SAND2008-1431C

Learning in Dynamic Environments with *Ensemble Selection* for Autonomous Outdoor Robot Navigation

Michael J. Procopio
Sandia National Laboratories
Next Generation Monitoring Systems
mjproco@sandia.gov

Jane Mulligan
University of Colorado at Boulder
Department of Computer Science
janem@cs.colorado.edu

Gregory Z. Grudic
University of Colorado at Boulder
Department of Computer Science
grudic@cs.colorado.edu

***Abstract*—Autonomous robot navigation in unstructured outdoor environments is a challenging area of active research. The navigation task requires identifying safe, traversable paths which allow the robot to progress toward a goal while avoiding obstacles. One approach is to apply Machine Learning techniques that accomplish *near to far learning* by augmenting near-field Stereo to identify safe terrain and obstacles in the far field. Some mechanism for applying *past learned experience* to the active navigation task is crucial for effective far-field classification.**

Recently, *Ensemble Selection* has been proposed as a mechanism for selecting and combining models from an existing *model library* and shown to perform well. We propose the adaptation of this technique to the time-evolving data associated with the outdoor robot navigation domain. Important research questions as to the composition of the model library, as well as how to combine selected models' output, are addressed in a two-factor experimental evaluation. We evaluate the performance of our technique on six fully labeled datasets, and show that our technique outperforms several baseline techniques that do not leverage past experience.

I. Introduction

Autonomous robot navigation in unstructured outdoor environments is a challenging area of active research and is currently unsolved. The navigation task requires identifying safe, traversable paths which allow the robot to progress toward a goal while avoiding obstacles (Fig. 1). Stereo is an effective tool in the near field, but for smooth long-range trajectory planning or fast driving an approach is needed to understand far-field terrain as well.

The data in this problem domain differ from those in traditional static contexts in that the incoming image data are streaming and batch-oriented. Further, the data are associated with *concept drift*, where the underlying distribution of the data and *target concept* can and do change over time. *Tracking* such drifting concepts, handling abrupt changes in the target concept, and recognizing *recurring contexts* all present unique challenges not present in static domains.

One approach to address the far-field terrain identification problem is to apply Machine Learning techniques to achieve *near-to-far learning* by augmenting near-field Stereo readings with learned classifications of the appearance of safe terrain and obstacles in the far field. *Classifier ensembles* containing multiple learned models can be employed as a mechanism to apply previously learned experience in this regard, enabling increased far-field terrain classification performance and providing a capability to address drifting concepts. Classifier ensembles are the focus of much recent Machine Learning research [1] and form the basis of many powerful and well-known techniques [2], [3].

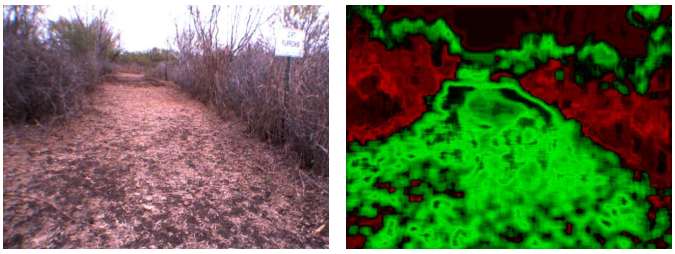


Fig. 1. Typical outdoor navigation scenario (left). Terrain classification on same image using the proposed algorithm, *Ensemble Selection* (right).

Our previous work [4], [5], [6] frames this problem as a supervised Machine Learning problem, where features based on image color are used to classify traversable terrain and obstacles in the far field. This previous work also motivates the use of classifier ensembles by demonstrating the shortcomings of first-order single-model techniques. In particular, with single-model-per-image approaches, there is no way to identify obstacles in the far field unless there are examples of those obstacles in the near field. Because this is not always the case, basic approaches to this problem lead to a common failure mode in outdoor autonomous navigation where incorrect trajectories are followed due to *short-sightedness* [7].

This paper explores in more detail the use of classifier ensembles to learn and store terrain models over time. These ensembles are constructed dynamically from a *model library* that is maintained while an autonomous vehicle navigates terrain towards some goal. The models in the ensemble are selected from the library and their outputs combined, dynamically and in real-time, in a manner designed to optimize predictive performance on far-field terrain. Towards this end, an adaptation of *Ensemble Selection* [8] is proposed as an effective means of selecting and combining models from an existing *model library*.

The contribution of this research is three-fold; in this paper, we:

The first author gratefully acknowledges the support of Sandia National Laboratories in supporting this research. Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy under contract DE-ACO4-94AL85000.

1) Propose the adaptation of Ensemble Selection to dynamic environments, and demonstrate its effective use in the outdoor robot navigation problem domain;
2) Conduct experiments to answer important questions for the community, in particular, (a) whether or not previously learned models on similar terrain can be leveraged to boost performance, and (b) how to best combine the outputs of multiple models selected from the library; and
3) Contribute natural, hand-labeled datasets taken from the problem domain and shown to contain time-varying concepts.

The remainder of this paper is organized as follows. First, in the remainder of this section, characteristics of the dynamic data associated with the problem domain are explored, and a general formulation of the library/ensemble approach is given. In Section II, the proposed algorithm, an adaptation of Ensemble Selection, is given. The experimental approach is outlined in Section III, and the results of the experiments are discussed in Section IV. Finally, conclusions and future work are provided in Section V.

A. Characteristics of Data in Dynamic Environments

The problem domain is associated with a dynamic, or *time-evolving*, environment. In contrast to more common environments, for example static ones typified by datasets found in the UCI Machine Learning Repository [9], the data dealt with in this paper are associated with many unique characteristics:

Large scale. The data arrive in the form of images at a rate of up to 30 frames/sec. Typically, there may be on the order of 50,000 training points to choose from in each 640×480 image, as determined by Stereo near-field labels. Feature dimension can also be significant, and is dependent on feature type; for color histogram (used in the experiments in this research), feature dimension d is fixed at 15 (details are given in [6]).

Streaming and Temporal. The data come arrive sequentially and in a streaming manner. Each new batch of data available represents a group of data that is associated with a frame more recent in time than the previous.

Batch-Oriented. The data is not a constant stream that can be sampled from arbitrarily. Rather, the data arrive in batches, or *chunks*, corresponding to the pixels of a single incoming image. This supports the use of traditional batch learner Machine Learning algorithms.

Real-Time. The data must be processed in real-time. There is generally not time to resample the data (e.g., with Boosting), and there is generally no time to examine previous data. Processing and taking action on the most recent incoming image (frame) is critical, because the robot will be physically moving, and obstacle avoidance and path planning are time-sensitive tasks.

Noisy. Noise can enter into the system in any number of ways, e.g., via sensor hardware (cameras), natural phenomenon (camera lens flare), and inconsistent class labels for the supervised training data (due to limitations of stereo processing).

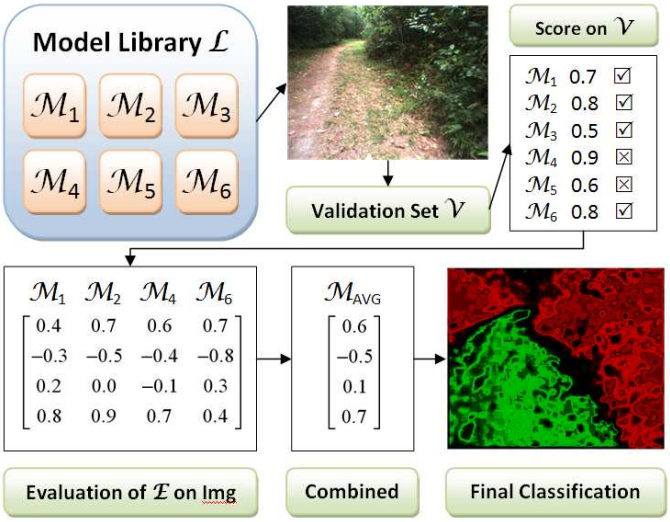


Fig. 2. Illustration of model selection and model combination from a library of models. For an incoming image I, a labeled validation set $\mathcal{V}$ is extracted from near-field Stereo labels; each model $\mathcal{M}_i$ in the library $\mathcal{L}$ is evaluated on $\mathcal{V}$. Models are selected according to some scheme to form the active ensemble $\mathcal{E}$. All models in $\mathcal{E}$ are evaluated on the entire image, and the resulting output of each model is combined. The combined output represents the final terrain classification of image I.

Drifting Concepts. The underlying distribution of the incoming data will change over time. These changes are typically gradual (the current terrain slowly changes). In some cases, the changes can be abrupt (the lighting changes, something unexpected entered the scene, or a lens flare or other camera anomaly occurred). They can also be more systematic (the robot begins a new mission in different terrain). Finally, contexts can be recurring in the sense that the terrain may gradually drift back to what the robot previously traversed some number of frames ago.

B. General Formulation of the Ensemble Approach

Classifier ensembles can be a powerful mechanism for increasing the effectiveness of machine learning techniques in a variety of problem domains. An ensemble of classifiers is simply a collection of one more classifiers (or models). An ensemble can be constructed in a variety of ways; for example, it can be dynamically created over time in response to the incoming data stream. Alternatively, a *library* of one or more models may already exist in memory, and ensembles can be selected from this library to optimize performance on the current test data. These two approaches can also be combined; in this case, a library of models is available but is also modified on-line by adding new models over time (and, if appropriate, pruning irrelevant models from the library).

The procedure for evaluating a classifier ensemble on an incoming test point $\mathbf{x}$ is more involved than for the single model case. The procedure, as described below, is illustrated in Fig. 2.

First, the classifier ensemble must be formed. This is referred to as *model selection*. Generally, for a library $\mathcal{L}$ containing M models, an ensemble $\mathcal{E}$ of K models is selected from $\mathcal{L}$,

where $K \leq M$. In practice, K can be fixed [4]; alternatively, K can be determined automatically as provided for in certain algorithms, e.g., the Ensemble Selection algorithm considered in this paper.

The K individual models that are selected to comprise $\mathcal{E}$ are sometimes referred to as *experts* or *constituent classifiers*. As a basic example, each model in $\mathcal{L}$ could be scored on its performance on validation data for the current image, with the top K scoring models selected for the final ensemble. (In this paper, Ensemble Selection accomplishes this model selection in a more elaborate manner.)

After model selection is performed, the ensemble is ready to be applied to the current task (image). Each of the selected models $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K$ in $\mathcal{E}$ is applied to the test point $\mathbf{x}$. (In practice, when the models are evaluated over an entire image, the models will be evaluated on a collection of test points (pixels), denoted X, a matrix.) The raw output of each individual model $\mathcal{M}$ at each test point $\mathbf{x}$ is denoted z; hence, the output of each individual model $\mathcal{M}$ on *multiple* test points X is denoted $\mathbf{z}$, a vector. Finally the output of all K models $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K$ on test matrix X is denoted Z, a matrix of raw model output values representing the uncombined raw ensemble output on a collection of test points (here, pixels in an image).

The uncombined final output Z of ensemble $\mathcal{E}$ on test data X is denoted by $\mathbf{q}$ and is a composite function of each individual constituent classifier's raw output. Arriving at $\mathbf{q}$ from Z is referred to as *model combination*. In the most basic sense, model combination can be a simple average of the raw outputs of each model over the test points. This is sometimes referred to as an *unweighted majority vote*. More powerfully, a *weighted average* can be used, resulting in some experts having stronger influence (i.e., more say) in the final ensemble output. Both approaches are considered in the experimental study done for this paper.

II. Proposed Approach and Motivation

A. Ensemble Selection for Dynamic Environments

Caruana et al. describe in [8] and [10] an algorithm called *Ensemble Selection*. Ensemble Selection involves selecting models from a pre-built *library of models*. This different from the approach taken by many related algorithms, for example, SEA [11], WCEA [12], and ACE [13], where a single, dynamic ensemble of models is maintained in memory. In contrast, the actual ensemble is selected from the library, and only this "active ensemble" is applied to the current image.

Ensemble Selection was originally envisioned for use in static environments. The adaptation of Ensemble Selection to dynamic environments such as the problem domain is novel. The basic ensemble selection procedure, taken from [8], is shown below.

1) Start with the empty ensemble.
2) Add to the ensemble the model in the library that maximizes the ensemble's performance to the error metric on a hillclimb (validation) set.
3) Repeat Step 2 for a fixed number of iterations of until all of the models have been used.
4) Return the ensemble from the nested set of ensembles that has maximum performance on the hillclimb (validation) set.

Thus, Ensemble Selection proceeds in a *forward greedy stepwise* manner.

Apparent in the algorithm is an inherent lack of parameterization; Ensemble Selection is considered for this reason to be *automatic* for the terrain classification task, a major advantage in this problem domain where optimal values for parameters are usually task/scene-dependent. The automatic nature of Ensemble Selection, combined with excellent demonstrated performance in the literature (on static domains), forms the basis for its proposed use in this problem domain.

The following sections discuss the operation of the Ensemble Selection algorithm. First, the underlying base learner is described; then, the procedure for ensemble construction (model selection), associated stopping criteria, and model combination is given. The section concludes with an illustrative example of the algorithm in action.

B. Base Learner

Generally, any type of base learner may be used with Ensemble Selection. Indeed, when introduced, Ensemble Selection was shown to perform well when the underlying library contained models trained with different machine learning algorithms (SVMs, decision trees, etc.) and with different learning parameters. Such diversity is shown to be beneficial in ensemble learning [1]. However, in this research, ensembles and libraries are homogenous in the sense that they contain models from one single type of base learner.

The base learner used in this research is a linear SVM with special scaling applied for obtaining probabilistic output. This scaling technique, first proposed by Grudic in [14] and later formalized by the author in [4], is characterized by the use of histograms to approximate the density of the hyperplane distance of the model outputs. This serves as a mechanism to estimate the past training "density" associated for a given test point $\mathbf{x}$ with which the model is presented. In short, the probabilistic output given by this model comes directly from its estimate of model applicability, in turn based on the distribution of the training data. This approach contrasts with that taken by Platt's scaling method [15], which returns probabilities based on signed hyperplane distance instead of point-to-hyperplane distance densities.

This capability lends itself directly to establishing *model applicability* for any given evaluation point $\mathbf{x}$. The motivation for developing this capability is a common problem with traditional classification techniques: traditionally, resulting models are applied "blindly" over some test data (here, an image), including parts of the test set (areas of the image) that may differ significantly from the training set. In this situation, the resulting model output has little or no meaning. The density approach here addresses this issue by providing an *pointwise* estimate of model applicability, where the model responds

more strongly to terrain that more closely correlates with the data on which the model was trained.

This pointwise applicability is very powerful. Used directly, the density estimate (which is scaled to be on $[0, 1]$) can be used as a *confidence value* in the given output class. (Note that this value does not represent a true probability in a Bayesian sense.)

The underlying algorithm used by the histogram method described above is linear SVM, implemented by LIBLINEAR [16], a very fast linear SVM implementation created by the authors of LIBSVM [17].

C. Ensemble Construction

Construction of the ensemble $\mathcal{E}$ proceeds in the canonical Ensemble Selection manner. Prior to ensemble construction, a balanced validation set, here denoted $\mathcal{V}$, is constructed as outlined in Sec. III-F. The ensemble is iteratively optimized to this data by adding one model at a time from the library $\mathcal{L}$ that maximizes the ensemble's performance at each step on $\mathcal{V}$. An example of this procedure is given in Table I.

D. Stopping Criteria

Ensemble Selection proceeds in the above manner until one of the stopping criteria is met. The stopping criteria are dictated by any of the three conditions below:

1) There is no model in the library that, when added to the ensemble, results in higher ensemble performance compared to just the current ensemble alone;
2) The active ensemble reaches a maximum size, fixed (by hand) at 16; and
3) While the algorithm is executing, it is interrupted and a final "best-effort" classification answer is requested by the robot.

E. Evaluation

Once any of the stopping criteria met, model selection is concluded. At this point, ensemble $\mathcal{E}$ has been selected from the library. The selected models in $\mathcal{E}$ are then each evaluated over the image. Each model's output must then be combined. The combined output of each model is the final output of the algorithm for the given input image.

F. Combination of Selected Models

After each model is evaluated over the image, those models' outputs must be combined. There are a number of ways to perform model combination; three such methods are examined below. Each method is a level in the Model Combination experimental variable.

1) Unweighted Majority Vote: One of the simplest form of combining the outputs of multiple models is achieved by taking the unweighted average of each model's output at each evaluation point. This is generally referred to as an *unweighted majority vote* of each expert's prediction.

2) Maximum Confidence Wins: Instead of taking the average of all models in $\mathcal{E}$ at each text point, the *most confident* value could be selected. This has the property that, if a model is the most confident for a particular point, its output at that point is used in the final classification. However, the fact that this is the case has no bearing on any other points; the pointwise operations are independent of each other. Once concern with this approach is that the output could be an outlier and/or the model could simply be incorrect at the given point.

3) Weighting by Confidence: When a weighted average is desired, typically, a single weight representing the importance of or belief in that model is obtained; the weighted average is then taken at each point in the evaluation set with the same set of model weights. How model weights are obtained is an area of active research; for example, weights could be derived from the model's historical performance or its performance on validation data from the current frame. Because the model outputs are confidence values, *the outputs themselves can be weights*. This allows for a unique set of weights to be used at each evaluation point, instead of just one weight per model.

G. Demonstration of Algorithm Operation

Consider the following scenario, taken directly from the results. An experimental run is being conducted on Dataset DS1B using Ensemble Selection. The library is initially empty, but models are trained on each incoming image and added to the library as the experimental run progresses. For this example, the experimental run is at frame 45, and accordingly model $\mathcal{M}_{45}$ trained on image I_{45} has just been added to the library $\mathcal{L}$. $\mathcal{L}$ thus consists of $M = 45$ models, numbered 1 through 45.

In the context of the above scenario, the iterative operation of Ensemble Selection is illustrated in Table I. In this example, the best single model in the library $\mathcal{L}$ of 45 models scored 0.82412 using the mean CCA metric on near-field validation data $\mathcal{V}$ provided by Stereo. By choosing a subset of models from the ensemble, that score increased monotonically to 0.94657 by the time the algorithm terminated, a 15% increase over the best single model. The plot of the increasing ensemble score over time is illustrated in Fig. 3.

H. Summary

This section detailed the operation of the dynamic-environments adaptation of the Ensemble Selection algorithm as applied to the terrain classification problem domain. This implementation was shown to be *automatic*, requiring no parameterization. The exact mechanics of the algorithm were given, including a discussion on the base learner; the mechanism used for combining model outputs and the related performance metric, CCA; and the three stopping criteria. The preceding three topics are novel contributions of this research, beyond what is prescribed by basic Ensemble Selection.

An optimized MATLAB implementation of Ensemble Selection for Dynamic Environments is contributed along with this paper and is available at [18].

TABLE I
DEMONSTRATION OF ENSEMBLE SELECTION OPERATION

Iteration	Model Indices / Ensemble Snapshot $\mathcal{E}_i$	Score of $\mathcal{E}_i$
0	[ ]	—
1	30	0.82412
2	30 43	0.88148
3	30 43 25	0.90181
4	30 43 25 45	0.91528
5	30 43 25 45 29	0.92378
6	30 43 25 45 29 42	0.93104
7	30 43 25 45 29 42 28	0.93483
8	30 43 25 45 29 42 28 40	0.93774
9	30 43 25 45 29 42 28 40 26	0.94007
10	30 43 25 45 29 42 28 40 26 38	0.94218
11	30 43 25 45 29 42 28 40 26 38 22	0.94399
12	30 43 25 45 29 42 28 40 26 38 22 41	0.94500
13	30 43 25 45 29 42 28 40 26 38 22 41 21	0.94590
14	30 43 25 45 29 42 28 40 26 38 22 41 21 44	0.94657
$\mathcal{E}_{final}$	30 43 25 45 29 42 28 40 26 38 22 41 21 44	0.94657

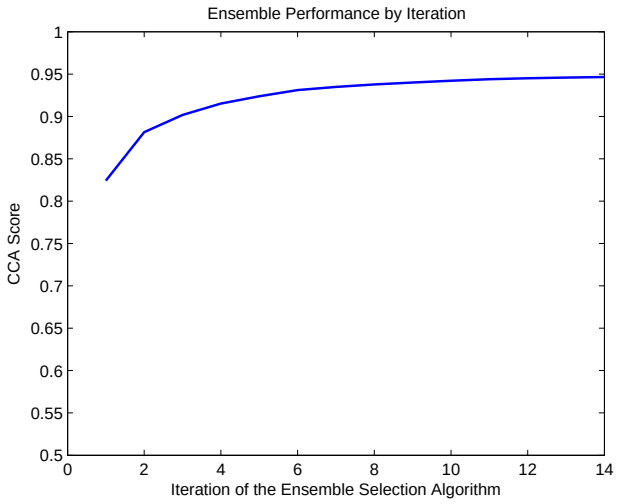


Fig. 3. Demonstration of the Ensemble Selection algorithm over time. This plot shows the mean CCA performance of the overall ensemble as it is iteratively constructed. The single best model alone scores 0.82412, while judiciously selecting the composition of the ensemble results in a final ensemble mean CCA of 0.94657, a 15% reduction in error.

III. EXPERIMENTAL DESIGN

A. Research Objectives

This paper aims to provide answers to the following research questions:

- Does the proposed adaptation of the Ensemble Selection algorithm work in dynamic environments, such as the problem domain?
- Can careful selection of models via Ensemble Selection outperform single-model-per-image approaches?
- Does the existence of any previously learned models on similar terrain in the library help performance?
- Does the manner in which models are combined result in any significant performance differences?

B. Experimental Approach

To answer these questions, an experimental framework was developed. The following points are central to this framework:

Real data: Experiments are to be performed on image sequences taken from outdoor scenarios using standard hardware found on existing robot platforms.

Varied datasets: Different terrains pose different problems, and a variety of terrain, seen under different lighting conditions, is necessary to fully test any approach.

Hand-labeled "ground truth" images: To produce meaningful performance metrics and comparisons we require ground-truth data. In this study we evaluate the output of our technique against test images hand-labeled by a human, which means all parts of the image (not just the near field) are considered in the evaluation.

Randomized experiments: In this paper, three randomized experiments are used to determine mean algorithm performance and the associated variance due to randomness present in various parts of the system.

C. Datasets

Time-evolving domains have a history of being evaluated on artificial datasets, e.g., "moving hyperplane," where the concept drift is introduced manually and any correlation to real-world problems is unestablished. This motivated the creation of *natural* datasets taken from the problem domain. The natural datasets used here are taken from actual logged test runs by robots competing in the DARPA LAGR program [7], and are shown in [6] to contain time-varying (drifting) concepts. They are part of the contribution of this paper.

Overall, three scenarios are considered. Each scenario is associated with two distinct datasets, each representing a different lighting condition. Hence, there are six datasets in all. The terrain appearing in the datasets varies greatly, with combinations of ground plane type (mulch vs. dirt vs. woods), foliage, natural obstacles (trees, dense shrubs) and man-made obstacles (hay bales). Lighting conditions range from overcast with good color definition (e.g, DS1B), to very sunny, causing shadows and saturation (e.g., DS2A). Representative images from each dataset are shown in Fig. 4. Additional images and descriptions for each dataset are provided in [6].

Each dataset consists of a 100-frame hand-labeled image sequence. Each image was manually labeled, with each pixel being placed into of three classes: Obstacle, Groundplane, or Unknown. If it was difficult for a human to tell what a certain area of an image was—even when using context—then that region was labeled as Unknown.

The datasets, hand-labelings, and a tool to aid in labeling have all been made publicly available on the web at [18]. They are in MATLAB format (version 6 compatible) and include both pre-calculated stereo masks and feature images for ease of use. These can also be used directly in future experiments, as the included pre-calculated stereo mask and feature image for each frame are the same ones used in the experiments done for this paper.

Fig. 4. Representative images from each of the six datasets: DS1A and DS1B (top); DS2A and DS2B (middle); and DS3A and DS3B (bottom).

D. Experimental Variables

Along with the more general goal of empirically demonstrating the performance of the proposed approach, the experiments are further associated with two factors, or experimental variables. The first factor is whether or not the library $\mathcal{L}$, at the start of the experimental run, is *pre-populated* with models from similar (but not identical) terrain. If so, those models will be available from the onset of the experimental run for selection and application to incoming images. If not, then only models from the current terrain will be available. In each case, models from the current run are added to the library, one model per image. The second factor is the model combination technique, described in Sec. II-F, and has three levels.

Everything else in the experiments is kept fixed, e.g., feature extraction technique, base learner and associated learning parameters, Stereo parameters, etc.

E. Performance Metric

The performance metric used in this study is Area Under the ROC Curve (AUC), a summary statistic associated with ROC (Receiver Operating Characteristic) curves [19]. AUC is excellent, well-calibrated ranking metric, and is equivalent to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance.

F. Balanced Training and Validation Sets

In these experiments, training and validation sets are always created with a balanced class distribution, where there are equal numbers of groundplane and obstacle examples. The creation of such balanced sets is motivated by the assumption made by many classifiers that training examples are evenly distributed among different classes [20]. Unbalanced training data sets are thus generally considered an unfavorable condition, and in some cases may require special handling by the algorithm or by the end user.

The mechanism used to create balanced training datasets is straightforward and is trivial when the condition of a *maximally large* balanced training data set is imposed. (With the use of LIBLINEAR linear SVM implementation, such large datasets do not pose a large computational burden.) With such a condition in place, the training data set should not only be balanced, but should be as large as possible given the two-class population of all possible training labels. Thus, all of the T training examples from the minority class are selected as training data, and a random sample of size T of the majority class is also selected. Finally, these two sets, both of size T, are combined to create a balanced training data set of size $2T$. When validation (or holdout) data are needed, validation sets are created in a similar manner.

G. Far-Field Evaluation and the Far-Field Band

In general, the aim of this research is to understand terrain in the *far field*, between 10m and 100m away from the robot. Traditional approaches such as Stereo are generally able to identify obstacles in the near-field. However, navigation relying solely on understanding the near-field terrain is the source of a number of common navigational failure modes, e.g., stereo short-sightedness, as described in Sec. I of this paper. This motivates experiments aimed at evaluating the performance of approaches specifically in the far-field.

The region in the two-dimensional image that represents the "far field" for the purposes of these experiments is comparatively small. The far field is formally defined to be the area of the image that corresponds to beyond 10m of the robot, but within 100m. In front of this region is the near field, handled adequately by stereo. Further back from this region are typically areas above the horizon line. In between these extremes lies the "far-field band," comprising 8.40% of the image. Further details are given in [6]. *Only pixels in the far-field band are taken into account when scoring algorithm output.*

IV. Experimental Results

Raw experimental data for the study is given in Table II. Overall, the data are indicative of very strong performance of the Ensemble Selection algorithm adapted for the terrain classification task. In particular, the performance shown here is significantly better overall than the one-model-per-image results published in [6], which used the same datasets and experimental approach.

Surprisingly, there was no statistically significant difference in the results for starting an experimental run with an empty library, versus starting it with a pre-populated library of models taken from similar terrain (but not the actual current dataset). That is not to say, necessarily, that such models would not be useful in a classification context; it may point to a shortcoming of Ensemble Selection's ability to correctly select and apply previously learned terrain models. Another possibility is that those models may overfit to the validation data, and thus generalize poorly in the far-field (where the experimental evaluation takes place in this study). We speculate that the most useful models for the current image are probably terrain models learned *most recently on the actual current course (or mission)*. Certainly, these indifferent results for this experimental variable do not support the additional computational burden that was observed.

Fig. 5 compares the output of Frame 1 of DS3A for the two different starting library scenarios. The image on the left shows reasonable classification, and is achieved by just a single model built on the actual frame. The image on the right shows terrain classified by multiple models selected from a pre-populated ensemble of models from similar terrain. Here, these models are useful; the segmentation appears more robust, more confident, and areas of uncertainty in the far-field in the first image are filled in in the second. This particular case demonstrates the power of leveraging previously learned terrain models to achieve better far-field classification.

The most significant result of the study was a clear difference in the performance of the model combination methods. Overall, the *unweighted majority vote* model combination mechanism was the best performer, and this was a statistically significant result at the 90% confidence level. Weighting by confidence was a close second (and, for some datasets, did not perform statistically better or worse than an unweighted majority vote). Taking the maximum confidence value at each point resulted in the worse performance overall and also for each dataset. The message here is clear; this factor is significant, and a simple average yields the best results.

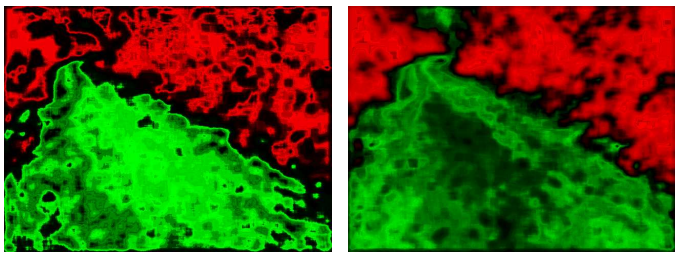
Fig. **??** compares the three different model combination methods for Frame 50 of DS1B. The unweighted majority vote output shows more conservative classification, particularly in areas that are strongly misclassified in the other two model combination methods (e.g., the lower left region of the image).

The difference in the performance of the unweighted majority vote and maximum-confidence model combination methods is shown clearly in the plot in Fig. 7. In the beginning of the experimental run, the performance difference is not much, but quickly becomes apparent as the run proceeds. The unweighted average technique outperforms on each frame.

TABLE II
SUMMARY OF EXPERIMENTAL RESULTS – AUC

Dataset	Lib Type	Combination Method		
		Unweighted[a]	Max Confidence[b]	Weighted Conf[c]
DS1A	**EMPTY**	90.56 ±0.26[d]	84.52 ±0.50	89.15 ±0.43
	POPULATED	89.60 ±0.52	82.81 ±1.22	88.28 ±0.23
DS1B	**EMPTY**	91.47 ±0.10	86.30 ±0.95	89.33 ±0.29
	POPULATED	91.41 ±0.28	86.13 ±0.72	90.55 ±0.83
DS2A	**EMPTY**	97.18 ±0.05	95.15 ±0.89	97.52 ±0.36
	POPULATED	97.46 ±0.12	94.21 ±0.30	97.32 ±0.12
DS2B	**EMPTY**	85.41 ±0.96	78.43 ±0.48	83.48 ±1.42
	POPULATED	85.03 ±0.67	77.92 ±0.73	84.33 ±0.85
DS3A	**EMPTY**	99.52 ±0.08	99.01 ±0.04	99.05 ±0.08
	POPULATED	99.86 ±0.02	99.29 ±0.05	99.52 ±0.01
DS3B	**EMPTY**	99.05 ±0.02	98.11 ±0.18	98.83 ±0.04
	POPULATED	99.57 ±0.02	98.53 ±0.04	99.51 ±0.03
OVERALL[e]	**EMPTY**	93.95 ±0.20	90.40 ±0.17	93.02 ±0.27
	POPULATED	93.91 ±0.23	89.95 ±0.03	93.35 ±0.25

[a] Unweighted majority vote (simple average).
[b] Select single most confident model.
[c] Weighted average, weighted by confidence.
[d] Standard deviation of three repeated measures.
[e] Overall performance, mean over all datasets.



(a) Starting with Empty Library

(b) Starting with Existing Library

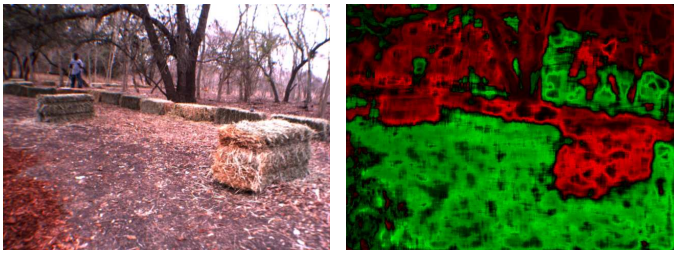
Fig. 5. Comparison of starting library types for frame 1 of DS3A.

V. CONCLUSIONS AND FUTURE WORK

This paper proposed the adaptation of Ensemble Selection, initially designed for static domains, to the dynamic, time-evolving environments associated with terrain classification in the outdoor robot navigation problem domain. This algorithm was selected because it is an *automatic* approach to addressing the model selection and model combination challenges involved in leveraging classifier ensembles for terrain segmentation.

The results of the experimental analysis support three primary conclusions. First, Ensemble Selection is an effective technique to apply multiple model learning to the terrain segmentation task. Its performance exceeds previously published results for single-model-per-image approaches. Second, the presence of pre-computed models in the model library available at the start of an experimental run did not result overall in any statistically significant performance differences. Third, the model combination technique was a significant factor and a simple average, or unweighted majority vote, was found to be the combination technique that resulted in the best performance.

Finally, this paper contributes six novel, hand-labeled, natural datasets to the community. These datasets are taken



(a) RGB Image DS1B Frame 50 (b) Unweighted Majority Vote

(c) Single Most Confident (d) Weighted by Confidence

Fig. 6. Comparison of output for Model Combination methods. The source RGB image is shown in 6(a).

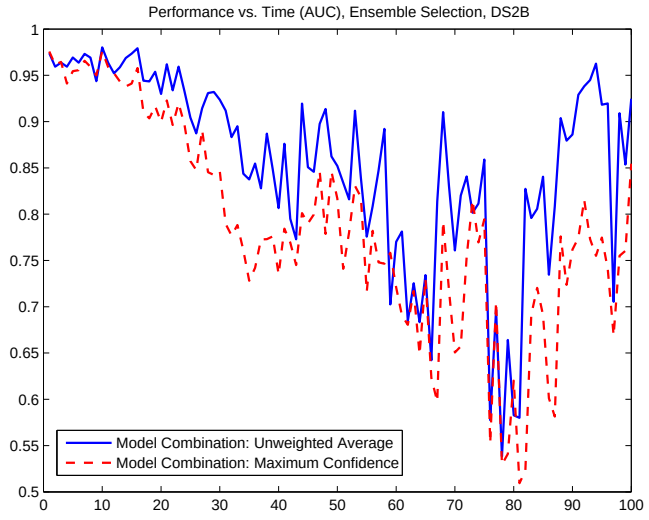


Fig. 7. Performance vs. Time for Ensemble Selection in DS2B for two different model combination methods.

from the problem domain and provide the basis for future experimentation.

Future work will focus on minimizing overfitting of the selected ensemble on near-field validation data which can result in reduced accuracy in the far field; various mechanisms for doing so are discussed in [8] and [10].

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of Sandia National Laboratories, the National Science Foundation, the DARPA LAGR program, and reviewers' comments.

REFERENCES

[1] T. G. Dietterich, "Ensemble methods in machine learning," in *MCS '00: Proceedings of the First International Workshop on Multiple Classifier Systems*. London, UK: Springer-Verlag, 2000, pp. 1–15.

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

[3] R. Schapire, "The boosting approach to machine learning: An overview," 2001.

[4] M. J. Procopio, J. Mulligan, and G. Grudic, "Long-term learning using multiple models for outdoor autonomous robot navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[5] M. J. Procopio, T. Strohmann, A. R. Bates, G. Grudic, and J. Mulligan, "Using binary classifiers to augment stereo vision for enhanced autonomous robot navigation," University of Colorado at Boulder, Boulder, CO, Tech. Rep. CU-CS-1027-07, April 2007.

[6] M. J. Procopio, "An experimental analysis of classifier ensembles for learning drifting concepts over time in autonomous outdoor robot navigation," Ph.D. dissertation, University of Colorado at Boulder, Department of Computer Science, December 2007, available at http://ml.cs.colorado.edu/~procopio/phdthesis/.

[7] L. Jackel, E. Krotkov, M. Perschbacher, J. Pippine, and C. Sullivan, "The DARPA LAGR program: Goals, challenges, methodology, and Phase I results," *Journal of Field Robotics*, vol. 23, pp. 945–973, November/December 2006.

[8] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *ICML '04: Proceedings of the twenty-first international conference on Machine learning*. New York, NY, USA: ACM Press, 2004, p. 18.

[9] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz, "UCI repository of machine learning databases," 1998, http://www.ics.uci.edu/~mlearn/MLRepository.html.

[10] R. Caruana, A. Munson, and A. Niculescu-Mizil, "Getting the most out of ensemble selection," in *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 828–833.

[11] W. N. Street and Y. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification," in *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM Press, 2001, pp. 377–382.

[12] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM Press, 2003, pp. 226–235.

[13] K. Nishida, K. Yamauchi, and T. Omori, "Ace: Adaptive classifiers-ensemble system for concept-drifting environments." in *Multiple Classifier Systems*, 2005, pp. 176–185.

[14] G. Grudic, J. Mulligan, M. Otte, and A. Bates, "Online learning of multiple perceptual models for navigation in unknown terrain," in *FSR '07: Proceedings of the International Conference on Field and Service Robotics*, 2007.

[15] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schoelkopf, and D. Schuurmans, Eds., 2000, pp. 61–74.

[16] C.-J. Lin, R. C. Weng, and S. S. Keerthi, "Trust region Newton method for large-scale logistic regression," in *Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007, software available at http://www.csie.ntu.edu.tw/~cjlin/liblinear.

[17] C. C. Chang and C. J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[18] M. J. Procopio, "Hand-labeled DARPA LAGR datasets," http://ml.cs.colorado.edu/~procopio/.

[19] F. J. Provost and T. Fawcett, "Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions," in *Knowledge Discovery and Data Mining*, 1997, pp. 43–48. [Online]. Available: citeseer.ist.psu.edu/article/provost97analysis.html

[20] J. Zhang and I. Mani, "KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction," in *Proceedings of the ICML 2003 Workshop on Learning from Imbalanced Datasets*, 2003.