Terrain Segmentation with On-Line Mixtures of Experts for Autonomous Robot Navigation

Michael J. Procopio[1], W. Philip Kegelmeyer[1], Greg Grudic[2], and Jane Mulligan[2]

[1] Sandia National Laboratories,
Albuquerque, NM 87185; Livermore, CA 94551
{mjproco,wpk}@sandia.gov
[2] Department of Computer Science,
University of Colorado at Boulder,
Boulder, CO 80309
{grudic,janem}@cs.colorado.edu

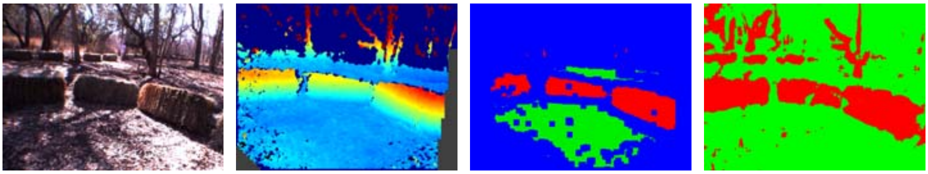
Abstract. We describe an on-line machine learning ensemble technique, based on an adaptation of the *mixture of experts* (ME) model, for predicting terrain in autonomous outdoor robot navigation. Binary linear models, trained on-line on images seen by the robot at different points in time, are added to a model library as the robot navigates. To predict terrain in a given image, each model in the library is applied to feature data from that image, and the models' predictions are combined according to a single-layer (flat) ME approach. Although these simple linear models have excellent discrimination in their local area in feature space, they do not generalize well to other types of terrain, and must be applied carefully. We use the distribution of training data as the source of the *a priori* pointwise mixture coefficients that form the soft gating network in the ME model. Single-class Gaussian models are learned during training, then later used to perform density estimation of incoming data points, resulting in pointwise estimates of *model applicability.* The combined output given by ME thus permits models to abstain from making predictions for certain parts of the image. We show that this method outperforms a less sophisticated, non-local baseline method in a statistically significant evaluation using natural datasets taken from the domain.

Keywords: Mixture of Experts, Classifier Ensembles, Local Classifier Accuracy, Online Learning, Terrain Segmentation, Autonomous Robot Navigation.

1 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 that allow the robot to progress towards a goal while avoiding obstacles. Stereo vision allows for obstacle avoidance in the

J.A. Benediktsson, J. Kittler, and F. Roli (Eds.): MCS 2009, LNCS 5519, pp. 385–397, 2009.



(a) RGB Image (b) Gnd. Plane Dev. (c) Stereo Labels (d) Classification

Fig. 1. Demonstration of near-to-far learning using stereo. In (c) and (d), and throughout this paper, red represents nontraversable obstacle (positive); green represents traversable groundplane (negative).

near field (here, within 10 m of the robot). However, navigating solely on near-field terrain readings can lead to a common failure mode in outdoor autonomous navigation where incorrect trajectories are followed due to *nearsightedness*, or an inability to distinguish safe and unsafe terrain in the far field [1].

To address near-sighted navigational errors, *near-to-far learning* is often used [2,3]. The near-to-far approach uses both appearance and stereo information from the near field as inputs for training appearance-based models; these models are then applied in the far field in order to predict safe terrain and obstacles farther out from the robot where stereo readings are unavailable.

Near-to-far learning using stereo is demonstrated in Figure 1. For a given RGB image (1a), stereo disparity is computed using a stereo camera pair; from this data, a groundplane model is fit and subtracted out, resulting in an estimate of groundplane deviation (1b). Near-field stereo labels from both the groundplane and obstacle classes are identified according to small and large groundplane deviation values, respectively (1c); these near-field stereo labels are sampled to create a balanced training set. Next, features are extracted from the image at the pixels of this training set; here, *color histograms* are used [3]. A model is then trained on the resulting near-field feature data. The resulting model is evaluated over the image, including the far field, to arrive at a final terrain predictions (1d). These terrain predictions are used by the robot's path planning system to influence the robot's low-level navigation [1].

Recently, the use of classifier ensembles to learn and store terrain models over time for application to future terrain has been investigated [3,4,5]. These ensembles are constructed dynamically from an on-line *model library* that is maintained as the robot navigates terrain towards some goal. For an incoming image, the outputs of the models in the resulting ensemble are combined, dynamically and in real-time, in a manner designed to optimize predictive performance on far-field terrain. This previous work achieved classifier fusion by taking a linear combination of model outputs using *one* weight per each model in the ensemble. This technique's primary disadvantage is that models cannot be experts *locally*, i.e., at pixel resolution, which is problematic because a given model's discrimination ability may not apply everywhere in the image (i.e., in input space).

2 Terrain Segmentation with Mixtures of Experts

To overcome the shortcomings of the single-weight *non-local* methods noted above, we describe an efficient on-line machine learning ensemble technique. It is based on an adaptation[1] of the *mixture of experts* (ME) model [6,7,8]. The general aim of the approach is to combine multiple experts learned over time by extracting a soft partitioning of the feature space to yield *local accuracy estimates* [9], while also respecting the real-time domain requirement.

On-line mixtures of experts and related mixture models are not novel, nor is the use of Gaussian models to partition the input space and inform where local linear models are applicable. In particular, Sato and Ishii in [10] use the Normalized Gaussian Network [11], or *NGnet*, as the basis for their proposed on-line EM algorithm, used in turn to fit model parameters.

The primary differentiating contribution of this paper is threefold. First, we place explicit emphasis on permitting the mixture of experts to *abstain* from making predictions, where appropriate. Second, our overall approach is for the purpose of binary classification, which requires adaptation and extension of the *NGnets* above. This is contrast to the the approaches taken in [10] and [11], which are framed in a regression context and hence do not model class-conditional data distributions. Finally, this approach has not been previously applied to the terrain segmentation task or to other open problems in the autonomous robot navigation domain.

2.1 Overview of the ME Approach for Terrain Segmentation

During navigation, terrain in an outdoor scene is to be classified as either traversable (groundplane) or nontraversable (obstacle). Two-class appearance-based linear models, trained on-line on images seen by the robot at different points in time, are added to a model library as the robot navigates; the training data is not kept after the model has been added to the library. When terrain segmentation is required, each model in the library is applied to feature data extracted from the current image, and the models' predictions are combined according to the single-layer (flat) ME model. Because mixing coefficients in the ME model are functions of the input data, models can be experts locally in feature space.

Model Abstinence. A key benefit of our approach is model *abstinence*, i.e., when it is determined that a model does not apply to some point $\mathbf{x}$ and hence should be permitted to *abstain* (or mostly abstain) from making a prediction at that point. Not only can individual models abstain, but in our approach, the entire ensemble can abstain at $\mathbf{x}$. Such behavior is desirable in the autonomous

[1] Our approach adopts the mixture of experts architecture from [6]. This is fundamentally a conditional mixture model in which the mixing coefficients, like the expert response, are functions of the input. We do not explicitly fit the model using EM as outlined in [7].

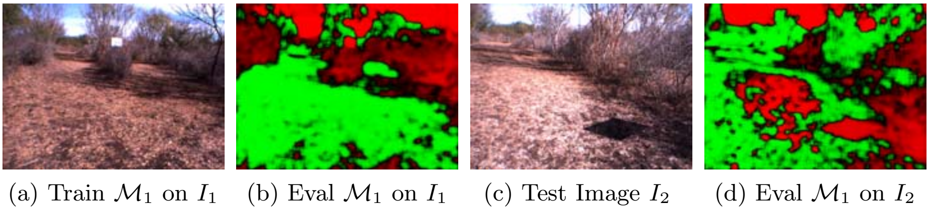

(a) Train $\mathcal{M}_1$ on I_1 (b) Eval $\mathcal{M}_1$ on I_1 (c) Test Image I_2 (d) Eval $\mathcal{M}_1$ on I_2

Fig. 2. Good specialization but poor generalization of linear experts. A model trained on near-field training data from an image (2a) yields reasonable terrain classification on that image (2b). That same model, when applied to a different test image (2c), yields reasonable segmentation in some parts of the image, but not in others (2d).

robot navigation domain, because the robot's path planning system (the *planner*) maintains a probabilistic *cost map* over time used for path planning [1]. If there is not a sufficient basis to make terrain predictions, then the resulting fully abstaining ensemble will result in no updates to the robot's cost map. The next incoming frame would then offer a new opportunity for terrain prediction and cost map updates.

Behavior of Linear Models on Domain Data. Linear models are very efficient to train, which is important in this real-time domain because training and terrain segmentation are done on-line on incoming images while the robot is navigating. We have found that these simple linear models are specialists in their area of feature space, yielding good segmentation on terrain similar to that on which they are trained, but they but do not generalize well to other terrain. Blindly applying these models without regard to their *applicability* to the input is problematic, and can lead to poor terrain prediction as the model is forced to generalize to regions of feature space on which it was not trained.

Fig. 2 shows an example of this.[2] From features extracted from a particular training image (2a), a model is trained. Terrain prediction is reasonable when the model is applied to the image on which it was trained (2b). For a test image with similar terrain appearing later in same data set (2c), the model trained on the original image is applicable only in certain areas (2d), yet still remains equally confident everywhere.

Local Applicability Estimates. For determining where models are applicable, we use the *distribution of training data* as the source of the mixing coefficients that form the soft *gating network* in the ME model. This gating network determines which models are applicable and to what extent for each input (feature

[2] In Fig. 2 and in similar figures throughout the paper, red coloring indicates non-traversable obstacle terrain prediction (model output approaching +1); green coloring indicates traversable groundplane terrain prediction (model output approaching −1); and color intensity indicates prediction confidence, with black representing full uncertainty (model output of 0).

vector) $\mathbf{x}$ in some set of data X. During training of model $\mathcal{M}$, in addition to training the linear model $(\mathbf{w}, b)$, a multivariate Gaussian model $\mathcal{G}$ is fit to each of the C classes of training data (here, $C = 2$). Thus, a model $\mathcal{M}$ comprises $\{\mathcal{G}_{c=1}, \mathcal{G}_{c=2}, \mathbf{w}, b\}$. These Gaussian models are trivial to train and very efficient to evaluate on incoming feature data.

Later, during evaluation (i.e., when segmenting terrain for an incoming image), the density models $\mathcal{G}$ are used to determine how similar the incoming data is to the data on which a previously learned expert was trained; the density model thus provides pointwise estimates of model applicability. The mixing coefficients, which are functions of $\mathcal{G}$ and $\mathbf{x}$, are combined with the expert output at $\mathbf{x}$ according to the ME model to yield the final ensemble output.

Because they are learned on the same training data, a natural concern would be that the mixing coefficients (model applicability estimates) and the model output are not independent. We examined this and determined that the two variables are poorly correlated ($R^2 \approx 0.4$), and conclude that they measure different things.

This approach is similar in principle to that proposed by Grudic *et al.* [12], who also sought a mechanism for applying models only where applicable. There, as here, density models are used to inform when and where to apply models. In that approach, histogram-based density models are used, trained on decision boundary distances of holdout data evaluated through the linear model. Our approach, in contrast, is based on Gaussian density models learned directly from training data. Further, in Grudic's approach, the final classifier output is the output of the density model response, the input for which is the output of the experts (i.e., decision boundary distances). In contrast, in our approach, final classifier output is a fusion of the density model response with the expert output, per the ME model.

Mixture of Experts Model. The mixture of experts (ME) model [6,7,8] on which our approach is based is a type of *conditional mixture model* where the mixing coefficients are functions of the input, shown in Eq. 1:

$$p(\mathbf{t}|\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\mathbf{x}) p_k(\mathbf{t}|\mathbf{x}) \ , \tag{1}$$

where the individual component densities $p_k(\mathbf{t}|\mathbf{x})$ are the *experts*, and the mixing coefficients $\pi_k(\mathbf{x})$ are known as *gating* functions [13].

ME Model Adaptation. Importantly, this initial research involves an adaptation of the ME model, where expert predictions are scaled to be on $[-1, +1]$ (see Eq. 2), and hence cannot be considered true probabilities. For this reason, $p(\mathbf{t}|\mathbf{x})$ also falls on $[-1, +1]$. In this scaling, values approaching -1 indicate groundplane predictions of increasing confidence, values approaching $+1$ indicate obstacle predictions of increasing confidence, and values approaching 0 indicate increasing uncertainty.

This modification is motivated by numerical considerations, as shown by the following scenario. Consider some test point $\mathbf{x}$ for which an expert predicts fully uncertain output, e.g. when $\mathbf{x}$ lies on that expert's decision boundary. Numerically, the uncertain expert prediction must be propagated, regardless of applicability (mixing coefficient). Scaling the expert output on $[-1, +1]$ accommodates this requirement, where uncertain expert output is represented as 0. Otherwise, if expert output were on $[0, 1]$ where uncertain output is represented as 0.5, the final prediction will be distorted by the applicability estimate falling on $[0, 1]$.

In the future, more sophisticated evolutions of our approach will eliminate the need for this adaptation in part by the inclusion of a fully uncertain, generic component that models everything the other components fail to model.

2.2 Experts: Logistic Regression Models

The experts in our technique are *logistic regression* models [14], a linear classification method common in statistics and machine learning, appropriate for predicting the certainty of a binary outcome. Moreover, they are very efficient to compute on large-scale data, motivating their use in the real-time scenarios considered here.

Given a data instance $\mathbf{x}$ and associated model weights $(\mathbf{w}, b)$, logistic regression calculates a continuous probability of the positive output class y for some test instance $\mathbf{x}$ according to the following probability model:

$$P(y = \pm 1 | \mathbf{x}, \mathbf{w}, b) = \frac{1}{1 + \exp\left[-y(\mathbf{w}^T \mathbf{x} + b)\right]} \ , \tag{2}$$

where $\mathbf{w}$ and b are estimated when training the model by minimizing the negative log-likelihood on training data [15].

2.3 Mixing Coefficients: Gaussian Density Models

The mixing coefficients in our technique are determined by Gaussian density models fit to training data when training the expert. When training expert k, a single multivariate Gaussian model $\mathcal{G}_{k,c}$ is learned for each class c of training data from the current image, using the sample mean and covariance of that data. During terrain segmentation, for each test point (i.e., feature vector corresponding to a pixel in the image) $\mathbf{x}$, the mixing coefficients for model k are determined by the response of $\mathcal{G}_{k,c}$ at $\mathbf{x}$:

$$\mathcal{G}_{k,c}(\mathbf{x}|\boldsymbol{\theta}_{k,c}) = \frac{1}{(2\pi)^{d/2} \left|\boldsymbol{\Sigma}'_{k,c}\right|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{k,c})^T (\boldsymbol{\Sigma}'_{k,c})^{-1} (\mathbf{x} - \boldsymbol{\mu}_{k,c})\right] , \tag{3}$$

where $\mathbf{x}$ is a d-dimensional feature vector, $\boldsymbol{\mu}_{k,c}$ is a d-dimensional mean vector, $\boldsymbol{\Sigma}'$ is a scaled $d \times d$ covariance matrix, and $|\boldsymbol{\Sigma}'|$ denotes the determinant of $\boldsymbol{\Sigma}'$. $\boldsymbol{\theta}_{k,c}$ is a parameter set comprising:

$$\boldsymbol{\theta}_{k,c} = \{\boldsymbol{\mu}_{k,c}, \boldsymbol{\Sigma}_{k,c}, \alpha\} \ , \tag{4}$$

where $\boldsymbol{\mu}_{k,c}$ and $\boldsymbol{\Sigma}_{k,c}$ are the sample mean and covariance, respectively, of the training data used to fit $\mathcal{G}_{k,c}$. Finally, $\boldsymbol{\Sigma}_{k,c}$ is subject to scaling by some factor α and application of additive white Gaussian noise ε with mean μ_n and variance σ_n^2:

$$\boldsymbol{\Sigma}'_{k,c} = \alpha \left[\boldsymbol{\Sigma}_{k,c} + \varepsilon \right] , \quad \text{where} \quad \varepsilon \sim \mathcal{N}(\mu_n, \sigma_n^2) \ . \tag{5}$$

Gaussian Noise $\boldsymbol{\varepsilon}$. In implementation, we found that the covariance $\boldsymbol{\Sigma}$ was often singular, and hence not invertible, for feature data typical in this domain; this has been encountered before in similar scenarios when $d \ll N$ [16]. While numerically stable, computing the pseudoinverse was found to yield unsatisfactory results in the response of Eq. 3 for our input. Our solution, known in the literature [16], was to apply a small amount of additive white Gaussian noise to the feature data such that the inverse of the resulting covariance matrix would be defined (Eq. 5). We define this noise ε to be $\sim \mathcal{N}(\mu_n, \sigma_n^2)$, set $\mu_n = 0$, and take an adaptive approach for σ_n^2: starting at 0.01, it is increased by 0.01 until the resulting scaled covariance matrix becomes invertible.

Covariance Scaling Parameter $\boldsymbol{\alpha}$. In initial experimentation with the ME technique, we observed correct behavior in terms of strong response of Eq. 3 near the Gaussian peak (sample mean $\boldsymbol{\mu}_k$), and monotonically decreasing values as $|\mathbf{x} - \boldsymbol{\mu}_k|$ increased. However, the cutoff was too sharp, with most of the output distributed below a value of 0.1. The intuition is that the peak of the density model was too steep (i.e., the distribution was too peaked). Our solution was to scale the covariance $\boldsymbol{\Sigma}$ by some factor α in order to have a more gradual decline in the response of Eq. 3 for increasing $|\mathbf{x} - \boldsymbol{\mu}_k|$. We used $\alpha = 8.0$ in the experiments, which was determined ad-hoc from a sensitivity analysis. In a more sophisticated approach, it is natural that α be data-driven; we will investigate this possibility in future work.

Scaling of the Density Output. The response of the density output of $\mathcal{G}_{k,c}$ in Eq. 3 is scaled such that the value at the peak of the Gaussian distribution (i.e., at the mean $\boldsymbol{\mu}_{k,c}$) is 1. Hence, the mixing coefficient $\pi_{k,c}(\mathbf{x})$ will be maximal when $\mathbf{x}$ is close to the sample mean of the data used to train the model. This scaling is achieved by dividing the response of Eq. 3 by its output at the sample mean of the data used to train the model:

$$\pi_{k,c}(\mathbf{x}|\boldsymbol{\mu}_k) = \frac{\mathcal{G}_{k,c}(\mathbf{x})}{\mathcal{G}_{k,c}(\boldsymbol{\mu}_{k,c})} . \tag{6}$$

Determination of $\boldsymbol{\pi_k}(\mathbf{x})$ from $\boldsymbol{\pi_{k,c}}(\mathbf{x})$. The final mixing coefficient $\pi_k(\mathbf{x})$ is simply the *maximum* of the C single-class density model outputs $\pi_{k,c}$ at $\mathbf{x}$:

$$\pi'_k(\mathbf{x}) = \max_c \left[\pi_{k,c}(\mathbf{x}) \right] . \tag{7}$$

Alternatives for future investigation include determining $\pi_k(\mathbf{x})$ as either the simple or the weighted mean of the C density model outputs $\pi_{k,c}(\mathbf{x})$, instead of the maximum.

2.4 Modifications Permitting Ensemble Abstinence

With a traditional mixture model, although individual models can abstain from making predictions, the entire ensemble cannot, because $\sum_k \pi_k(\mathbf{x}) = 1$. This effectively assumes that the correct model, or combination of models, exists in the ensemble for a given data instance $\mathbf{x}$, which we do not wish to assume. In particular, we wish to allow the entire *a posteriori* ensemble output to abstain for certain $\mathbf{x}$.

Consider the case where all models in the ensemble predict +1 for $\mathbf{x}$, but whose mixing coefficients at $\mathbf{x}$ are all 0.1. In the traditional ME approach, the mixing coefficients would be scaled to sum to 1, resulting in a full confidence +1 output. This output is not reflective of the underlying models' low applicability at $\mathbf{x}$.

On the other hand, if no scaling is done, undesirable behavior will result. Consider if there were 10 experts in the ensemble, all predicting +1 at $\mathbf{x}$, each with mixing coefficients of 0.1. With no scaling of the mixing coefficients, their cumulative ME sum would result in a final, full-confidence prediction of +1. In the same scenario but with 20 such experts, without scaling the final ME output would be greater than 1; some scaling is needed to bound the output.

In short, a scaling approach is needed that bounds final ME ensemble output, but also allows the ensemble to abstain if its underlying experts all wish to abstain. Our solution is to scale the sum of the K mixing coefficients at $\mathbf{x}$ to the K experts' *mean density model response* at $\mathbf{x}$, such that $\sum_k \pi_k(\mathbf{x}) = \frac{1}{K}\sum_k \pi'_k(\mathbf{x})$. This is achieved by dividing each unscaled mixing coefficient $\pi'_k(\mathbf{x})$ by the total number of experts K:

$$\pi_k(\mathbf{x}) = \frac{\pi'_k(\mathbf{x})}{K} \ . \tag{8}$$

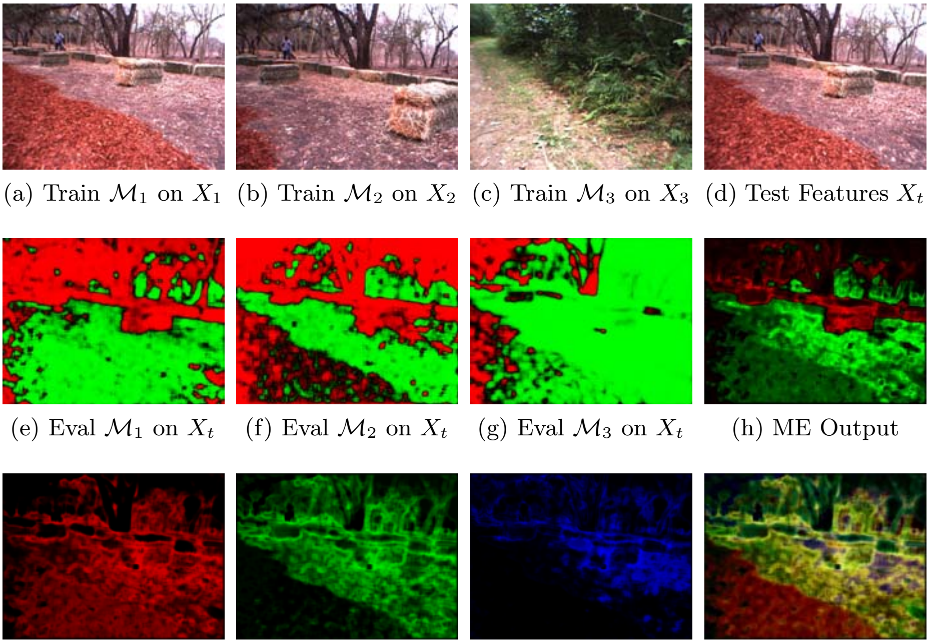
Thus, if the individual experts all have low applicability estimates, the final ensemble output will be reflective of this and will have an appropriate low-confidence response.

2.5 Conceptual Overview of the ME Approach Applied to the Domain

A conceptual overview of the approach as applied to terrain segmentation is shown in Fig 3. This is also the basis of the experimental approach discussed in Sec. 3.1.

Consider an ensemble composed of three models $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$. These models are trained on the corresponding feature data sets X_1, X_2, and X_3, extracted from images I_1, I_2, and I_3, respectively (shown in Figs. 3a–3c). A fourth image I_t, and its associated feature data set X_t, is the current target image requiring terrain classification (Fig. 3d). (Note that I_1 and I_t are similar, but not identical, frames.)

Each of the three linear models $\{\mathbf{w}_k, b_k\}$ in the ensemble is applied to X_t (Eq. 2); their corresponding terrain predictions are shown in Figs. 3e–3g. In the ME approach, X_t is also evaluated through the two density models $\{\mathcal{G}_{k,c=1}, \mathcal{G}_{k,c=2}\}$



(a) Train $\mathcal{M}_1$ on X_1 (b) Train $\mathcal{M}_2$ on X_2 (c) Train $\mathcal{M}_3$ on X_3 (d) Test Features X_t

(e) Eval $\mathcal{M}_1$ on X_t (f) Eval $\mathcal{M}_2$ on X_t (g) Eval $\mathcal{M}_3$ on X_t (h) ME Output

(i) $\pi_1(X_t)$ (j) $\pi_2(X_t)$ (k) $\pi_3(X_t)$ (l) Composite $\boldsymbol{\pi}(X_t)$

Fig. 3. Conceptual overview of ME approach for terrain segmentation

learned from each the three training images, yielding the mixing coefficients π_k (Eqs. 3–8), shown in Figs. 3i–3k. The *composite coverage* $\boldsymbol{\pi}$ of the three experts is shown in Fig. 3l. The final terrain classification output from the ME approach (Eq. 1) is given in Fig. 3h.

3 Experimental Evaluation

3.1 Approach

Baseline Algorithm. We compare the ME approach to a basic *unweighted average* baseline method [3], in which the terrain classification from each model in the ensemble is averaged together to arrive at the final terrain classification.[3]

Data Sets. The evaluation is performed using six hand-labeled natural data sets taken from the domain, recently contributed by the authors [5] and made publicly available [17]. Each dataset consists of a 100-frame hand-labeled image sequence. The images were manually labeled, with each pixel being placed into one of three classes: Obstacle, Groundplane, or Unknown; further details are available [3].

[3] This can be seen as a general case of *majority voting*, appropriate when individual expert output is on $[0, 1]$ and final ensemble output on $[0, 1]$ is also desirable.

Method. The experimental method follows the conceptual approach outlined previously in Sec. 2.5 and Fig. 3. We conducted a series of 3,000 randomized experiments, drawing one training image at random from each of the six datasets for each experiment. A testing image, with known ground truth labeling, was also drawn. Six models were learned on the training images, and each model was then applied to the test image. To obtain the output of the unweighted average baseline method, the terrain classification was combined using a simple average. For the ME approach, mixing coefficients were determined, and then combined with the linear model output according to the ME model (Eq. 1) to arrive at the final ME terrain segmentation.

Evaluation. We evaluated our approach only on the pixels in that portion of the image occurring in the far field, i.e., greater than 10 m but less than 100 m from the robot. We used standard binary classification accuracy (ACC), and since the data is roughly 3:1 skew, we also report ACC for baselines of predicting all of the same class. We report the mean ACC and std. dev. across all randomized experiments.

3.2 Results and Discussion

Experimental data is provided in Table 1; sample output is shown in Fig. 4.

Statistical Analysis. The scores shown in Table 1 represent mean values of scores from 3,000 randomized experiments. Within each experiment, the scores for each method were determined by evaluation on the same test image.

Thus, a dependent-samples analysis can be performed to determine whether or not the difference in performance among algorithms is statistically significant. Because the distribution of the differences between the paired samples was found to be non-normal, the equivalent nonparametric test statistic, the *Sign* test [18], was used.

This test provided sufficient statistical evidence to infer that the medians of the differences between each population are not 0, significant at the 95% confidence level.[4] We conclude that the ME approach outperforms the baseline approach in this evaluation.

Discussion. These results illustrate the benefits of allowing individual models to abstain. In one scenario (Figs. 4a–4c), the ME method assigns low weight to the incorrect models, allowing the correct model (i.e., the one model of the six that matches the scenario of the target image I_t) to carry the most weight in the final ensemble output. In a different scenario (Figs. 4d–4f), for certain parts of the image, none of the models in the ensemble were applicable, allowing "uncertain" regions to pass through to the final output. In both cases, if models were blindly applied without regard to local accuracy, incorrect terrain predictions would have been made.

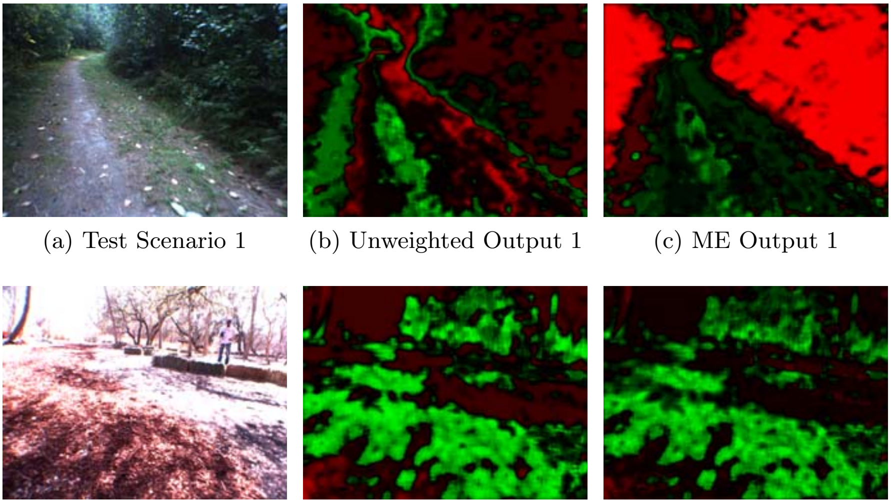
[4] Although the standard deviations for the scores appear high, they are similarly high for each group of samples, and are due to within-sample variance.

Table 1. ME Algorithm Performance vs. Baseline

Algorithm	**Score (ACC)**^{a,b}
Mixture of Experts (Local)	0.759 ± .21
Unweighted Average (Non-Local)	0.683 ± .24
Predict all Obstacle	0.726 ± .16
Predict all Groundplane	0.274 ± .16

[a] Binary accuracy, thresholded at 0.5.

[b] Mean and standard deviation of 3,000 randomized experiments.



(a) Test Scenario 1 (b) Unweighted Output 1 (c) ME Output 1

(d) Test Scenario 2 (e) Unweighted Output 2 (f) ME Output 2

Fig. 4. Experimental snapshots from two scenarios

In our experiments, we observed the technique to be real-time in our parallel implementation, when run on hardware with computational performance comparable to that of a typical robotic platform. An in-depth study of the computational characteristics of the proposed ME approach is an area for future work.

4 Conclusions and Future Work

In this paper, we presented a novel adaptation of the mixture of experts model for determining model applicability, and applied this technique to the terrain segmentation problem in the outdoor autonomous robot navigation domain. This method accommodates the key domain constraints associated with near-to-far learning for autonomous robot navigation: models are learned over time; model training and evaluation must be performed in real time; training data is not kept once a model has been trained, due to storage limitations; and training data, derived from stereo, may not be available for the target image requiring terrain segmentation.

We evaluated our approach, a local method, against a non-local, unweighted average as a baseline in a statistically significant evaluation, and concluded that the proposed mixture of experts approach outperforms the unweighted average baseline in far-field terrain prediction performance. In particular, the ME approach's inherent ability to permit individual experts to abstain from making strong predictions on a local (i.e., pointwise) basis allows for the more applicable models' predictions to carry the most weight in the final ensemble prediction.

Future work. We identify three key areas of ongoing future work. First, we plan to conduct a more in-depth experimental evaluation, varying key factors such as data-driven methods for determining α and variants of the *class density combination* function (Eq. 7) to determine impact on performance. This analysis will also involve comparison against another local method, described earlier in the paper, which takes an alternative approach for estimating local accuracy [12]; the computational performance of each will be considered.

Second, we will investigate posing this problem more formally in the hierarchical mixtures of experts (HME) context [7], using EM to fit model parameters, and making use of validation data from the target image to influence mixing coefficients.

Finally, while the notion of a mixture model with the ability to abstain appears useful, the theory behind its implementation needs to be improved. The inclusion of an uncertain, generic component should greatly improve the theoretical expression.

Acknowledgments. The authors gratefully acknowledge the contribution of Sandia National Laboratories, the National Science Foundation, the DARPA LAGR program, and the anonymous referees' insightful comments. Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

References

1. Jackel, L., Krotkov, E., Perschbacher, M., Pippine, J., Sullivan, C.: The DARPA LAGR program: Goals, challenges, methodology, and Phase I results. Journal of Field Robotics 23(11-12), 945–973 (2006)
2. Howard, A., Turmon, M., Matthies, L., Tang, B., Angelova, A., Mjolsness, E.: Towards learned traversability for robot navigation: From underfoot to the far field. Journal of Field Robotics 23(11–12), 1005–1017 (2006)
3. Procopio, M.J., Mulligan, J., Grudic, G.: Learning terrain segmentation with classifier ensembles for autonomous robot navigation in unstructured environments. Journal of Field Robotics 26(2), 145–175 (2009)
4. Procopio, M.J., Mulligan, J., Grudic, G.: Long-term learning using multiple models for outdoor autonomous robot navigation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 2007, pp. 3158–3165 (2007)

5. Procopio, M.J., Mulligan, J., Grudic, G.: Learning in dynamic environments with Ensemble Selection for autonomous outdoor robot navigation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 2008, pp. 620–627 (2008)
6. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation 3(1), 79–87 (1991)
7. Jordan, M.I., Jacobs, R.A.: Hierarchical mixtures of experts and the EM algorithm. Neural Computation 6, 181–214 (1994)
8. Jacobs, R.A.: Methods for combining experts' probability assessments. Neural Computation 7(5), 867–888 (1995)
9. Woods, K., Kegelmeyer, W., Bowyer, K.: Combination of multiple classifiers using local accuracy estimates. IEEE Trans. Pattern Analysis and Machine Intelligence 19(4), 405–410 (1997)
10. Sato, M.A., Ishii, S.: On-line EM algorithm for the normalized gaussian network. Neural Computation 12(2), 407–432 (2000)
11. Moody, J., Darken, C.J.: Fast learning in networks of locally-tuned processing units. Neural Computation 1(2), 281–294 (1989)
12. Grudic, G., Mulligan, J., Otte, M., Bates, A.: Online learning of multiple perceptual models for navigation in unknown terrain. In: FSR 2007: Proceedings of the International Conference on Field and Service Robotics (2007)
13. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, New York (2006)
14. Cox, D.R., Snell, E.J.: Analysis of Binary Data, 2nd edn. Chapman Hall, London (1989)
15. Lin, C.J., Weng, R.C., Keerthi, S.S.: Trust region Newton method for large-scale logistic regression. J. Mach. Learn. Res. 9, 627–650 (2008), `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`
16. Grudic, G., Mulligan, J.: Outdoor path labeling using polynomial mahalanobis distance. In: Proceedings of Robotics: Science and Systems (RSS), Philadelphia, PA (2006)
17. Procopio, M.J.: Hand-labeled DARPA LAGR datasets (2007), `http://ml.cs.colorado.edu/~procopio/labeledlagrdata/`
18. Hollander, M., Wolfe, D.A.: Nonparametric statistical methods, 2nd edn. Wiley-Interscience, New York (1999)