Knowing When We Don't Know: Introspective Classification for Mission-Critical Decision Making

Hugo Grimmett Rohan Paul Rudolph Triebel Ingmar Posner
Mobile Robotics Group, University of Oxford, UK
{hugo, rohanp, rudi, ingmar}@robots.ox.ac.uk

***Abstract*— Classification *precision* and *recall* have been widely adopted by roboticists as canonical metrics to quantify the performance of learning algorithms. This paper advocates that for robotics applications, which often involve mission-critical decision making, good performance according to these standard metrics is desirable but *insufficient* to appropriately characterise system performance. We introduce and motivate the importance of a classifier's *introspective* capacity: the ability to mitigate potentially overconfident classifications by an appropriate assessment of how qualified the system is to make a judgement on the current test datum. We provide an intuition as to how this introspective capacity can be achieved and systematically investigate it in a selection of classification frameworks commonly used in robotics: support vector machines, LogitBoost classifiers and Gaussian Process classifiers (GPCs). Our experiments demonstrate that for common robotics tasks a framework such as a GPC exhibits a superior introspective capacity while maintaining commensurate classification performance to more popular, alternative approaches.**

I. Introduction

The semantic mapping of a robot's workspace has become a popular line of research in recent years. A rich body of work now exists in which semantic labels are generated based on a variety of sensor modalities and classification frameworks (see, for example, [1]–[7]). Often, this is done with an implicit understanding that the application is agnostic to the classification method used: after all, for a number of classification frameworks the resulting *precision* and *recall* — quantities commonly used to characterise performance — are often commensurate across a wide variety of applications.

Contrary to this now established status-quo, we advocate that high precision and recall are desirable but do not suffice to fully characterise classification performance in robotics. The dimension missed is that spawned by a robot's ability to take action in ambiguous situations. For example, the robot may query a human operator or seek additional data for disambiguation rather than committing to a potentially incorrect class decision. Crucially, and central to this paper, this requires the classifier output to reflect an amount of ambiguity appropriate to a given situation. Even when hard class assignments are avoided by optimising an expected cost or reward, as is the case for most mission-critical decision making, a *realistic* estimate of uncertainty when modelling the state of the world is crucial; an autonomous car that misses a single traffic light with high confidence can suffer disastrous consequences (see, for example, Fig. 1).

We argue that a classifier which is uncertain when it makes mistakes but certain when classification is correct, is more desirable than a classifier which makes correct and incorrect decisions with similarly high confidence. We are therefore looking for a classifier's capacity to mitigate its assessment by an appropriate measure as to how 'qualified' it is to make a call given its own prior experience, usually in the form of training data. Following classical decision theory (e.g. [8]), mistakes are penalised by means of a loss function. However, if the underlying classification framework leads to an overconfident estimate of the class label, then it will often be ineffective regardless of the high costs imposed. Our work investigates this *introspective* capacity in a number of classification frameworks commonly used in robotics: support vector machines (SVMs), LogitBoosting and Gaussian Process classifiers (GPCs). Our treatment and findings apply to any aspect of robotics where action is required based on inference driven by raw sensor data. Here we choose to frame our exposition in the domain of autonomous driving, where mission-critical decisions equate to *safety*-critical decisions. To the best of our knowledge this is the first work in robotics characterising the introspective properties of commonly used classification frameworks.

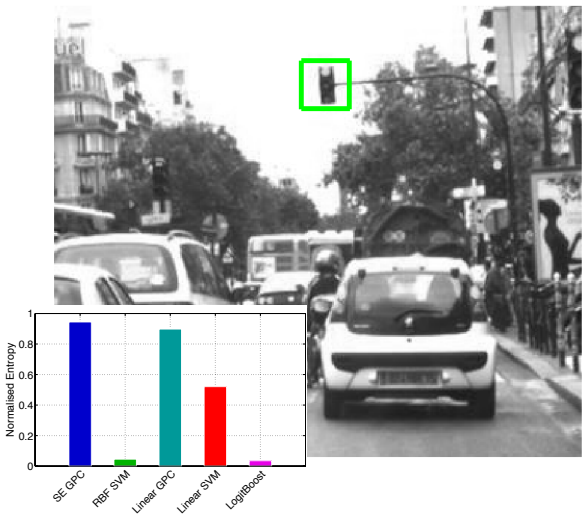


Fig. 1: Uncertainty in classification output as measured using normalised entropy for traffic light detectors based on five different classification frameworks applied to the window shown in green. Note that *all classifiers incorrectly* label this window as *background* (class decisions are not shown). However, the GPC variants do so with a significant amount of uncertainty while the others are inappropriately overconfident. Mission-critical decisions based on overconfident output will lead to catastrophic failure while an appropriately high amount of uncertainty when committing a mistake allows for remedial action to be taken. Providing this more germane output is the introspective quality we seek.

II. Related Works

For a number of years now robots have routinely generated and consumed higher-order abstractions from raw sensor data. Successful applications are as diverse as the detection of ground traversability (e.g. [9]), the detection of lanes for autonomous driving (e.g. [10]), the consideration of classifier output to guide trajectory planning and exploration (see, for example, [11], [12]) or the active disambiguation of human-robot dialogue [13]. These works commonly exploit classification output on a model-trust basis: systems are optimised with respect to precision and recall and egregious misclassifications — including vastly over-confident marginal distributions obtained from some classification frameworks — are accepted as par for the course. However, the suitability of the classification framework employed with respect to its introspective capacity has not previously been considered in robotics. Thus, we consider motivating, defining and investigating introspection in a robotics context to be the primary contribution of our work.

The concept of introspection as introduced here is closely related to considerations in active learning, where uncertainty estimates and model selection steps are often employed to guide data selection and gathering for an incremental learning algorithm. Kapoor *et al.* [14], for example, present an active learning framework for object categorization using a GPC where classifications of large uncertainty (as judged by posterior variance) lead to a query for a ground-truth label and are subsequently used to improve classification performance. Joshi *et al.* [15] address multi-class image classification using SVMs and propose criteria based on entropy and best-versus-second-best measures (see Section III) for disambiguating uncertain classifications. Holub *et al.* [16] propose an information-theoretic criterion that maximises expected information gain with respect to the entire pool of unlabeled data available. Hospedales *et al.* [17] discuss optimising rare class discovery and classification using a combination of generative and discriminative classifiers.

Our treatment of introspection is further informed by an ongoing discussion in the machine learning community regarding how to best account for variance in the space of feasible classifier models when training on, essentially, an incomplete set of data. For example, Tong and Koller [18] present an incremental algorithm for text classification using SVMs and the notion of a *version space*, the set of consistent hyperplanes separating the data in a feature space induced by the kernel function. Zhang et al. [19] introduce a max-margin classifier achieving better generalisation to unseen test data given a limited training corpus. Here, distinctiveness of training instances is assessed using the local classification uncertainty. A global classifier then incorporates these uncertainties as margin constraints, yielding a classifier that places less confusing instances farther away from the global decision boundary. We share the intuition that accounting for variance in version space when selecting a model leads to an increased introspective capacity. As a secondary contribution, therefore, our results serve to further corroborate this intuition.

III. Introspection and Uncertainty

Introspection concerns not the final class decision but rather the confidence with which this decision is made. The concept is motivated by the desire to take appropriate action when a classifier indicates high uncertainty. Our approach to introspection is grounded in the fact that the often cited assumption of independent and identically distributed (*iid*) training and test data is routinely violated in robotics: in the limit of continuous operation in the real world, one-shot classifier training is unlikely to be performed on a complete (or even fully representative) set of data.

Let a classifier map an input $\mathbf{x} \in \mathbb{R}^d$ to one of a set of classes $\mathcal{C} = \{C_1, \ldots, C_{|\mathcal{C}|}\}$ via an associated label $y \in \mathcal{C}$. Prior to training, domain specific knowledge is often used to constrain the family of classification models employed (for example in the form of a kernel, a covariance function or a type of base classifier). Classifier training then involves learning a set of (hyper-) parameters given a training dataset $\{\mathcal{X}, \mathcal{Y}\}$, where $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{|\mathcal{X}|}\}$ denotes the set of feature vectors and $\mathcal{Y}$ denotes the set of corresponding class labels. The training data implicitly give rise to a probability distribution over the set of all possible models within the chosen family, $\mathcal{M}$, such that

$$\{\mathcal{X}, \mathcal{Y}\} \rightarrow p(m \mid \mathcal{X}, \mathcal{Y}), \quad m \in \mathcal{M}. \tag{1}$$

With a slight abuse of notation, m here denotes any member of the family of possible models, $\mathcal{M}$. In reality it is a function of the datum evaluated. In the following we make this relationship explicit by conditioning on both a model (or family of models) as well as on a test datum $\mathbf{x}_*$. Typically, training leads to the selection of a *single* model, $\tilde{m}$ from $\mathcal{M}$ such that a prediction y_* for a new, unseen feature vector $\mathbf{x}_*$ is obtained by approximating

$$p(y_* \mid \mathcal{X}, \mathcal{Y}, \mathbf{x}_*) \approx p(y_* \mid \tilde{m}, \mathbf{x}_*), \quad \tilde{m} \in \mathcal{M}. \tag{2}$$

This is illustrated in Figure 2(a). Common examples of this type of classification framework include SVMs and Boosting classifiers, where an optimisation is performed to select the best model given the training data (see Section IV). The *iid* assumption here is inherent since it is assumed that $\tilde{m}$ remains the best model for all predictions of unseen data. Breaking this assumption therefore often renders the chosen model suboptimal.

An alternative to the single model approach are classification frameworks which take into account the *entire set* of possible models in the specified family, such that

$$p(y_* \mid \mathcal{X}, \mathcal{Y}, \mathbf{x}_*) \approx p(y_* \mid \mathcal{M}, \mathbf{x}_*). \tag{3}$$

This case is illustrated in Figure 2(b). Here the shading indicates the distribution $p(m \mid \mathcal{X}, \mathcal{Y})$ with darker shades indicating increased probability. To aid intuition, predictions of four randomly selected members of $\mathcal{M}$ are also illustrated. Final predictions are made by taking into account opinions from all members of $\mathcal{M}$, often via the computation of an expectation such as for a GPC (see Section IV). Crucially, when considering an expectation over all of $\mathcal{M}$, the increased variance in feasible (and therefore likely) models at a distance from the training data leads to a moderation of the class predictions. This is the introspective quality we seek.

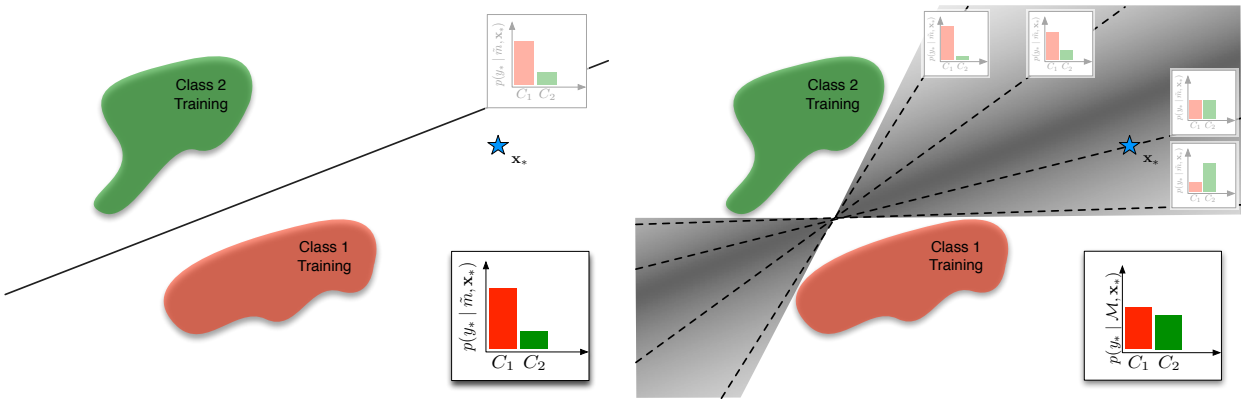


Fig. 2: An illustration of the two types of classification frameworks considered: (a) during training a *single* model is selected to classify an unknown datum $\mathbf{x}_*$ some way removed from the training data; (b) training leads to a distribution over models which is considered entirely to arrive at the final prediction. This illustration is for the family of linear models (indicated by solid (a) and dashed (b) lines). Each predictor is further annotated with its individual prediction. The overall predictive distribution is shown in the bottom right of each subplot. The shading in part (b) indicates the probability weights associated with individual models. Darker regions contain more weight. Note that the overall predictive distribution in (a) stems from the single model used and is, in this case, inappropriately confident. In part (b), however, the overall predictive distribution is moderated by computing the expectation over all models. This gives rise to a much more appropriate uncertainty estimate — the introspective quality we seek. (Best viewed in colour.)

A. *Quantifying Introspection*

In order to characterise the introspective capacity of a classification framework a transferable measure of the inherent uncertainty in the classification output is required. For this purpose, we use an information-theoretic quantity known as normalised entropy, H_N, defined as

$$H_N \quad = \quad -\sum_{C_i \in C} p(y = C_i \mid \mathbf{x}) \log_{|C|} \left[p(y = C_i \mid \mathbf{x}) \right]. \qquad (4)$$

This is equivalent to the Shannon entropy measure normalised by its maximum, which is the entropy of the $|C|$-dimensional uniform distribution, $\log(|C|)$. The result is a measure ranging between 0 and 1 where a *higher* value indicates *greater* uncertainty in the classifier's belief.

An alternative uncertainty measure proposed in the active learning literature is the best-versus-second-best (BvSB) heuristic [15] calculated as the difference between the largest and the second largest class likelihood estimates. This measure attempts to characterise the reliability of the maximum likelihood estimate rather than encoding the shape of the full distribution over class labels. The BvSB and normalised entropy measures are closely related in the binary-classification setting which is the case in this paper. We use normalised entropy throughout the remainder of this work due to its appealing information-theoretic interpretation.

IV. Classification Frameworks

We now present a brief overview of the specific classification frameworks considered in this work: SVMs, LogitBoost classifiers and GPCs. We focus on properties pertinent to introspection. Specifically, we describe the mechanism by which parameters are learned and how probabilistic output is obtained. For simplicity, but without loss of generality, this work considers predominantly binary classification such that $C = \{C_1, C_2\}$. For consistency we adhere to notation commonly found in the literature where a discriminant function is often denoted as $f(\cdot)$. We note that this is equivalent to a particular model m as described in the previous section.

A. *Support Vector Classification*

SVM classification is well established in robotics so that we provide here only an overview[1]. SVMs are based on a linear discriminant framework which aims to maximise the margin between two classes. The model parameters are found by solving a convex optimisation problem, thereby guaranteeing the final classifier to be the best feasible discriminant given the training data. Once training is complete, predictions on future observations are made based on the signed distance of the observed feature vector from the optimal hyperplane, such that

$$f(\mathbf{x}_*) = \sum_{i=1}^{N} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}_*) + b, \qquad (5)$$

where N is the size of the training set, α_i refers to a Lagrange multiplier associated with datum i, b denotes a bias parameter and $k(\mathbf{x}_i, \mathbf{x}_j)$ denotes the kernel function. Both α_i and b are obtained by training, and α_i is then non-zero only for *support vectors* $\mathbf{x}_i$. The kernel function amounts to a scalar product between two data, which have been transformed from d-dimensional feature space into some higher dimensional space. The nature of this mapping between spaces is inherent in the choice of kernel and need not be specified explicitly (the kernel trick). The regularization and kernel parameters are learnt using cross-validation. We discuss our choices of kernel functions in Section IV-D.

In its original form, the SVM classifier output is an uncalibrated real value. A common means of obtaining a probabilistic interpretation is Platt's method [21]. Here, using a hold-out set not used for classifier training, a parametric sigmoid model is fit directly to the class posterior

[1]For a detailed account the reader is referred to, for example, [20].

$p(y_* = C_1 \mid f(\mathbf{x}_*))$, such that

$$p(y_* = C_1 \mid f(\mathbf{x}_*)) \quad = \quad \frac{1}{1 + \exp(af(\mathbf{x}_*) + b)}. \qquad (6)$$

The sigmoid parameters a and b are chosen via cross-validation using a model-trust optimisation procedure. Note that class likelihoods are derived here using only a *single* estimate of the discriminative boundary obtained from the classifier training procedure. No other feasible solutions are considered. Hence, the predictive variance of the discriminant $f(\mathbf{x})$ is not taken into account while determining probabilistic output [22]. Further, no guarantees exist that the optimisation itself is well-behaved[2].

B. LogitBoosting Classifiers

Boosting is a widely used classification framework which involves training an ensemble of weak learners in sequence. The error function used to train a particular weak learner depends on the performance of the previous models [8]. Each weak learner, $h(\mathbf{x})$ is trained using a weighted form of the dataset in which the data weights depend on the performance of the previous classifiers. Predictions from a boosted classifier are obtained using a weighted combination of the individual weak learner outputs such that

$$\mathrm{sgn}(f(\mathbf{x}_*)) \quad = \quad \mathrm{sgn}\left(\sum_{i=1}^{M} w_i h_i(\mathbf{x}_*)\right), \qquad (7)$$

where M is the number of weak learners used.

LogitBoost [24] is a popular choice for a boosting-based classifier as it directly outputs class probability estimates. Weak learners are often chosen to be decision trees and training is conducted by fitting additive logistic regression models by stage-wise optimisation (using Newton steps) of the Bernoulli log-likelihood. The algorithm works in the logistic framework and yields a predictor function $f(\mathbf{x})$ learnt from iterative hypothesis training. Cross-validation is used to set hyper-parameters like the learning rate, tree depth, and the number of boosting rounds. The class-conditional probabilities are obtained from the predictor function as

$$p(y_* = C_1 \mid \mathbf{x}_*) \quad = \quad \frac{\exp(f(\mathbf{x}_*))}{\exp(f(\mathbf{x}_*)) + \exp(-f(\mathbf{x}_*))}. \qquad (8)$$

The procedure possesses asymptotic optimality as a maximum likelihood predictor [24], [25]. However, the method of converting the output of the predictor function to class-conditional probabilities is not fully probabilistic and does not account for variance in the underlying predictor function[3].

C. Gaussian Process Classification

Binary classification using a Gaussian Process (GP) [22], [26] is formulated by first introducing a *latent* function $f(\mathbf{x})$ and then applying a logistic function σ to obtain the prediction $p(y_* = C_1 \mid \mathbf{x}_*) = \sigma(f(\mathbf{x}_*))$. A GP prior is placed on the latent function $f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ characterized by a *mean* function $\mu(\mathbf{x})$ and a *covariance* (or kernel) function $k(\mathbf{x}, \mathbf{x}')$. GPC training establishes values for the hyper-parameters specifying the kernel function k by maximising the log marginal likelihood of the training data.

Probabilistic predictions for a test point are obtained in two steps. First, the distribution over the latent variable corresponding to the test input is obtained using Equation (9). Here, $p(f \mid \mathcal{X}, \mathcal{Y}) = p(\mathcal{Y} \mid f)p(f \mid \mathcal{X})/p(\mathcal{Y} \mid \mathcal{X})$ is the posterior distribution over latent variables.

$$p(f_* \mid \mathcal{X}, \mathcal{Y}, \mathbf{x}_*) = \int p(f_* \mid \mathcal{X}, \mathbf{x}_*, f)p(f \mid \mathcal{X}, \mathcal{Y})df. \qquad (9)$$

This is followed by *marginalising* over the latent f_* to yield the class likelihood $p(y_* = C_1 \mid \mathcal{X}, \mathcal{Y}, \mathbf{x}_*)$ as

$$p(y_* = C_1 \mid \mathcal{X}, \mathcal{Y}, \mathbf{x}_*) = \int \sigma(f_*)p(f_* \mid \mathcal{X}, \mathcal{Y}, \mathbf{x}_*)df_*. \qquad (10)$$

Exact inference is analytically intractable due to the non-Gaussian logistic likelihood function. Instead we leverage expectation propagation (EP) [27], a method widely used for this purpose.

The GPC framework offers two key benefits over the other approaches discussed here [22]. Firstly, the classification output has a clear probabilistic interpretation as it directly results in the class likelihood. In contrast, neither the SVM nor the Boosting framework provide inherently probabilistic output but instead estimate a suitable calibration. Secondly, and crucially, the GP formulation addresses uncertainty or *predictive variance* in the latent function $f(\mathbf{x})$ via *marginalisation* (or averaging) over all models induced by the training set resulting in the estimate $p(y_* = C_1 \mid \mathcal{X}, \mathcal{Y}, \mathbf{x}_*)$ from Equation (10)[4]. Again this is in contrast to the SVM or Boosting estimate $p(y = C_i \mid \hat{f}, \mathbf{x}_*)$ that rely on a single discriminant estimate $\hat{f} : \mathcal{X} \rightarrow \mathcal{Y}$ learnt via minimisation. In the context of introspection, the ability to account for predictive variance is a key advantage of generative classification approaches[5].

D. Kernel Types

Evaluation of the discriminant function for an SVM and the covariance matrix for GPC inference both require the specification of a kernel function, $k(\cdot, \cdot)$. A rich body of literature exists on different choices of kernels for both frameworks. However, since our focus here is on a like-for-like comparison of different classification frameworks we choose two representative kernels rather than performing exhaustive model selection to optimise performance for a particular application. Firstly, as an example of the simplest kernel function available, we consider the linear kernel defined as

$$k_{LIN}(\mathbf{x}_i, \mathbf{x}_j) \quad = \quad \mathbf{x}_i^T \mathbf{x}_j + c, \qquad (11)$$

where c is a constant real number. The linear kernel is an apt choice where a linear separation of the data in feature space leads to adequate performance or where computational

[2]Throughout this work we use LIBSVM [23] for SVM training, calibration and testing.

[3]Throughout this work we use the Matlab implementation of LogitBoost for classifier training and testing.

[4]This process also gives rise to the well known property of increased variance while far away from the data in GP regression.

[5]Throughout this work we use the GPML toolbox [28] for GPC training and testing.

efficiency is of the essence. Often, however, a more sophisticated, non-linear kernel is required. In this category we use the *squared exponential* (SE) function as a canonical representative. The SE kernel with length scale parameter l is defined as

$$k_{SE}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2l^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right). \quad (12)$$

In the context of an SVM, the SE function is more commonly known as a *radial basis function* (RBF). In the following we will adhere to convention and refer to SE GPCs and RBF SVMs.

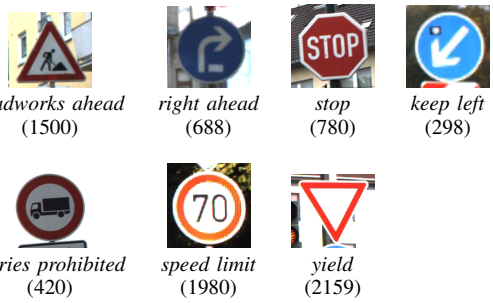
V. Experimental Results

Our experiments investigate the introspective capacity of the classifiers introduced in Section IV in an autonomous driving setting. Specifically, we focus on the *classification* of road signs and the *detection* of traffic lights. In investigating both classification and detection we aim to address the full spectrum of applications commonly encountered in robotics. The two are distinct in that classification addresses the case where a decision is made between two, well-defined classes (e.g. two types of traffic signs) and investigates classifier performance as a third, previously unseen class is presented. The detection case is arguably the more common one in semantic mapping where a single class is separated from a broad (in terms of intra-class variation) *background* class. Here, the concept of a previously unseen class does not exist but the inherent assumption is that the data representing the background class are sufficiently representative to capture any non-class object likely to be encountered. In practice, this is often not the case, leading to a significant number of misclassifications. While it could be argued that this problem is ameliorated somewhat by expanding the dataset used for training, we propose that the complexity of the workspaces encountered during persistent, long-term autonomy will keep perplexing even the most rigorously trained classifier.

A rich body of work on the detection and classification of road signs and traffic lights has established a successful track record of template-based features for this purpose. Specifically, we leverage the approach proposed by Torralba *et al.* [29] in which a dictionary of partial templates is constructed, against which test instances are matched. A single feature consists of an image patch (ranging in size from 8×8 to 14×14 pixels) and its location within the object as indicated by a binary mask (32×32 pixels). For any given test instance, the normalised cross-correlation is computed for each feature in the dictionary. Therefore, per instance (or window, in the detection case) a feature vector of length d is obtained, where d is the size of the dictionary. We found empirically that $d > 200$ leads to negligible performance increase in classification. Throughout our experiments we therefore set $d = 200$.

A. Introspection in Classification

This section investigates classification output when a third, previously unseen class is presented to the classifier. As examples of classes typically encountered in autonomous driving applications we use a subset of the German Traffic Sign Benchmark (GTSRB) dataset [30], which comprises over 50,000 loosely-cropped images of 42 classes of road signs, with associated bounding boxes and class labels. From this dataset we specifically focus on the seven classes shown in Table I. We arbitrarily select two classes for training: *stop* and *lorries prohibited*. To investigate the efficacy of the features used and training procedures employed, classifiers were trained separating these two classes using a balanced training set of 400 data (200 per class) and applying a canonical training procedure for each classifier type, including five-fold cross-validation where appropriate. Classifier performance was evaluated using standard metrics on a hold-out set of another 400 class instances (200 of each class) of the same two classes. The results are shown in Table II. Classification performance is commensurate across all classifiers. The corresponding precision-recall curve confirms the near-perfect separation of the classes and has been omitted here as it is otherwise uninformative. The classifiers are next retrained using the full 800 training data (400 per class) and the same canonical training procedures. They are then applied to 500 instances of the previously unseen class *roadworks ahead*. The resulting normalised entropy histograms are shown in Figure 3. The mean normalised entropies for the GPC-based classifiers are significantly higher than those of the other classification frameworks, indicating that the the GPC-based classifiers exhibit greater uncertainty in their judgement. Conversely, the RBF SVM and the LogitBoost classifier are extremely confident in their classifications with a very narrow distribution around a relatively low value of normalised entropy. This was an effect consistently observed throughout our experiments, which we attribute to the relatively gradual decay of the estimated class posterior probabilities through feature space often encountered far away from the decision boundary. Features from an unseen class which are located in feature space at a distance from the decision boundary



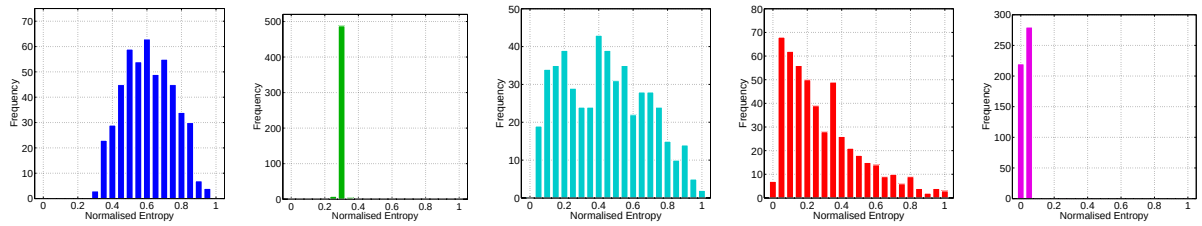
roadworks ahead (1500) | *right ahead* (688) | *stop* (780) | *keep left* (298)

lorries prohibited (420) | *speed limit* (1980) | *yield* (2159)

TABLE I: The seven classes of the German Traffic Sign Recognition Benchmark dataset considered in our work. The numbers in brackets indicate the number of data available per class.

Classifier	Precision	Recall	F_1
SE GPC	1.000	0.990	0.995
RBF SVM	1.000	0.995	0.997
Linear GPC	1.000	0.990	0.995
Linear SVM	1.000	0.990	0.995
LogitBoost	1.000	0.965	0.982

TABLE II: Classification performance when separating *stop* sign from the *lorries prohibited* signs. Note that different class combinations were found to yield classifiers of similar quality.



(a) SE GPC (b) RBF SVM (c) Linear GPC (d) Linear SVM (e) LogitBoost

Fig. 3: Normalised entropy histograms of the marginal probabilities for five classifiers trained on the road sign classes *stop* and *lorries prohibited* and tested on 500 instances of the unseen class *roadworks ahead*. Higher normalised entropy implies more uncertainty in classifier output. Note that the mean normalised entropy for the SE GPC is higher than that of the others.

Test Class	Classifier	Normalised Entropy	
		$\mu\pm$ std. err.	$\sigma\pm$ std. err.
	SE GPC	**0.504 ± 1.92E-03**	0.110 ± 9.35E-05
	RBF SVM	0.313 ± 1.33E-04	0.012 ± 2.12E-06
	Lin GPC	0.245 ± 9.34E-04	0.173 ± 9.19E-05
	Lin SVM	0.106 ± 4.77E-04	0.107 ± 2.51E-04
	Logit	0.015 ± 2.72E-05	0.009 ± 4.11E-05
	SE GPC	**0.487 ± 1.70E-03**	0.139 ± 7.67E-05
	RBF SVM	0.310 ± 1.13E-04	0.017 ± 3.72E-06
	Lin GPC	0.286 ± 8.09E-04	0.179 ± 6.24E-05
	Lin SVM	0.076 ± 3.72E-04	0.097 ± 2.43E-04
	Logit	0.012 ± 1.79E-05	0.007 ± 1.27E-05
	SE GPC	**0.723 ± 4.91E-04**	0.186 ± 1.59E-04
	RBF SVM	0.306 ± 1.03E-04	0.095 ± 7.96E-05
	Lin GPC	0.680 ± 4.75E-04	0.235 ± 1.11E-04
	Lin SVM	0.634 ± 7.29E-04	0.267 ± 4.28E-05
	Logit	0.021 ± 1.07E-04	0.031 ± 7.10E-04
	SE GPC	0.804 ± 6.08E-04	0.163 ± 1.72E-04
	RBF SVM	0.335 ± 1.43E-04	0.050 ± 1.26E-05
	Lin GPC	**0.811 ± 4.39E-04**	0.184 ± 1.66E-04
	Lin SVM	0.642 ± 3.24E-04	0.294 ± 9.19E-05
	Logit	0.017 ± 3.62E-05	0.018 ± 2.06E-04
	SE GPC	**0.259 ± 2.36E-03**	0.116 ± 1.37E-04
	RBF SVM	0.255 ± 1.28E-04	0.027 ± 5.26E-06
	Lin GPC	0.155 ± 9.27E-04	0.140 ± 2.61E-04
	Lin SVM	0.043 ± 7.82E-05	0.059 ± 1.26E-04
	Logit	0.007 ± 1.29E-07	0.007 ± 2.31E-05

TABLE III: Mean and standard deviation normalised entropies (including standard errors) from ten iterations of classifier training and testing, each with a randomly created dictionary and both training and test datasets resampled. Results are presented for classifiers trained on the road sign classes *stop* and *lorries prohibited* and tested on five different unseen classes as shown.

therefore only span a very narrow range of estimated class posterior probabilities.

In order to mitigate any influences of the specific feature set used and the specific training and test data selected we repeated the above experiment across a number of random dictionaries, data samples and unseen classes. Specifically, for each of five different unseen classes, we perform ten iterations of classifier training and testing with a random dictionary and training and test datasets resampled for each run. The results, presented in Table III, are consistent with those in Figure 3 in that the GPCs tend to be more uncertain while SVM and LogitBoost are more confident with an often significantly narrower distribution of normalised entropy values. The results in Table III indicate that the gap in uncertainty between the different frameworks is more pronounced for some unseen classes than for others. We attribute this to the varying degree of similarity in feature space between the unseen class and the classes in the training set. A more in-depth analysis of this phenomenon remains future work.

Classifier	Precision	Recall	F_1
SE GPC	0.976	0.909	0.941
RBF SVM	0.982	0.931	0.956
Linear GPC	0.970	0.912	0.940
Linear SVM	0.979	0.929	0.953
LogitBoost	0.963	0.928	0.945

TABLE IV: Performance on a holdout set of 2000 instances of classifiers trained on data from the TLR data set.

B. Introspection in Detection

We investigate the same classification frameworks as before on the task of traffic light detection. To this end we use the Traffic Lights Recognition (TLR) dataset [31], which is a sequence of colour images taken by a monocular camera from a car driving through central Paris. The TLR dataset comprises of just over 11,000 frames, where most of the traffic lights have been labeled with bounding boxes and further metadata such as the status of the signal or whether a particular label is ambiguous (e.g. the image suffers from motion blur, the scale is inappropriate or a traffic light is facing the wrong way). A few traffic lights have been omitted altogether. As suggested by the authors of [31], we exclude from our experiments any labels of class *ambiguous* or *yellow signal* and any instances which are partially occluded. We also remove any section of the sequence where the car is stationary and the lights are not changing. We split the dataset into two parts (at frame 7,200 of 11,178), with an approximately equal number of remaining labels in each part and with no physical traffic lights in common. Positive data are extracted as labeled. Negative *background* data are extracted by sampling patches of random size and position from scenes in the dataset while ensuring that the patches do not overlap with positive instances. The data are then split into training and test sets and classifiers are trained as before.

Again, we first verify the efficacy of the features selected and the training procedures employed. Table IV shows the classification performance for classifiers trained on 1,000 examples and evaluated on a hold-out set of 2,000 data. For completeness, Figure 4 shows the corresponding precision-recall curve. As before, classification performance according to conventional metrics is commensurate across all frameworks. In Figure 5, however, we demonstrate how the lack of introspection can impact classification performance when accept/reject decisions are guided by classification confidence. Specifically, we show the *cumulative* effect of accepting classifications below a given uncertainty threshold. First we note that when classifications are accepted at any level of uncertainty (i.e. up to and including unity normalised

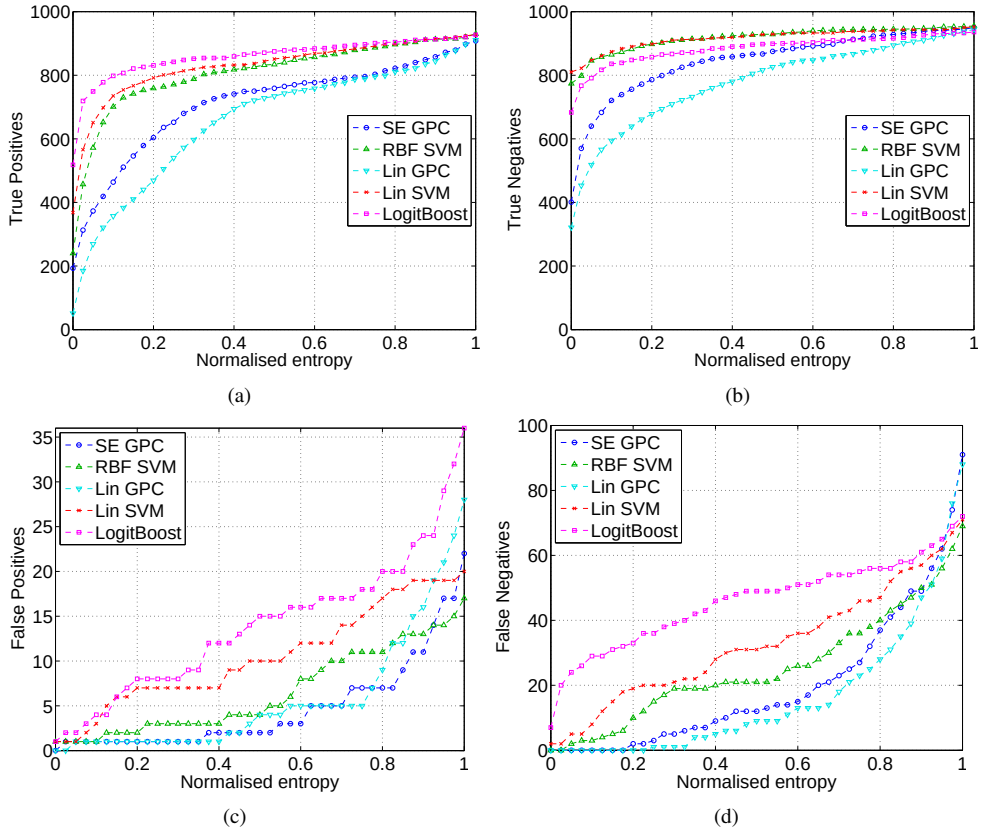


Fig. 5: Cumulative frequency plots of classification confusion (true positives, true negatives, false positives, and false negatives) against normalised entropy. The classifiers have been trained on 500 traffic lights against 500 background patches, and tested on 1,000 instances of each. Note that lower normalised entropy implies more certainty in classification. A more introspective classifier is one that exhibits higher uncertainty (as witnessed by larger normalised entropy in its output) when processing difficult instances. Consequently, class decisions on output above a given normalised entropy threshold are deferred since the output is deemed ambiguous. This is desirable since a single bad decision can have disastrous consequences. (Best viewed in colour.)

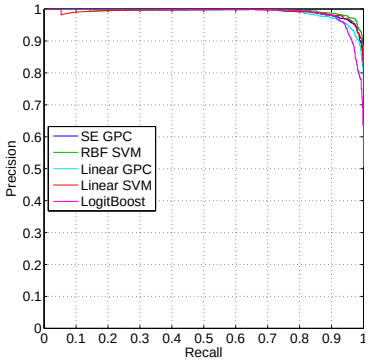


Fig. 4: Precision-recall graph for traffic light detection. Classifier performance is commensurate for all frameworks. (Best viewed in colour.)

entropy) all classification frameworks are commensurate in terms of true positives and true negatives (top row of Figure 5). This further corroborates the accuracy figures in Table IV. However, true positive and negative classifications occur generally at higher certainty (i.e. as normalised entropy tends to zero) for SVMs and LogitBoost classifiers than for the GPC variants. The latter are overall less certain about a significant number of correct classifications. The bottom row of Figure 5 indicates that SVMs and LogitBoost classifiers are also significantly more confident when *misclassifying* data (an example of this is also shown in Fig. 1). Significant numbers of mistakes are made at relatively low normalised entropy thresholds. The GPC variants, in contrast, accumulate comparable numbers of classification errors only at higher normalised entropy thresholds. The price paid for this more realistic assessment of the classification output is a reduction in correct classifications above the normalised entropy threshold. Note that this does not mean that subsequent samples are misclassified. It only implies that some other remedial action might be taken — for example obtaining label confirmation from a human or gathering otherwise additional data to aid disambiguation.

VI. Conclusions

This work demonstrates how performance metrics traditionally used in machine learning for classifier training and evaluation may be insufficient to characterise system performance in a robotics context, where a single misjudgement can have disastrous consequences. To remedy this shortcoming, we propose the concept of *introspection*: the ability to mitigate potentially overconfident classifications by a realistic assessment of predictive variance. Our experimental results imply that, despite commensurate performance as measured by more conventional metrics, GPCs possess a more pronounced introspective capacity than other classification frameworks commonly employed in robotics. We attribute this to their accounting, at test time, for predictive variance over the space of feasible classification models. This is in contrast to other commonly employed classification frameworks which often only consider a one-shot (ML or

MAP) solution. GPCs appear therefore better suited than the other frameworks investigated to applications where a realistic assessment of classification accuracy is required. Crucially, this includes many decision-making problems commonly encountered in robotics.

We have not, at this stage, considered the computational complexity of the approaches presented. Though GPCs in their basic form are notoriously expensive, more elaborate schemes exist which reduce the computational burden required for GPC inference. Our future work will investigate a variety of these schemes for suitability for real-time performance in autonomous driving tasks. Our work also holds implications for robotic active learning and exploration, which opens up additional avenues of research we intend to explore.

VII. Acknowledgements

This work is funded under the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement Number 269916 and by the UK EPSRC Grant Number EP/J012017/1. We gratefully acknowledge advice from Ian Baldwin on parameter selection for boosting trees in Matlab. We thank Prof. Paul Newman for his support and encouragement for this work.

References

[1] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Y. Ng, "Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data." in *CVPR (2)*, 2005, pp. 169–176.

[2] O. Ḿartínez-Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, "Supervised semantic labeling of places using information extracted from sensor data," *Robot. Auton. Syst.*, vol. 55, no. 5, pp. 391–402, 2007.

[3] I. Posner, M. Cummins, and P. Newman, "A generative framework for fast urban labeling using spatial and temporal context," *Autonomous Robots*, 2009. [Online]. Available: http://dx.doi.org/10.1007/s10514-009-9110-6

[4] B. Douillard, D. Fox, and F. Ramos, "Laser and vision based outdoor object mapping," in *Proceedings of Robotics: Science and Systems IV*, Zurich, Switzerland, June 2008.

[5] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *Intl. Conf. on Robotics and Automation*, 2012, pp. 3515 –3522.

[6] S. Sengupta, P. Sturgess, L. Ladick, and P. H. S. Torr, "Automatic dense visual semantic mapping from street-level imagery," in *Robotics and Autonomous Systems, IEEE International Conference on*, 2012.

[7] R. Paul, R. Triebel, D. Rus, and P. Newman, "Semantic categorization of outdoor scenes with uncertainty estimates using multi-class Gaussian process classification," in *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, 2012, to appear.

[8] C. Bishop, *Pattern recognition and machine learning*. springer New York, 2006, vol. 4.

[9] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the DARPA Grand Challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[10] A. Huang and S. Teller, "Probabilistic Lane Estimation using Basis Curves," in *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[11] D. Meger, P. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. Little, and D. Lowe, "Curious george: An attentive semantic robot," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 503–511, 2008.

[12] J. Velez, G. Hemann, A. Huang, I. Posner, and N. Roy, "Planning to perceive: Exploiting mobility for robust object detection," in *Proc. ICAPS*, 2011.

[13] S. Tellex, P. Thaker, R. Deits, T. Kollar, and N. Roy, "Toward information theoretic human-robot dialog," in *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[14] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Gaussian processes for object categorization," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 169–188, 2010.

[15] A. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009, pp. 2372–2379.

[16] A. Holub, P. Perona, and M. Burl, "Entropy-based active learning for object recognition," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. IEEE, 2008, pp. 1–8.

[17] T. Hospedales, S. Gong, and T. Xiang, "Finding rare classes: Active learning with generative and discriminative models," *Knowledge and Data Engineering, IEEE Transactions on*, no. 99, pp. 1–1, 2011.

[18] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *The Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[19] W. Zhang, X. Stella, and Y. S. Teng, "Power svm: Generalization with exemplar classication uncertainty," in *Computer Vision and Pattern Recognition, 2012. CVPR 2012. IEEE Conference on*. IEEE, 2012.

[20] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[21] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances In Large Margin Classifiers*. MIT Press, 1999, pp. 61–74.

[22] C. Rasmussen and C. Williams, "Gaussian processes for machine learning. 2006," *The MIT Press, Cambridge, MA, USA*.

[23] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[24] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Annals of Statistics*, vol. 28, p. 2000, 1998.

[25] T. Hastie and R. Tibshirani, *Generalized additive models*. Chapman & Hall/CRC, 1990.

[26] C. Williams and D. Barber, "Bayesian classification with gaussian processes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 12, pp. 1342–1351, 1998.

[27] T. Minka, "Expectation propagation for approximate bayesian inference," in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2001, pp. 362–369.

[28] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (GPML) toolbox," *J. Mach. Learn. Res.*, vol. 11, pp. 3011–3015, Dec. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1953029

[29] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing visual features for multiclass and multiview object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 854–869, May 2007.

[30] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, no. 0, pp. –, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608012000457

[31] R. C. of Mines ParisTech, "Traffic lights recognition (TLR) data set," http://www.lara.prd.fr/benchmarks/trafficlightsrecognition.