seasons and weather/lighting conditions. After the robot had operated in a few different environments, it was observed that training the voxel classifier on labeled data collected from certain environments was hurting performance on other environments, and it became necessary to selectively drop some training data from the training set in order to improve voxel classification accuracy on the environment in which the robot was currently operating.

Table 1 displays the difference in voxel classification accuracy between using a labeled training set drawn from the other environments the robot has operated in and one drawn from the current environment the robot is operating in. It shows that in many cases the performance of the voxel classifier on a particular environment is significantly handicapped by training on labeled data from the other environments.

Figure 1: Randomly-sampled images from the the labeled voxel data-set collected for the UPI program reflect the wide variety in environments in which the Crusher robot has operated. Different regions of the country, and even different times of day, can produce substantially different distributions within voxels. As shown in Table 1, minimizing training error over all the environments is not as effective as training only on environments similar to the one the robot is currently operating in.

Environment				Training Set Drawn From:	
Name	Location	Season	Terrain	Others	Current
Taylor	PA	Winter & Snow	Woods & Grassland	71.8%	92.0%
Gascola	PA	Summer	Woods & Grassland	70.3%	94.3%
Ft. Bliss	TX	Winter	Scrubland	74.1%	98.5%
Sommerset	PA	Spring	Woods & Grassland	93.0%	95.5%
Ft. Drum	NY	Summer	Woods & Grassland	97.6%	98.7%
Gascola	PA	Winter	Woods & Grassland	94.7%	95.6%
Willow St.	PA	Fall	Urban	88.3%	98.9%

Table 1: It is possible to dramatically improve the accuracy of the voxel classifier component of Crusher's perception system by training its parameters only on data that is relevant to the environment the robot is currently operating in. This table compares classification accuracy in several different environments when the voxel classifier training set is drawn from the current environment with the accuracy when the training set is drawn from other environments.

The difference between environments can also be seen directly in terms of the voxel feature vectors. Figure 2 shows the result of projecting the feature vector associated with each labeled voxel into the most discriminative 2-D subspace, as computed by Linear Discriminants Analysis (LDA) to separate "road", "vegetation" and "obstacle" classes. Each plot shows the labeled voxel data from a particular environment as points colored by class. The decision boundaries of the optimal maxent classifier for each environment are shown as black lines. These optimal decision boundaries can change substantially between environments.

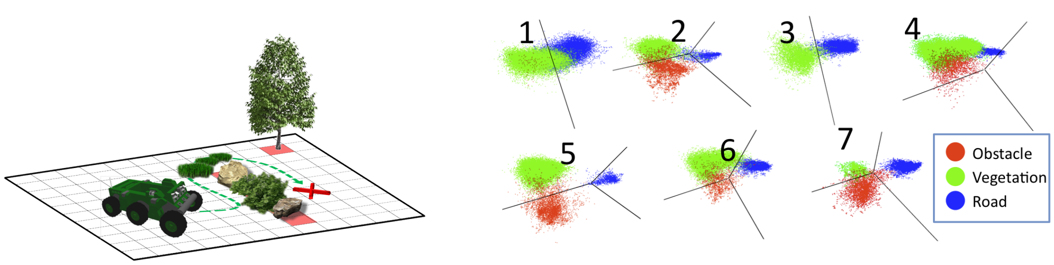

Figure 2: (Left) Mobile robot Navigation problem. The "Crusher" robot uses a variety of onboard sensors to plan a safe path (green line) to a goal location through hazardous off-road environments (red X). (Right) Different environments produce voxel class distributions, with substantially different optimal classification boundaries. Shown is the distribution of labeled data for 7 environments in the training set that had a good mix of "rigid" (road & obstacle) and "non-rigid" (vegetation) voxels. Labeled voxels are ploted as colored points in the most discriminative 2-D subspace (as computed by multi-class LDA across the entire training set). The optimal classification boundaries in each case are denoted by black lines.

2.1 Automatic Adaptation From Unlabeled Data

The environmental variability faced in problems such as outdoor autonomous driving can be mitigated somewhat by taking advantage of the stream of unlabeled data the robot receives from its current environment. With certain assumptions, this unlabeled data can be used to allow the robot to automatically adapt its training set to match a test environment without requiring human intervention to label additional examples or decide which examples should be dropped from the training set. The unlabeled examples collected continuously by the robot are automatically used to estimate an importance weight for each example in the training set that will cause the training set to better approximate the test environment. The classifier is then retrained to minimize error over this re-weighted version of the training set. This approach is an example of importance sampling and is often referred to as adapting to "covariate shift" [1], and bears similarities to methods such as [3] which use unlabeled data to train a simple classifier that is locally accurate in a complex, non-linear space. The joint distribution in the labeled test set, $P^*(Y, X, \text{test})$ is effectively approximated with an importance weighted version of the training set, where the weights are chosen by the ratio of generative density models of $P(X)$. See [4, 1] for details on this procedure as well as the assumptions necessary for its validity.

2.2 Training Classifiers Instead of Estimating Densities

An "importance weight", $w_i = \frac{P(\vec{x}_i|\text{test})}{P(\vec{x}_i|\text{train})}$, for each example can be calculated by estimating the density of unlabeled samples from the training and test environments. However, density estimation is difficult to do accurately in high dimensional spaces, and estimating two high-dimensional distributions in order to compute the scalar weight for each data point is an unnecessarily difficult estimation problem. An alternative approach introduced in [2] instead estimates a single conditional distribution, $p(\text{train}|X)$, with a probabilistic classifier such as logistic regression. Effectively, instead of estimating the probability density function of feature vectors in the training and testing sets independently, we can train a logistic regression classifier to predict if a particular feature vector came from the test or the training set.

3 Experiments

Terrain classification forms a key component of Crusher's perception system. In this task features computed from camera and laser data of a local region of 3-D space (voxel) are classified into a rigid or a non-rigid terrain type. For certain parts of the system the rigid terrain type is subdivided into "road" voxels believed to be the ground surface and "obstacle" voxels believed to be above the ground surface. During the course of the UPI program, a large dataset of labeled voxels was collected from a variety of different terrain and seasonal conditions. From this data-set seven environments were defined that each had a mix of labeled rigid and non-rigid voxels. For each adaptation experiments, one of the seven environments was used as a test set and the other N-1 environments were used to form a training set. For computational efficiency the test environment was randomly subsampled to 40,000 voxels and the training environment was subsampled to 360,000 voxels. Equally sized unlabeled data samples were also collected by randomly sampling from the sensor logs that the labeled samples were drawn from.

3.1 Voxel Classification Results

The classification-based reweighting technique provided a performance boost for most of the environments tested. The results on each of the seven environments are shown in Table 2, and graphically in Figure 3. In all environments except the "Willow Street" test site, adaptation with the classifier-based method improved performance. Estimating the probability densities with Gaussian Mixture Models (GMMs) was not as effective on average, and fitting the GMMs proved to be more computationally intensive, and less beneficial than training logistic regression classifiers to estimate the data weights.

As the goal of domain adaptation is to approximate the test set with a weighted version of the training set, Table 2 also lists the "optimal" classification results of directly training on the test set. These results were included to show that due to the limited capacity of the linear classifier used for these experiments and the significant amount of label noise that exists in the training set (due to effects like vehicle pose error and labeling mistakes), it is not possible to achieve 100% classification accuracy even by training directly on the test set. Over all environments, the Logistic Regression (LR) classifier based algorithm achieved an average of 20% of the gap between the training set and the optimal results. The GMM based algorithm improved on some environments but was harmful on others.

Environment	Original	GMM	LR	Optimal
Taylor Winter Snow	72.0%	69.5%	82.9%	92.1%
Gascola Summer	70.4%	70.2%	74.0%	94.5%
Bliss Winter	74.3%	75.2%	76.5%	98.5%
Sommerset Spring	92.9%	92.9%	93.4%	95.5%
Drum summer	97.6%	97.7%	98.0%	98.7%
Gascola Winter	94.6%	94.6%	94.9%	95.5%
Willow Street	88.2%	89.1%	87.3%	98.9%
Mean	84.3%	84.2%	86.7%	96.2%

Table 2: Adaptation Performance on each environment. Environments which are poorly matched to the full training set show the greatest gains. The left column is the test set performance of classifiers trained on the full training set. Estimating importance weights with logistic regression (LR) proved better on most environments than GMM-based density estimation. In all cases except environment 7, adaptation from unlabeled data using the classifier-based method improved classification performance. The right column shows the result of training directly on the test set for each environment, and provides an upper bound on the performance of any domain adaptation algorithm. The regularization parameter was selected by leave-one-out cross validation.

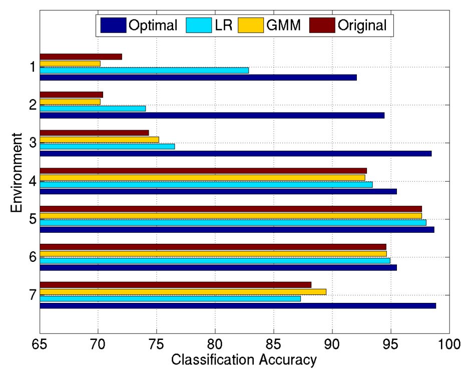

Figure 3: Performance on most environments benefitted from reweighting. The top bar in each group is the performance of the classifier trained on the original training set and tested on this environment. The middle bars shows the performance after adaptation with both approaches to importance weight estimation. Finally, the bottom bar shows the "optimal" performance possible if the classifier were trained on the actual test set.

3.2 System-level Improvement

A preliminary experiment was also conducted on the impact of this algorithm on final-system performance, which showed promising performance improvements. The robot traversed a 1.5 Km course twice, once with classifiers trained from the original labeled training set, and again with classifiers trained from an adapted training set. The adaptation algorithm made the robot more willing to call dead November vegetation non-rigid, and led to faster speeds and a more efficient route which shaved 13% off of the total run. However due to limited robot availability further system-level experiments were not possible.

Acknowledgments

Funding for this work was provided by the Advanced Research Projects Agency (DARPA) under contract Unmanned Ground Combat Vehicle PerceptOR Integration (UPI, contract number MDA972- 01-9-0005) and the U.S. Army Research Laboratory, under contract Robotics Collaborative Technology Alliance (contract number DAAD19-01-2-0012). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies or endorsements of the the U.S. Government.

References

[1] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, 2000.

[2] S. Bickel, C. Sawade, and T. Scheffer, "Transfer learning by distribution matching for targeted advertising," in *Advances in Neural Information Processing Systems 21, Proceedings of the 22nd Annual Conference on Neural Information Processing Systems*. Vancouver, Canada: MIT, 2009.

[3] H. Zhang, A. Berg, M. Maire, and J. Malik, "Svm-knn: Discriminative nearest neighbor classification for visual category recognition," *2006 IEEE Computer Society Conference on Computer ...*, Jan 2006.

[4] D. M. Bradley, "Learning in modular systems," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 2009.