

# Introspective Classification for Robot Perception

Hugo Grimmerud Rudolph Triebel Rohan Paul Ingmar Posner

**Abstract**—In robotics, the use of a classification framework which produces scores with inappropriate confidences will ultimately lead to the robot making dangerous decisions. In order to select a framework which will make the best decisions, we should pay careful attention to the ways in which it generates scores. *Precision* and *recall* have been widely adopted as canonical metrics to quantify the performance of learning algorithms, but for robotics applications involving mission-critical decision making, good performance in relation to these metrics is insufficient. We introduce and motivate the importance of a classifier’s *introspective* capacity: the ability to associate an appropriate assessment of confidence with any test case. We propose that a key ingredient for introspection is a framework’s potential to increase its uncertainty with the distance between a test datum its training data.

We compare the introspective capacities of a number of commonly used classification frameworks in both classification and detection tasks, and show that better introspection leads to improved decision-making in the context of tasks such as autonomous driving or semantic map generation.

## I. INTRODUCTION

In robotics, the outputs of our classification frameworks are intended to be used to make decisions. We want the output of a classifier to help the robot decide whether to stop at a traffic light, whether to slow down in front of a pedestrian, or how to populate a semantic map. The processes by which we go from data to decision must be very carefully examined not least when our robots’ behaviours can impact the safety of humans sharing their workspace.

A common way to improve a robot’s interactions is to give it prior information in the form of a semantic map, informing the robot about how its environment behaves and how it can interact with it. In almost all safety-critical applications these maps are hand-made, because the current state-of-the-art solutions to automatic mapping are not robust enough to ensure the high quality of maps required for safe, autonomous robot operation.

Following classical decision theory, in situations where a poor choice of action can incur a large cost (e.g. driving forwards into another vehicle, or incorrectly placing a particular semantic label in a map), a robot will only choose that action if its classifier is supremely certain. In practice, we see that commonly-used classification frameworks can assign extremely high certainty or confidence to classifications which turn out to be incorrect. In order to avoid these large costs, it follows that when our classifiers make mistakes they should do so only with high uncertainty.

For example, a classification error can occur when the classifier is presented with a test datum which is unlike anything it saw during training. This could be as a result of the training set not containing the true class of this test datum, or because it is a new viewpoint of an existing class.

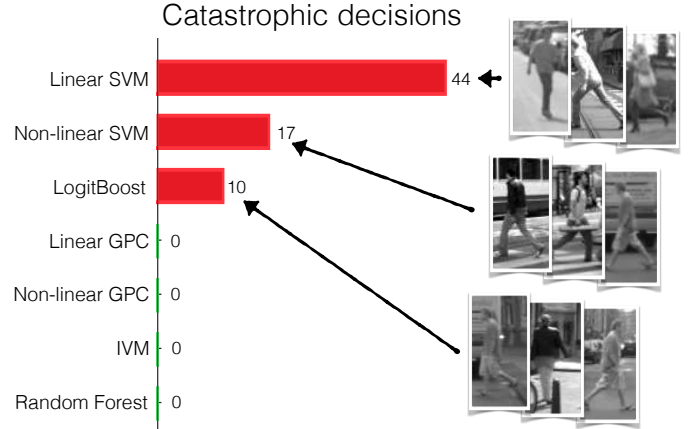


Fig. 1:

In the context of autonomous driving, the same parked car can appear very differently given changing weather or time of day. These new and unusual test data are common in practice since the training data can never be fully representative of the continually evolving and complex environments in which our robots operate. In this situation, we argue that the appropriate response is for our classifier to respond with high uncertainty.

There is a tendency to choose one particular classification framework over another based on the standard metrics of classification: precision and recall. Here we show that these are insufficient to characterise whether a classifier will provide appropriate uncertainties. Without these appropriate uncertainties, our robots are doomed to make costly and potentially catastrophic decisions.

Therefore, rather than using a classifier which makes correct and incorrect decisions with similarly high confidence, it is preferable to use a classifier which makes mistakes only with high uncertainty, and correct classifications with high certainty. Hence, we seek classifiers with the capacity to adjust the confidence of a particular classification on the basis of how ‘qualified’ they are given their own prior knowledge, embodied by their training data. If a classification framework leads to an overconfident estimate of the class label, then the entire decision making process may be ineffective. Our work investigates this *introspective* capacity in a number of classification frameworks commonly used in robotics: the Support Vector Machine (SVM), LogitBoost, the Random Forest, the Gaussian Process Classifiers (GPC), and the Informative Vector Machine (IVM). We use the term ‘introspective’ to describe a classifier that gives appropriate probabilities, thus making true classifications with confidence and makes mistakes only with high uncertainty.

We carefully examine how these classification frameworks

use *distance* between training and test data to moderate the confidence in a classification. Intuitively, a test datum which is far away in feature space from the training data is more likely to be misclassified than one which lies in the middle of a dense cluster from one class, and thus the classification should be made with greater uncertainty. Most classification frameworks make use of a model, such that instead of calculating a distance from the test datum to all the training data, they calculate a distance between the test datum and the model (which can be further affected by the use of a kernel, effectively warping the feature space). Therefore, the choices of model and measure of distance greatly affect the uncertainty with which classifications are made. Some frameworks consider one single discriminant to separate the classes, while others average over a variety of possible discriminants. The results we present indicate that the latter tends to be characteristic of classifiers with a better sense of introspection, as a result of their ability to predict the variance from the responses of the individual discriminants for a test datum.

The key contributions of this work are:

- The concept of a classifier’s introspective quality, regarding how it expresses the relevance of its prior information when making detections,
- A comparison of how commonly-used classification frameworks generate probabilities, and insights regarding whether they are likely to display introspective qualities from a theoretic standpoint,
- Results to show the introspective behaviour of those classification frameworks when applied to tasks commonly tackled in robotics, such as classification and detection, and
- The further application of those classifiers to decision-making problems, and results which indicate that introspective classification leads to better decision-making. This motivates the opinion that considering the introspective quality of classification frameworks is critical in robotics.
- All of the above are evaluated using publicly-available data sets which are relevant to mobile robotics.

Some of this work has appeared in Grimm et al. [2013]. Here we present a more detailed treatment of the concepts and substantial evaluation on two additional publicly available data sets, along with the important implications of introspection in terms of decision making.

We start by offering a theoretical insight into why some classification frameworks may exhibit greater introspective qualities than others. We do so by examining the methods by which commonly used algorithms generate probabilities (Section III), and specifically detailing the key methods for the classifiers we are comparing (Section IV). Then we demonstrate the various behaviours of those classifiers in several scenarios related to autonomous driving. We consider the similar but nuanced cases of *classification* (estimating the likelihood that an image contains one particular class of object over another, Section V-C) and *detection* (estimating the likelihood that an image contains one privileged class of object over a background class comprising all other classes,

Section V-D). Finally, we demonstrate the behaviour of the classifiers in terms of decision making (in Section V-E). As we vary the relative costs of false positive and false negative errors,

## II. RELATED WORKS

For a number of years now robots have routinely consumed higher-order abstractions from raw sensor data. Successful applications are as diverse as the detection of ground traversability (e.g. Thrun et al. [2006]), the detection of lanes for autonomous driving (e.g. Huang and Teller [2010]), the consideration of classifier output to guide trajectory planning and exploration (see, for example, Meger et al. [2008], Velez et al. [2011]) or the active disambiguation of human-robot dialogue [Tellex et al., 2012]. 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. [2010], for example, present an active learning framework for object categorisation using a GPC where classifications with 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. [2009] address multi-class image classification using SVMs and propose criteria based on entropy and best-versus-second-best measures (see Section III-B) for disambiguating uncertain classifications. Holub et al. [2008] propose an information-theoretic criterion that maximises expected information gain with respect to the entire pool of unlabelled data available. Hospedales et al. [2013] discuss optimising rare class discovery and classification using a combination of generative and discriminative classifiers. In the related field of reinforcement learning, the authors of Li et al. [2008] present a general framework which determines whether enough labelled data have been provided to constrain certain problems. If the learners space of solutions is insufficiently constrained such that its output cannot be guaranteed to be within  $\epsilon$  of the true solution with probability  $1 - \delta$ , it asks for more labelled data. This accuracy guarantee is same for both false positive and false negative errors, and thus the framework is not appropriate for situations in which costs associated with those errors are imbalanced. In the context of autonomous systems, the costs are commonly imbalanced.

Our treatment of introspection is further informed by an ongoing discussion in the machine learning community regarding how best to account for variance in the space of

feasible classifier models when training on, essentially, an incomplete set of data. For example, Tong and Koller [2002] 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. [2012] 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.

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, Anguelov et al. [2005], Martínez-Mozos et al. [2007], Posner et al. [2009], Douillard et al. [2008], Pronobis and Jensfelt [2012], Sengupta et al. [2012], Paul et al. [2012]). We consider introspection to be paramount to reducing the human effort required to automatically generate semantic maps which we can then use for autonomous operation.

Niculescu-Mizil and Caruana [2005] recognise that the question of whether the probabilities produced by various classification frameworks are appropriate is important, a sentiment we clearly share. They conclude that poorly-calibrated frameworks (in a probabilistic sense) can be effectively corrected using an additional learned calibration using either Platt’s method or isotonic regression. They find Random Forests to perform well pre-calibration (although they did exhibit a tendency to be under confident, consistently with our findings), and that SVMs perform well after post-calibration. They associate the need for further calibration specifically to the classifiers using max-margin optimisation, rather than the treatment of distances in feature space and the distribution of models over version space, as we do. They also do not explore the effects of making decisions using these probabilities.

Berczi et al. [To appear 2015] have confirmed the introspective power of GPCs over SVMs, employing them to avoid areas of terrain for which the height may be misclassified.

### III. INTROSPECTION, UNCERTAINTY, AND DECISION MAKING

In this section we first describe a crucial property we expect classification frameworks to require in order to be introspective: marginalisation over possible models (Section III-A). Then we describe some measures of uncertainty, motivating the use of normalised entropy as the most appropriate measure (Section III-B). We finish by describing a manner in which to obtain decisions from probabilistic classification results, and motivate the practice of choosing outcome costs directly rather than adjusting thresholds to modify a robot’s behaviour (Section III-C).

#### A. Introspective Capacity

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 rarely 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  $C = \{C_1, \dots, C_c\}$  via an associated label  $y \in \{1, \dots, c\}$ , where  $c$  is the number of classes. 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 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  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  denotes the set of  $N$  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 (or *hypotheses*) within the chosen family,  $\mathcal{M}$ , such that

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

With a slight abuse of notation,  $m$  here denotes any member of the family of possible models,  $\mathcal{M}$ . 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

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

This is illustrated in Figure 2a. 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 conditioned on the training data, such that

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

This case is illustrated in Figure 2b. Here the shading indicates the distribution  $p(m | \mathcal{X}, \mathcal{Y})$  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.

Between the two extremes lies the Random Forest, which chooses a number of differing samples from  $\mathcal{M}$ , and averaging over the responses from these.

We believe that this marginalisation over plausible models in version space is a key component of an introspective classifier.

### B. Characterising Uncertainty

In order to characterise the introspective capacity of a classification framework, a well-tempered 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

$$H_N = -\sum_{i=1}^c p(y = C_i | \mathbf{x}) \log_c [p(y = C_i | \mathbf{x})]. \quad (4)$$

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, as shown by the blue curve in Figure 3.

An alternative uncertainty measure proposed in the active learning literature is the best-versus-second-best (BvSB) heuristic [Joshi et al., 2009] which equals 1 minus the difference between the largest and the second largest class likelihood estimates, as shown by the red curve in Figure 3. 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 that of this paper. We use normalised entropy throughout the remainder of this work for two reasons: firstly, it is formed from an information-theoretic point of view, compared to BvSB which is an ad-hoc heuristic; secondly, in multi-class settings it considers the entire distribution over classes, rather than BvSB which only takes into account only the two classes with the highest probability.

### C. Decision Making

Autonomous robots typically have at their disposal a set of actions, each of which is appropriate to particular situations. The difficulty lies in choosing which action to perform when there is uncertainty about the state of the world. Following standard decision theory [LaValle, 2006], we calculate the expected loss of performing a particular action when we have a set of likelihoods for each state of the world ( $p(C_1), p(C_2), \dots, p(C_{|C|})$ ), defined as:

$$\bar{L}(a) = \sum_{i=1}^{|C|} L(a, C_i) p(C_i), \quad (5)$$

where  $L(a, C_i)$  is the cost or loss associated with each potential outcome. We then choose to perform the action  $a$  which minimises this expected loss.

For our decision-making experiments, we later consider a simple scenario in which there are two states the world can be in: either there is an object in the way ( $C_2$ , e.g. a pedestrian, car, or traffic light), or there is not ( $C_1$ ). There are also two available actions  $a \in \{\text{stop}, \text{go}\}$ . We wish our robot to *stop* if there is an object in its path, or *go* if the way is clear. The losses will vary from application to application, but in the case of autonomous driving it is sensible to associate a very high cost to performing the *go* action when there is in fact an object in the way ( $C_2$ ), resulting in a collision, and a lesser cost to performing the action *stop* when the path is clear ( $C_1$ ), resulting in an unnecessary delay. While inefficient, this false positive error is more desirable than running a red light or colliding with another vehicle. Interestingly, in the case of driver assistance systems (e.g. automatic emergency braking) the costs are reversed: the loss associated with a false positive (an un-necessary emergency stop) is very large, and a false negative (a missed opportunity to perform an emergency stop) is a less undesirable outcome. In Figure 4a we show the expected losses of the two actions when there is equal cost associated with each type of error. We can see that the intersection between the two lines occurs at  $p(C_1) = p(C_2) = 0.5$ . As we increase the cost of a false negative (performing the *go* action when there is a person,  $C_2$ ), the range of detection probabilities  $p(C_2)$  which result in a *go* action reduces, as seen in Figure 4b. Since an introspective classifier is uncertain when it makes mistakes, the errors will be close to  $p(C_2) = 0.5$  as the teal distribution in Figure 4c, and so those errors will largely be subsumed by the *stop* action. A less introspective classifier will make more mistakes near the extremes of the class marginal spectrum (purple in Figure 4c) and so more of those errors will occur in the *go* region, resulting in a greater prevalence of very expensive errors.

Thus ideally, as we make the cost of a false negative much greater than that of a false positive, our classifiers become more and more cautious, employing safer actions and incurring less overall cost. Crucially, this relies upon the assumption that most of a classifier’s mistakes lie in the middle of the probability spectrum.

Another way to characterise the desirable introspective property is to consider the proportion of the errors contained in some window around  $p(C_2) = 0.5$ , represented by the orange box in Figure 4d. As we increase the half-length of the box from 0 to 0.5, we would like the proportion of errors to increase quickly, and then stabilise as we encompass the regions of high confidence, represented by the teal curve in Figure 4e. A less introspective classifier would have errors near  $p(C_2) = 0$  or  $p(C_2) = 1$ , and so the contained errors would resemble the purple curve in Figure 4e. We will present curves resembling these for each classifier from real data in Section V-E.

Often, the temptation is to tune the costs to steer the robot towards the ‘desired’ behaviour. Instead, we ought to focus on whether the costs are correct (because it is usually easier to quantify these than a probability threshold) and allow the decision theory to choose the behaviour which is true to the spirit of the cost function. This is only possible if the probabilities and uncertainties supplied by the classification

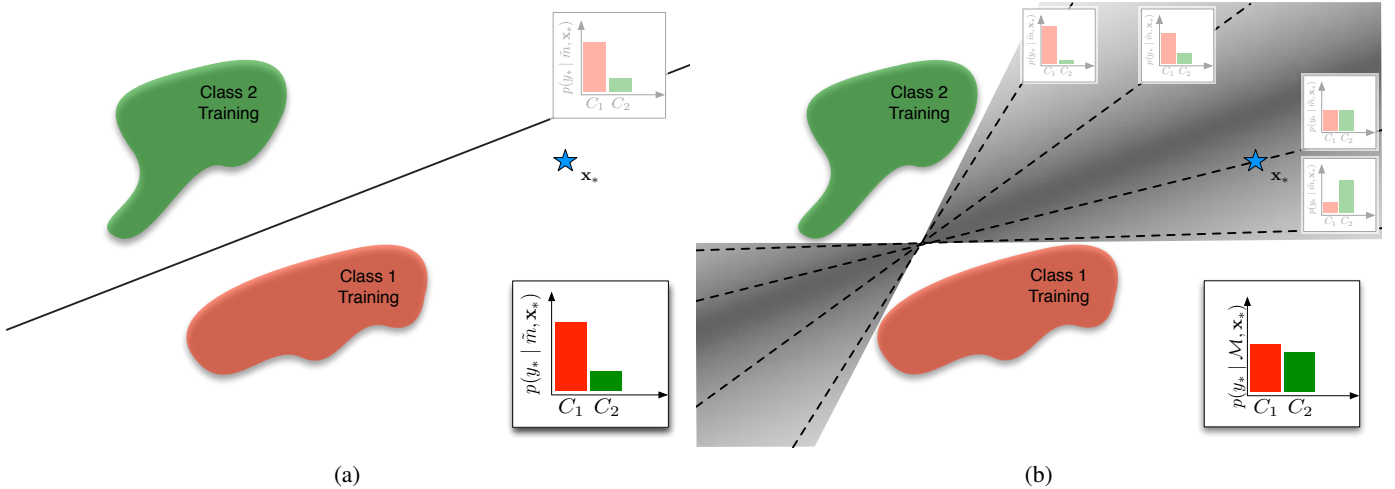


Fig. 2: An illustration of the two types of classification frameworks considered: (a) during training a *single* model is selected to classify an unknown datum  $\mathbf{x}_*$  some way removed from the training data; (b) training leads to a distribution over models which is considered entirely to arrive 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.)

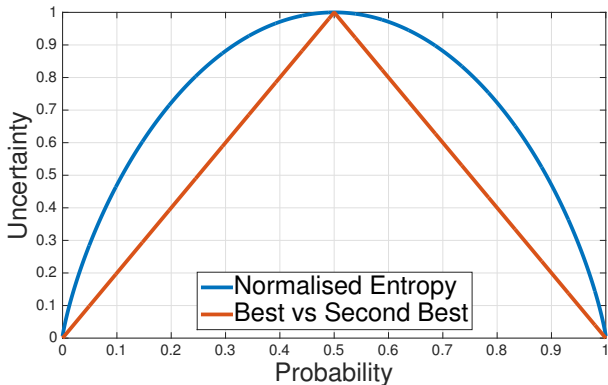


Fig. 3: Normalised entropy and best-versus-second-best as measures of uncertainty in the binary classification case.

frameworks are sensible.

#### IV. CLASSIFICATION FRAMEWORKS

We now present a brief overview of the specific classification frameworks considered in this work: SVMs, LogitBoost, the Random Forest, GPCs, and the IVM. The implementations of these are all off-the-shelf, using popular libraries detailed in each subsection. The goal is not to find the most accurate classifier, but rather to examine the consistency of the confidences with which certain decisions are made. We believe that this consistency in choosing the appropriate decision given the potential losses is an often ignored and paramount characteristic of classification frameworks, and that in the context of safety-critical tasks it could be worth accepting a decrease in accuracy if it results in an increased introspective

consistency. In the following descriptions of the frameworks we focus on properties pertinent to introspection, specifically how the use of distance between data affects the classification confidence, and what type of models they use. 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. For a detailed account the reader is referred to, for example, Burges [1998]. 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, defined by the weight vector  $\mathbf{w}$  and bias  $w_0$ , such that

$$f(\mathbf{x}_*) = \mathbf{w}^\top \phi(\mathbf{x}_*) + w_0 = \sum_{i=1}^N \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}_*) + w_0, \quad (6)$$

where  $N$  is the size of the training set,  $\alpha_i$  refers to a Lagrange multiplier associated with datum  $i$ ,  $w_0$  denotes a bias parameter,  $\phi$  refers to the feature map, and  $k(\mathbf{x}_i, \mathbf{x}_j)$  denotes the kernel function.

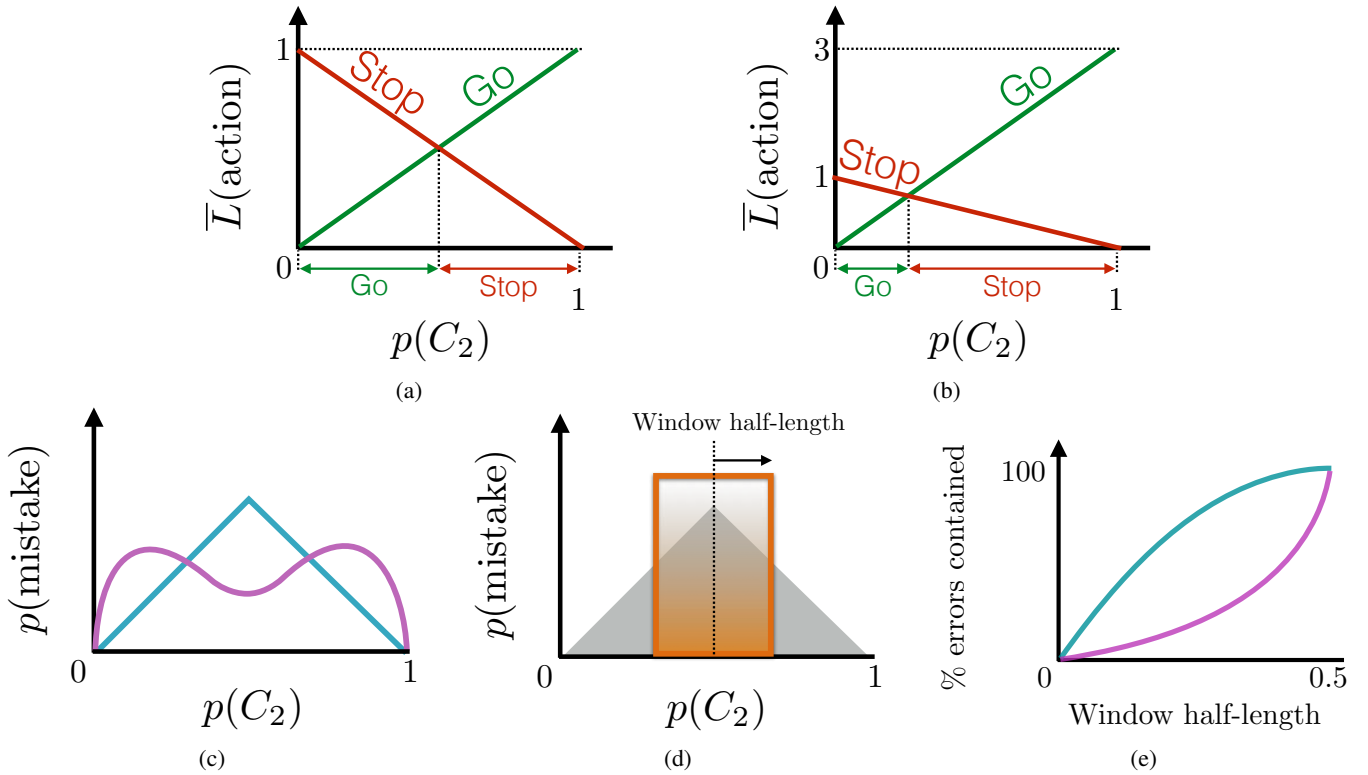


Fig. 4: (a) We have set equal cost to a false positive (take the *stop* action when there is, in fact, no person:  $C_1$ ) and the false negative (take the *go* action when there is a person:  $C_2$ ). The expected losses from the two actions meet at  $p(C_1) = p(C_2) = 0.5$ , and choose the action which minimises the expected loss.

(b) We have made the cost of a false negative three times the cost of a false positive, which reduces the probability region for which we choose the *go* action. By increasing the cost of accidentally hitting a pedestrian, we are trying to create a more cautious system, which will take the *stop* action more of the time.

(c) A more introspective classifier (teal) will make most of its mistakes with high uncertainty, when  $p(C_2)$  is near 0.5. Less introspective classifiers (purple) will make mistakes with low uncertainty.

(d) As we grow the orange box outward from the centre, we can calculate how many errors are contained for a particular distribution in (c).

(e) We show the result of plotting the number of errors contained as we grow the orange box for the two idealised classifiers in (c). The teal (more introspective) classifier catches more errors when the box is small than for the purple (less introspective) classifier. It also reaches steady-state because there are very few errors around  $p(C_2) = 0$  and  $p(C_2) = 1$ , when the classifier is confident.

The parameters  $\alpha_i$  and  $w_0$  characterising the discriminant function are obtained by an optimisation procedure, and  $\alpha_i$  is then non-zero only for *support vectors*  $\mathbf{x}_i$ . The SVM algorithm selects a particular weight vector (as defined by the *support vectors*), which gives rise to a *maximum* margin separator.

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 regularisation and kernel parameters are learnt using ten-fold cross-validation. We discuss our choices of kernel functions in Section IV-F.

In its original form, the SVM classifier output is an uncalibrated real value. A common means of obtaining a probabilistic interpretation is by using Platt's method [Platt, 1999]. This algorithm was later improved by Lin et al. [2007], which

is implemented in the library we use for all SVM training, calibration, and testing, LIBSVM [Chang and Lin, 2011]. Here, using a hold-out set not used for classifier training, a parametric sigmoid model is fit directly to the class posterior  $p(y_* = C_2 | f(\mathbf{x}_*))$ , such that

$$p(y_* = C_2 | f(\mathbf{x}_*)) = \frac{1}{1 + \exp(Af(\mathbf{x}_*) + B)}. \quad (7)$$

The sigmoid parameters  $A$  and  $B$  are determined using Newton's method with backtracking line search. 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 [Rasmussen and Williams, 2006]. Although there is no guarantee that the method converges, in general it works very well and finds the global optimum owing to the convexity of the



objective function.

### 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 [Bishop, 2006]. Each weak learner  $h(\mathbf{x})$  is trained using a weighted form of the dataset in which the 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

$$\text{sign}(f(\mathbf{x}_*)) = \text{sign}\left(\sum_{i=1}^M w_i h_i(\mathbf{x}_*)\right), \quad (8)$$

where  $M$  is the number of weak learners used.

LogitBoost [Friedman et al., 1998] is a popular choice for a boosting-based classifier as it natively outputs class probability estimates following a calibration via a sigmoid. 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 parameters like the learning rate, tree depth, and the number of boosting rounds. The class-conditional probabilities are obtained from the predictor function via

$$p(y_* = C_1 | \mathbf{x}_*) = \frac{\exp(f(\mathbf{x}_*))}{\exp(f(\mathbf{x}_*)) + \exp(-f(\mathbf{x}_*))}, \quad (9)$$

which is the same sigmoid used in the SVM in Section IV-A with parameters  $A = -2$  and  $B = 0$ . The procedure possesses asymptotic optimality as a maximum likelihood predictor [Friedman et al., 1998, Hastie and Tibshirani, 1990]. 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. In our experiments we use 500 learners for training. Throughout this work we use the MATLAB implementation of LogitBoost for classifier training and testing.

Because the LogitBoost classifier ultimately settles on a single decision boundary across the input space, we expect that it will suffer from the same introspective issues as the SVM.

### C. Gaussian Process Classification

Binary classification using a Gaussian Process (GP) [Williams and Barber, 1998, Rasmussen and Williams, 2006] is formulated by first introducing a *latent* function  $f(\mathbf{x})$  and then applying a sigmoid function  $\Phi$  (similar to the sigmoid described in Section IV-A, except that the predictive variance of the GP is used as well as the predictive mean) to obtain the prediction  $p(y_* = C_1 | \mathbf{x}_*) = \Phi(f(\mathbf{x}_*))$ . A GP prior is placed on the latent function  $f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$  characterised 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

$$p(f_* | \mathcal{X}, \mathcal{Y}, \mathbf{x}_*) = \int p(f_* | \mathcal{X}, \mathbf{x}_*, f) p(f | \mathcal{X}, \mathcal{Y}) df, \quad (10)$$

where  $p(f | \mathcal{X}, \mathcal{Y}) = p(\mathcal{Y} | f) p(f | \mathcal{X}) / p(\mathcal{Y} | \mathcal{X})$  is the posterior distribution over latent variables. This is followed by *marginalising* over the latent  $f_*$  to yield the class likelihood  $p(y_* = C_1 | \mathcal{X}, \mathcal{Y}, \mathbf{x}_*)$  as

$$p(y_* = C_1 | \mathcal{X}, \mathcal{Y}, \mathbf{x}_*) = \int \sigma(f_*) p(f_* | \mathcal{X}, \mathcal{Y}, \mathbf{x}_*) df_*. \quad (11)$$

Exact inference is analytically intractable due to the sigmoid likelihood function. Instead, we leverage expectation propagation (EP) [Minka, 2001], a method widely used for this purpose.

The GPC framework offers two key benefits over the other approaches discussed here [Rasmussen and Williams, 2006]. 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 an inherently probabilistic output in the Bayesian sense, 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  $p(y_* = C_1 | \mathcal{X}, \mathcal{Y}, \mathbf{x}_*)$  from Equation (11). This process also gives rise to the well known property of increased variance while far away from the data in GP regression. Again this is in contrast to the SVM or Boosting estimate  $p(y = C_i | \hat{f}, \mathbf{x}_*)$  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 Bayesian classification approaches. Throughout this work we use the GPML toolbox [Rasmussen and Nickisch, 2010] for GPC training and testing.

### D. The Informative Vector Machine

A key drawback of a GPC is its significant computational demand in terms of memory and run time during training and testing, more than any of the other frameworks considered here. This is due to the fact that the GP maintains a mean  $\mu$ , as well as a covariance matrix  $\Sigma$ , which is computed from a kernel function and is of size  $N \times N$ . A number of sparsification methods have been proposed in order to mitigate this computational burden. For efficiency, in this work we adopt one such sparsification method: the Informative Vector Machine (IVM) [Lawrence et al., 2002]. The main idea of this algorithm is to only use a subset of the training points denoted the *active set*,  $\mathcal{I}$ , from which an approximation  $q(f | X, \mathbf{y}) = \mathcal{N}(f | \mu, \Sigma)$  of the posterior distribution  $p(f | X, \mathbf{y})$  is computed. The IVM algorithm computes  $\mu$  and  $\Sigma$  incrementally, and at every iteration  $j$  selects the training point  $(\mathbf{x}_k, y_k)$  which maximises the entropy difference  $\Delta H_{jk}$  between  $q_{j-1}$  and  $q_j$  for inclusion

into the active set. As  $q$  is Gaussian,  $\Delta H_{jk}$  can be computed by

$$\Delta H_{jk} = -\frac{1}{2} \log |\Sigma_{jk}| + \frac{1}{2} \log |\Sigma_{j-1}|. \quad (12)$$

We use an efficient form of this, the details of which can be found in Lawrence et al. [2005]. The algorithm stops when the active set has reached a desired size. We choose this size to be a fixed fraction  $q$  of the training set, which we set to be 0.4. Throughout this work we use the IVM MATLAB toolbox [Lawrence] for both training and testing.

To find the kernel hyper-parameters  $\theta$  of an IVM, two steps are processed in a loop for a given number of times: estimation of  $\mathcal{I}$  from  $\theta$  and minimising the marginal likelihood  $q(\mathbf{y} | X)$ , thereby keeping  $\mathcal{I}$  fixed. Although there are no convergence guarantees, in practice a small number of iterations is sufficient to find good kernel hyper-parameters.

Importantly for our work, since inference with the IVM is similar to that with a GPC, the IVM retains the model averaging described in (11). We argue, therefore, that the IVM provides a significant and well-established improvement in processing speed over a GPC while maintaining its introspective properties (see Section V for details).

### E. Random Forests

Random Forests [Breiman, 2001] are made up of an ensemble of decision trees generated via bagging. Bagging (a portmanteau of “bootstrap aggregating”) involves creating multiple classifiers using different subsets of some aspect of the training data, in this case two aspects are bagged simultaneously: the training data, and the feature dimensions. During testing, the output  $p(C_2)$  is the fraction of the individual trees which classified the datum as being from that class.

The trees contain multiple binary nodes or branches, each of which thresholds on a particular feature dimension of the data, learning the threshold which helps split the training data into the two classes. We have set each tree to use a number of feature dimensions equal to the square root of the total number, as recommended by the literature, with a total of 500 trees. Throughout this work we use the Bagged Decision Tree functions in the MATLAB statistics toolbox (which is an implementation of Random Forests) for both training and testing.

This combination of many differing decision boundaries (one boundary per tree) represents a sampling and then averaging over the version space, similar to the marginalisation over version space which takes place in the Gaussian process classifier. A crucial difference is that in the GPC, each possible model is weighted by its likelihood, and in Random Forests each tree is weighted equally. However, these trees are carefully selected to separate the chosen subset of training data, so this biasing is in a sense a  $\{0,1\}$  weighting. This could be thought of as sampling 500 decision boundaries from the shaded region in Figure 2b and taking an expectation over their responses. This suggests that they should behave in a more introspective manner than the other single-discriminant frameworks like LogitBoost and the SVMs, but perhaps a more sensitively weighted combination of the trees could perform better.

### F. 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

$$k_{LIN}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j + r, \quad (13)$$

where  $r$  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 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

$$k_{SE}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2l^2} \|\mathbf{x}_i - \mathbf{x}_j\|^2\right). \quad (14)$$

In the context of an SVM, the SE function is more commonly known as a *radial basis function* (RBF).

## V. EXPERIMENTAL RESULTS

Our experiments investigate the introspective capacity of the classifiers introduced in Section IV in settings relating to autonomous driving. Specifically, we focus on two tasks: the *classification* of cropped images of road signs, and the *detection* of a salient class against a broad background class. For the detection case, we repeat our experiments across the three data sets detailed in Section V-A, which together contain traffic lights, cars, and pedestrians. In investigating both classification and detection we aim to address the full spectrum of applications commonly encountered in robotics. Classification addresses the case where a decision is made between two, well-defined classes (e.g. two types of traffic signs). We investigate classifier performance when a third, previously unseen class is presented. The detection case is arguably more commonplace, where a single class is separated from a broad (in terms of intra-class variation) *background* class (which is relevant in semantic mapping or detection). Here, the concept of a previously unseen class does not exist explicitly: the inherent assumption is that the data representing the background class capture any non-class object likely to be encountered. In practice this is rarely true, leading to a significant number of novel instances which often result in misclassification. While it could be argued that this issue is ameliorated somewhat by expanding the dataset used for training, we propose that the complexity of the feature space encountered during persistent, long-term autonomy will keep perplexing even the most expansively trained classifiers.

We then apply a decision-making process to the classifiers trained for detection, and show how the quality of each



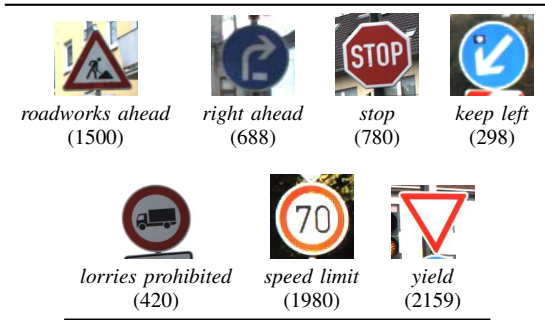


TABLE I: The seven classes of the German Traffic Sign Recognition Benchmark (GTSRB) dataset considered in our work. The numbers in brackets indicate the number of data available per class.

classifier’s decisions change depending on the values chosen for the cost function.

We finish by examining the uncertainty with which each classifier makes errors, and compare the idealised drawings from Figure 4 to the real curves generated from each of the three data sets.

#### A. Datasets

In order to demonstrate the consistency of the introspective capacities of the various frameworks, we evaluate our experiments on several commonly-used data sets which encompass several domains of robotics, namely the detection of various key classes on the road.

1) *Traffic Lights Recognition*: the Traffic Lights Recognition (TLR) dataset [of Mines ParisTech, 2010] is a sequence of colour images taken by a monocular camera from a car driving through central Paris. The TLR dataset comprises just over 11,000 frames, in which most of the traffic lights have been labelled 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 recommended by the authors, we exclude from our experiments any labels of class *ambiguous* or *yellow signal* and any instances which are partially occluded. 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 labelled. 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.

2) *GTSRB*: The German Traffic Sign Recognition Benchmark dataset [Stallkamp et al., 2012] 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. The images are resized according to the parameters in Table II, and then we use the Torralba features from Section V-B for classification.

Parameter	TLR	GTSRB	DP	KITTI
Cropped image height	30	32	96	26
Cropped image width	12	32	48	32
HOG cell size	n/a	n/a	10	10
N. of orientations	n/a	n/a	5	6
Final feature dimension	200	200	950	198

TABLE II: The parameters for the features for the TLR and GTSRB (using Torralba features) and the DP and KITTI data sets (using HOG features).

3) *Daimler Pedestrian*: The examples we use come from the Daimler multi-cue occluded Pedestrian data set (DP) [Enzweiler et al., 2010], and we use the non-occluded monocular intensity images. There are over 52,000 positive and 32,000 negative examples split into training and test sets. The images are resized according to the parameters in Table II, and then we use the HOG features from Section V-B for classification.

4) *KITTI*: The KITTI data set [Geiger et al., 2012] comprises over 7,400 non-sequential colour images from a camera pointing out from the front of a car driving through a German city. The images come with ground truth information for vehicles, with up to 15 in each frame. The images are cropped and resized according to the parameters in Table II, and then we use the HOG features from Section V-B for classification.

#### B. Features

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. [2007] 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  $8 \times 8$  to  $14 \times 14$  pixels) and its location within the object as indicated by a binary mask ( $h \times w$  pixels according to Table II). 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  $d = 200$ .

For the Daimler Pedestrian and KITTI data sets, we have chosen to use Histogram of Oriented Gradients (HOG) [Dalal and Triggs, 2005] features because the classes in question (pedestrians and cars, respectively) have much greater variation than traffic lights, and so a gradient-based feature method performs better than a template matching-based method, which is more appropriate for classes with consistent appearance. We use the implementation in *vlfeat* [Vedaldi and Fulkerson, 2010] and use parameters as detailed in Table II.

#### C. Introspection in Classification

This section investigates classification output when the classifiers are trained on two classes, and then a third, previously unseen class is presented to the classifier. This is an important experiment because classifiers deployed in real-world applications will encounter images which do not closely resemble

Data set	Training data		Test data	
	Positives	Negatives	Positives	Negatives
TLR	250	500	1000	2500
DP	250	500	8000	16000
KITTI	200	500	2000	5000

TABLE III: The number of training and test data of each class used for the detection experiments. The quantities of data from the GTSRB data set for the classification experiments are detailed in Section V-C.

Classifier	Precision	Recall	$F_1$
IVM	1.000	1.000	1.000
Non-linear GPC	1.000	1.000	1.000
Linear GPC	1.000	1.000	1.000
Non-linear SVM	1.000	1.000	1.000
Linear SVM	1.000	1.000	1.000
LogitBoost	1.000	1.000	1.000
Random Forest	1.000	1.000	1.000

TABLE IV: The classification performance when separating *stop* sign from the *lorries prohibited* signs from the GTSRB data set. Note that different class combinations were found to yield classifiers of similar quality.

the data used to train them, and an introspective classifier will respond to these with high uncertainty. As examples of classes typically encountered in autonomous driving applications we use a subset of the GTSRB dataset (see Section V-A2).

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). 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 IV, and show that classification performance by the commonly-used metrics (precision, recall, and  $F_1$  measure) is commensurate across all classifiers. The corresponding precision-recall curve confirms the perfect separation of the classes and has been omitted here as it is otherwise uninformative. The classifiers are then tested on 200 instances of previously unseen classes *roadworks ahead*, *keep left*, *70kph*, *yield*, and *right ahead*. The normalised entropy histograms for both the seen and the unseen test classes are shown in Figure 5. All classifiers are confident when tested on classes which were present in the training set, which is what we would expect. For the unseen test classes, the mean normalised entropies for the GPC-based classifiers (IVM, non-linear GPC, and linear GPC) and the Random Forests are more consistently high than those of the other classification frameworks, indicating that they reliably exhibit greater uncertainty in their judgement. Conversely, the LogitBoost classifier is extremely confident in all of its classifications with a very narrow distribution, and the non-linear and linear SVMs have inconsistent levels of uncertainty. These are effects consistently observed throughout our experiments, which we attribute to the manner in which the probabilities are estimated (as detailed in Section IV). The unseen sign for which the classifiers respond with the lowest uncertainty (greatest confidence) is the 70 kph sign. We propose that this is due to its similarity

with one of the training classes, namely the ‘lorries prohibited’ sign.

In order to mitigate any influences of 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 forty iterations of classifier training and testing with a random dictionary and training and test datasets resampled for each run. The results, presented in Table V, are consistent with those in Figure 5 in that the GPCs and Random Forest tend to be more consistently uncertain for the unseen test classes, while SVM and LogitBoost are more confident with an often significantly narrower distribution of normalised entropy values. The results in Table V 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 some unseen class and the classes in the training set.

We draw the conclusion that when faced with test data which are not represented by the training data, the GPC-based classifiers and Random Forest are more consistently uncertain than the other classifiers, which is the introspective behaviour we seek.

#### D. Introspection in Detection

We investigate the same classification frameworks as before on various detection tasks, which each have a salient positive class and a broad background class. We evaluate the classifiers on three data sets: TLR (traffic lights), Daimler Pedestrian, and KITTI (cars), as detailed in Section V-A.

As with the classification task, we first verify the efficacy of the features selected and the training procedures employed. Table VI shows the classification performance for classifiers trained using the number of data shown in Table III. We have chosen these values for two reasons. Firstly, we are trying to highlight low-probability catastrophic events, which will be few in number for the size of the test sets we are considering here, but over the life-long autonomy we envisage for our robots will occur in non-negligible numbers; larger training sets reduce the prevalence of these low-probability events, but will never be able to eliminate them. Secondly, we are using off-the-shelf implementations of commonly-used classification frameworks to keep the comparisons fair. We note that in autonomous driving scenarios we typically see more negative examples than positive examples, and so have kept the training and test sets roughly to the same 1:2 ratio of positives to negatives. While scanning an entire urban scene for pedestrians is likely to yield many thousands more negatives than positives, it is common [Enzweiler et al., 2012, Fairfield and Urmson, 2011] to use 3D information or prior maps to greatly reduce the portion of each image that needs to be scanned, and thus making the ratio of positive to negative windows much more even.

Figure 6 shows the corresponding precision-recall curves for the classifiers across the data sets. The detection task, having a varied background class and greater variation within the positive class, is more challenging than the classification

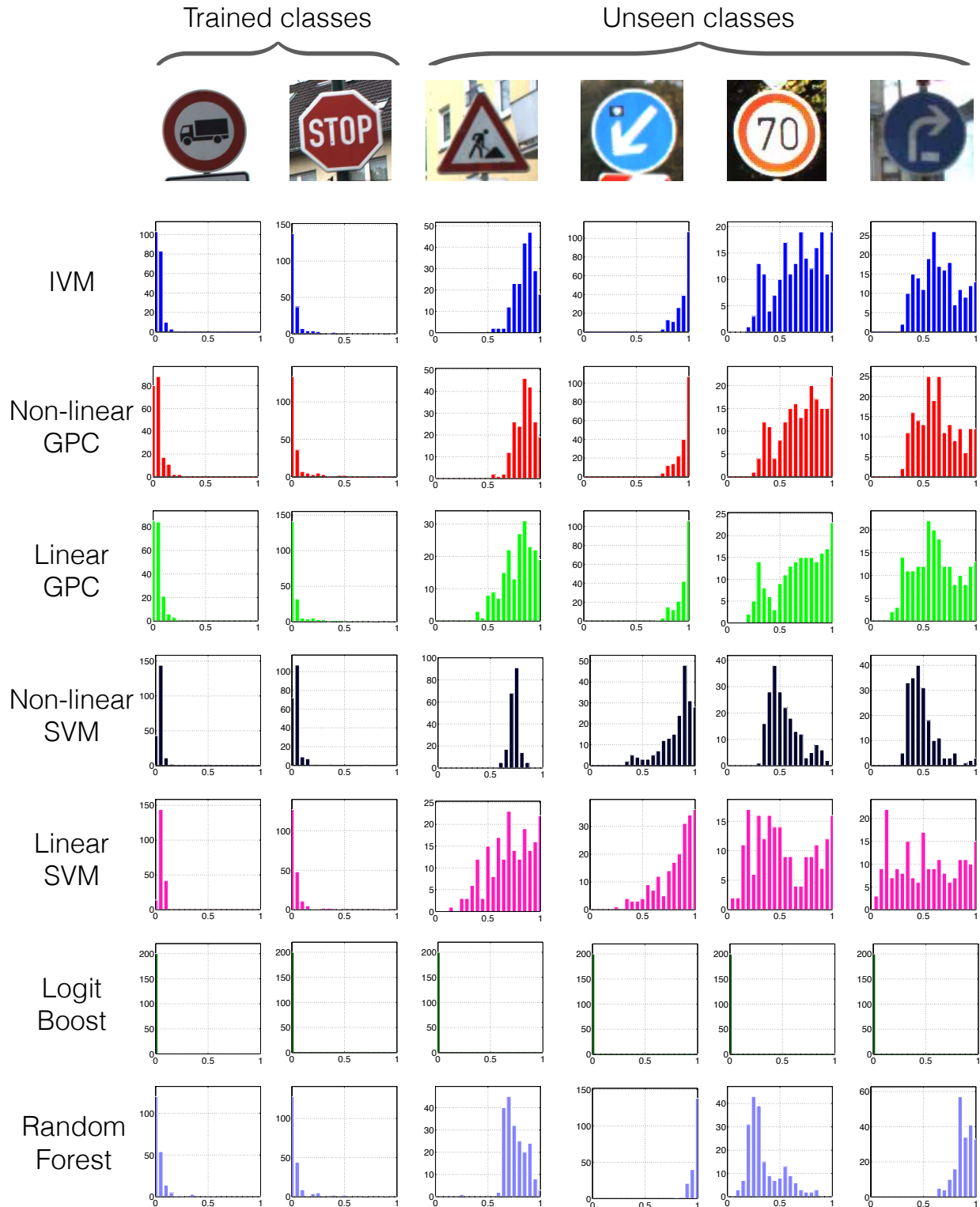


Fig. 5: Normalised entropy histograms (frequency vs NE) of the marginal probabilities for a variety of classifiers trained on the road sign classes *stop* and *lorries prohibited* and tested on not only the training classes, but also classes which do not appear in the training set (*roadworks ahead*, *keep left*, *70kph*, and *right ahead*). Higher values for normalised entropy imply more uncertainty in classifier output, so we expect the more introspective classifiers to be certain (low NE, left-hand end of the x-axis) on the trained classes and uncertain (high NE, right-hand end of the x-axis) for the unseen classes.






Test Class	Classifier	Normalised Entropy	
		$\mu \pm \text{std. err.}$	$\sigma \pm \text{std. err.}$
	IVM	<b>0.776 ± 0.081</b>	0.145 ± 0.030
	Non-linear GPC	0.751 ± 0.087	0.152 ± 0.029
	Linear GPC	<b>0.776 ± 0.108</b>	0.150 ± 0.041
	Non-linear SVM	0.476 ± 0.101	0.089 ± 0.056
	Linear SVM	0.664 ± 0.122	0.250 ± 0.041
	LogitBoost	0.019 ± 0.025	0.041 ± 0.073
	Random Forest	0.756 ± 0.137	0.149 ± 0.053
	IVM	<b>0.794 ± 0.117</b>	0.106 ± 0.026
	Non-linear GPC	0.779 ± 0.124	0.107 ± 0.024
	Linear GPC	0.777 ± 0.202	0.124 ± 0.058
	Non-linear SVM	0.537 ± 0.126	0.028 ± 0.036
	Linear SVM	0.494 ± 0.239	0.222 ± 0.049
	LogitBoost	0.016 ± 0.022	0.031 ± 0.059
	Random Forest	0.736 ± 0.166	0.078 ± 0.027
	IVM	0.539 ± 0.140	0.173 ± 0.023
	Non-linear GPC	0.546 ± 0.144	0.168 ± 0.023
	Linear GPC	<b>0.569 ± 0.166</b>	0.177 ± 0.026
	Non-linear SVM	0.407 ± 0.077	0.076 ± 0.053
	Linear SVM	0.315 ± 0.195	0.197 ± 0.058
	LogitBoost	0.008 ± 0.004	0.012 ± 0.026
	Random Forest	0.394 ± 0.121	0.138 ± 0.029
	IVM	0.579 ± 0.133	0.137 ± 0.020
	Non-linear GPC	0.577 ± 0.130	0.136 ± 0.019
	Linear GPC	0.585 ± 0.188	0.151 ± 0.029
	Non-linear SVM	0.488 ± 0.111	0.039 ± 0.034
	Linear SVM	0.177 ± 0.127	0.155 ± 0.056
	LogitBoost	0.014 ± 0.019	0.030 ± 0.056
	Random Forest	<b>0.668 ± 0.161</b>	0.113 ± 0.027
	IVM	<b>0.931 ± 0.025</b>	0.080 ± 0.026
	Non-linear GPC	<b>0.934 ± 0.021</b>	0.079 ± 0.023
	Linear GPC	0.925 ± 0.031	0.085 ± 0.027
	Non-linear SVM	0.641 ± 0.168	0.100 ± 0.047
	Linear SVM	0.705 ± 0.142	0.212 ± 0.049
	LogitBoost	0.059 ± 0.103	0.077 ± 0.127
	Random Forest	0.904 ± 0.089	0.088 ± 0.043

TABLE V: Mean and standard deviation of normalised entropies (including standard errors) from 40 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.

task. Classification performance according to the conventional metrics is commensurate across all frameworks. The Random Forest performs best for the TLR data set, and the non-linear SVM and IVM perform consistently highly in the Daimler Pedestrian and KITTI data sets. The GPC-based classifiers all have commensurate performance in terms of precision and recall.

In Figures 7, 8, and 9 we demonstrate how the lack of introspection can impact classification performance when accept/reject decisions are determined by classification confidence, with one figure per data set. 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 entropy) we get values which correspond to those in Table VI. It is desirable for a classifier to be close to the top left hand corner of the graphs pertaining to *true* classifications (top row) and close to the bottom right of the graphs pertaining to *false* classifications (bottom row). This would correspond to making true classifications with low

uncertainty (high confidence) and making incorrect decisions with high uncertainty.

Although the SVMs and LogitBoost classifiers generally make true positive and true negative classifications with higher certainty (i.e. low normalised entropy) than for the GPC variants, they are also more confident when making mistakes. This balance is discussed in more detail in Section VI, but in summary we consider the avoidance of high-confidence errors to be of primary importance, and after that, an increase in classifications which are both confident and true results in a more useful classifier.

The GPC-based classifiers (IVM, non-linear and linear GPCs) behave very similarly to each other particularly in the TLR and Daimler Pedestrian data sets, and perform very well in terms of making mistakes with very high uncertainty. The price paid for this more realistic assessment of the classification confidence 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.

The Random Forest is consistently uncertain in terms of all four decision outcomes across all data sets. This is because the probabilities it outputs are rarely far from  $p(C_2) = 0.5$ . The fact that it performs rather well in terms of accuracy, precision and recall indicates that it is under confident.

The difficulties of the data sets clearly vary from the PR curves and the confidences of the true detections, with TLR being the easiest, followed by Daimler Pedestrian data set, and then KITTI being the most challenging. This is likely to be a result of the variation within the positive class paired with the low number of positive exemplars in the training set (see Table III).

### E. Decision Making

In Section III-C we discussed the importance of the loss function  $L(a, C_i)$  and how it shapes the decision of which action  $a$  to choose, given some estimates of the state of the environment  $\{p(C_1), \dots, p(C_{|C|})\}$ . In robotics, we seek classification frameworks which allow our robots to make decisions which are faithful to the values instilled by the losses incurred for particular outcomes. For instance, if we make the cost associated with a particular outcome very large, then the actions which can lead to that outcome should be chosen more infrequently, or at least only when the classifier gives a very confident estimate of the state of the environment. The behaviour we seek is for classifiers to behave appropriately given any relative costs associated with the possible outcomes. We characterise this ‘appropriateness’ by comparing the total cost incurred when using each classifier as part of the decision-making pipeline, and we do so while varying the ratio of the costs of false positive and false negative outcomes. Note that we cannot mitigate the dangerous tendencies of less introspective classifiers by adjusting the costs; each decision is made by weighting the probabilities produced by a particular framework, and thus if the probabilities are a poor indicator

Classifier	TLR			Daimler Pedestrian			KITTI		
	Precision	Recall	$F_1$	Precision	Recall	$F_1$	Precision	Recall	$F_1$
IVM	0.995	0.916	0.954	0.953	0.872	0.911	0.868	0.725	0.790
Non-linear GPC	0.992	0.912	0.950	0.956	0.874	0.913	0.853	0.735	0.790
Linear GPC	0.988	0.899	0.941	0.956	0.875	0.914	0.816	0.708	0.758
Non-linear SVM	0.996	0.920	0.956	0.959	0.869	0.912	0.836	0.749	0.790
Linear SVM	0.967	0.910	0.938	0.932	0.876	0.903	0.813	0.709	0.757
LogitBoost	0.978	0.908	0.942	0.961	0.794	0.869	0.826	0.681	0.747
Random Forest	1.000	0.897	0.946	0.984	0.598	0.744	0.894	0.551	0.682

TABLE VI: The classifiers’ performances for the detection tasks across data sets according to conventional metrics. Precision, recall, and F-measure are calculated by thresholding the classifiers’ probabilities at 0.5. The SVMs and GPCs give very similar results across the data sets, with the Logitboost and Random Forest performing slightly worse than the others with the more difficult data sets.

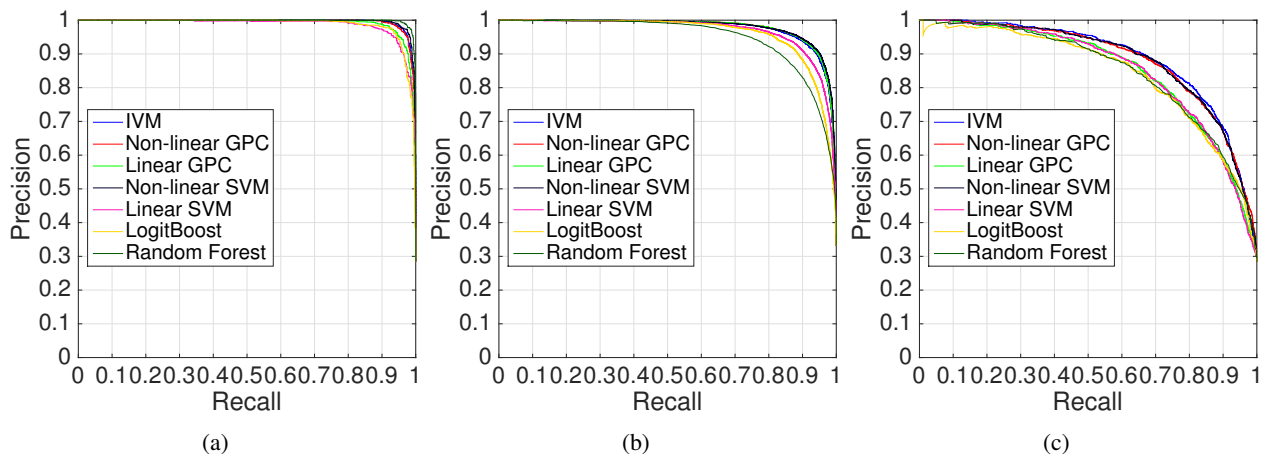


Fig. 6: Precision-recall curves for the (a) TLR, (b) Daimler Pedestrian and (c) KITTI data sets. Note the increasing difficulty of the data sets, and the consistency and commensurate nature of the classifiers in terms of these metrics. (Best viewed in colour.)

of the truth of the classification, wrong decisions will be made regardless of the costs set.

For each of the three data sets, we use the probabilistic output of the classifiers to drive the decision-making pipeline, and evaluate the decisions made. We set the costs of true positive and true negative outcomes as 0, and the cost of a false positive outcome as 1. The value for the last outcome, the false negative or missed pedestrian, is varied from 1 to  $10^7$ . This cost of the false negative error appears on the x-axes of Figures 10, 11, and 12. The y-axis of the *left-hand* figure in each pair denotes the number of true outcomes (both positive and negative together), and the y-axis of the *right-hand* figure denotes the total cost of all the decisions made.

These pairs of graphs demonstrate the trade-off between classifiers which avoid catastrophic decisions, and those which might be so cautious that they never take the higher-risk action. The *left-hand* graphs demonstrate the rate at which the classifiers’ decisions become more and more cautious as the cost of a false negative increases. We do not consider one to be superior to another in terms of introspection, albeit it may tell you about the usefulness of that classifier. On the right hand graphs, the ideal is for a curve to be as low as possible (close to the x-axis). This would represent a classifier which makes good decisions given any particular cost ratio.

In Section III-C we described another way to characterise the introspective tendencies of a classifier: by examining the distribution of errors across  $p(C_2)$ . In Figure 4 we showed that the number of cumulative errors as you increase the half-length of the orange box (from Figure 4d) from 0 to 0.5 can take various shapes, and we showed the shape induced by a more introspective classifier in teal (in Figure 4e). In Figure 13 we show those curves for our classifiers across the three data sets. Comparing it to Figure 4e, we see that for all three data sets, the curves for the classifiers which consider multiple discriminants (Random Forests and the GPC-based classifiers) are closer to the desirable teal curve than the single-discriminant classifiers (the SVMs and LogitBoost). The Random Forest very strongly resembles the teal curve across all data sets, and in the case of the KITTI data set the IVM does also. This is a further, strong indication of the introspective power of the Random Forests and GPC-based classifiers over the others.

## VI. DISCUSSION

To what degree is introspection a property of a single classifier, or of a classification framework? Can one SVM be more introspective than another SVM? Are GPCs invariably more



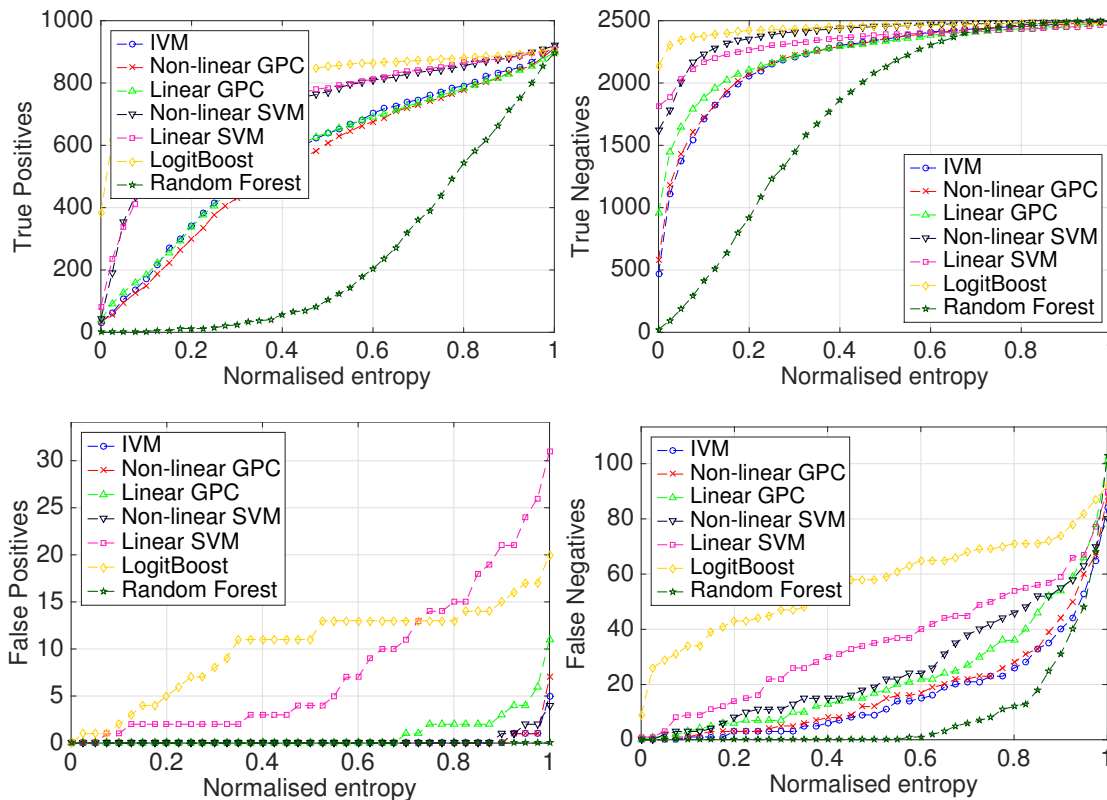


Fig. 7: Cumulative frequency plots of classification confusion (true positives, true negatives, false positives, and false negatives) against normalised entropy. The classifiers have been trained on 250 traffic lights against 500 background patches, and tested on 1,000 instances of traffic lights and 2,500 background patches. Note that lower normalised entropy implies more certainty in classification. A more introspective classifier is one that simultaneously exhibits higher uncertainty (as witnessed by larger normalised entropy in its output) when processing difficult instances and is more confident when it is correct. Consequently, class decisions 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.)

introspective than SVMs? The results in this paper indicate a consistent behaviour of particular classification frameworks across particular tasks (such as classification or detection), which we attribute to the manner in which the classifiers are designed (see Section IV). Thus we expect that introspection quality is inherent to classification *frameworks* rather than individual classifiers.

That said, varying the choice of kernel (and its parameters) do produce very different behaviours, and it may be possible to instil an improved introspective sense with an appropriate choice of kernel. The reason for this is that in every framework there is a link between classification confidence and distance in kernel space. In the GPC, a test datum which lies far away from the training data (in kernel space) yields a more uncertain classification. But how can we guarantee that new, unseen classes will be far away from our training data? In truth, we cannot. We can only hope that the kernel has found a warping of the feature space which adequately separates the two classes of training data, and that new, unseen classes will be sufficiently disparate from the training data in kernel space to yield an uncertain classification. Owing to the opaque nature of the kernel function, we cannot assume that points which are close together in feature space will also be close in kernel space. This brings into question the sanity of using distance

in kernel space as a metric for uncertainty in classification.

It should be noted that the sparse nature of the IVM could be expected to reduce its introspective capabilities at the expense of computational efficiency, when in fact it seems to outperform the non-linear GPC in many cases. We attribute this to the fact that the two implementations are from different libraries and it is likely that the optimisation procedures in the IVM are superior to those in the GPC, resulting in a better choice of hyper parameters and thus a more effective sense of how distance should relate to uncertainty.

In this paper we have investigated a variety of applications, feature types, class types and quantities, as well as the nuances between classification and detection. This is because it is not always possible to determine the introspective quality of a classification framework based on a single classifier and test set. Can we say whether a classifier which is uncertain about *all* decisions like the Random Forest is introspective or not? We suggest that it is perhaps introspective, but certainly not as useful as one which can also make correct classifications confidently. A very introspective classifier will have a strong correlation between confidence and correctness. Some situations, such as when a classifier is uncertain about everything, do not yield enough information to determine a classifier's introspective capacity. Another such situation is where a

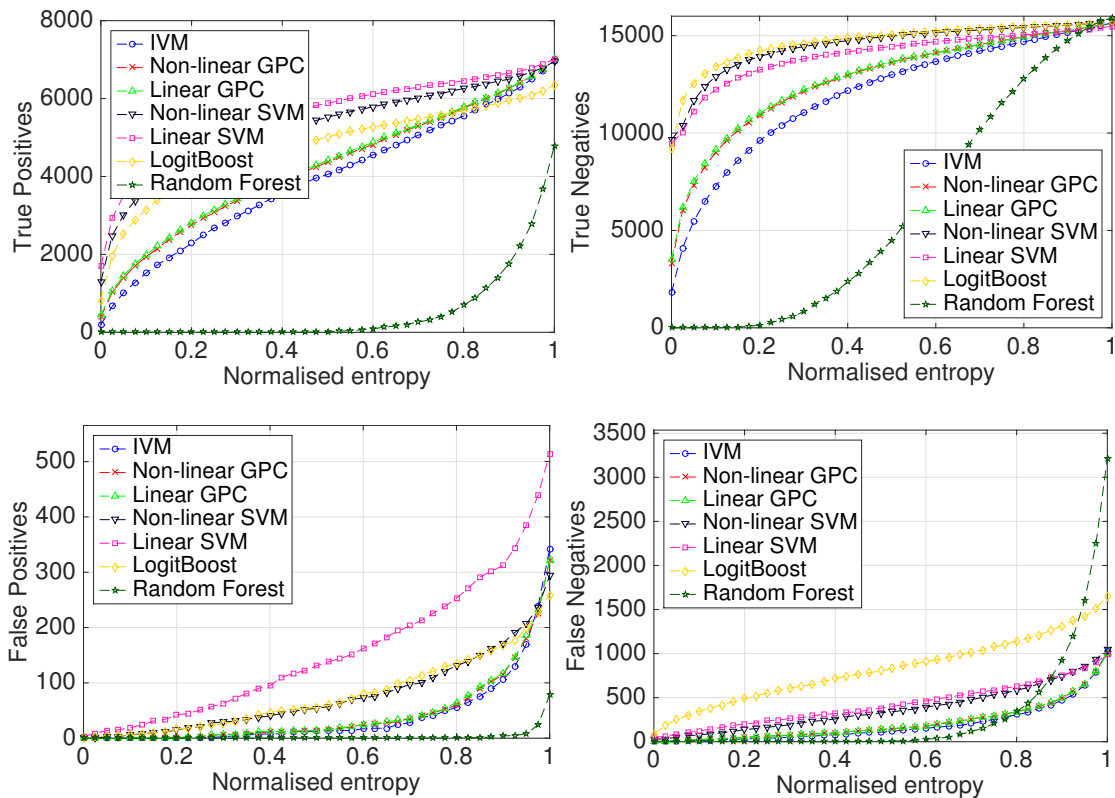


Fig. 8: Cumulative frequency plots of classification confusion (true positives, true negatives, false positives, and false negatives) against normalised entropy (uncertainty), using the Daimler Pedestrian data set. The classifiers are trained on 250 and 500 instances of pedestrians and background respectively, and are tested on 8,000 and 16,000 of those classes. See the caption for Figure 7 for more detail. (Best viewed in colour.)

classifier gives perfect classification results. Without errors, we cannot judge how a classifier deals with the unexpected (unless we evaluate it on truly alien data, such as in the unseen class experiment of Section V-C).

We have motivated the desire for a classifier to make mistakes with high uncertainty, and we have shown that if this is not the case, it will make unpredictable and expensive errors. In addition, for the classifier to be useful, we would also like its true classifications to be made with high confidence. All the classifiers we have investigated in this work have fallen short of both targets, but to varying degrees. If we cannot have both, is there a trade-off to be struck, and are some combinations more desirable than others? One difficulty lies in formally defining how a ‘perfectly introspective’ classifier should behave. We have approached this in terms of increased distance between training and test data leading to uncertainty, and that the degree of uncertainty should indicate the likelihood of a mistake being made. More rigidly, we propose that classifications with an uncertainty of 0.9 should be incorrect by one mistake in ten, on average. If this were true, we could make very well informed decisions.

Although we have found that the GPC-based classifiers exhibit a greater introspective quality than the other classifiers tested in this work, it must be said that they are still far from perfectly introspective, regardless of our choice of the definition of ‘perfect’. One obvious point of improvement for the Random Forest is that it is very under-confident when

it makes true classifications, which is also a feature of the GPC-based classifiers, although to a much lesser extent. It is interesting to note that IVMs with a successively higher active-set-fraction  $q$  get more and more confident, both in terms of true and false classifications, so it may be possible to improve them by simply training them on more data. If an introspective classifier’s predictions all have high uncertainty, this could be a useful sign that the problem is too complex and more training data are required.

## VII. 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 an appropriate assessment of predictive variance. Our experimental results imply that, despite commensurate performance as measured by more conventional metrics, GPC-based classifiers possess a more pronounced introspective capacity than other classification frameworks commonly employed in robotics, maintaining a useful balance between being confident when they are correct, and uncertain when they are making mistakes. We attribute this to their consideration of distance between data, and accounting for predictive variance over the space



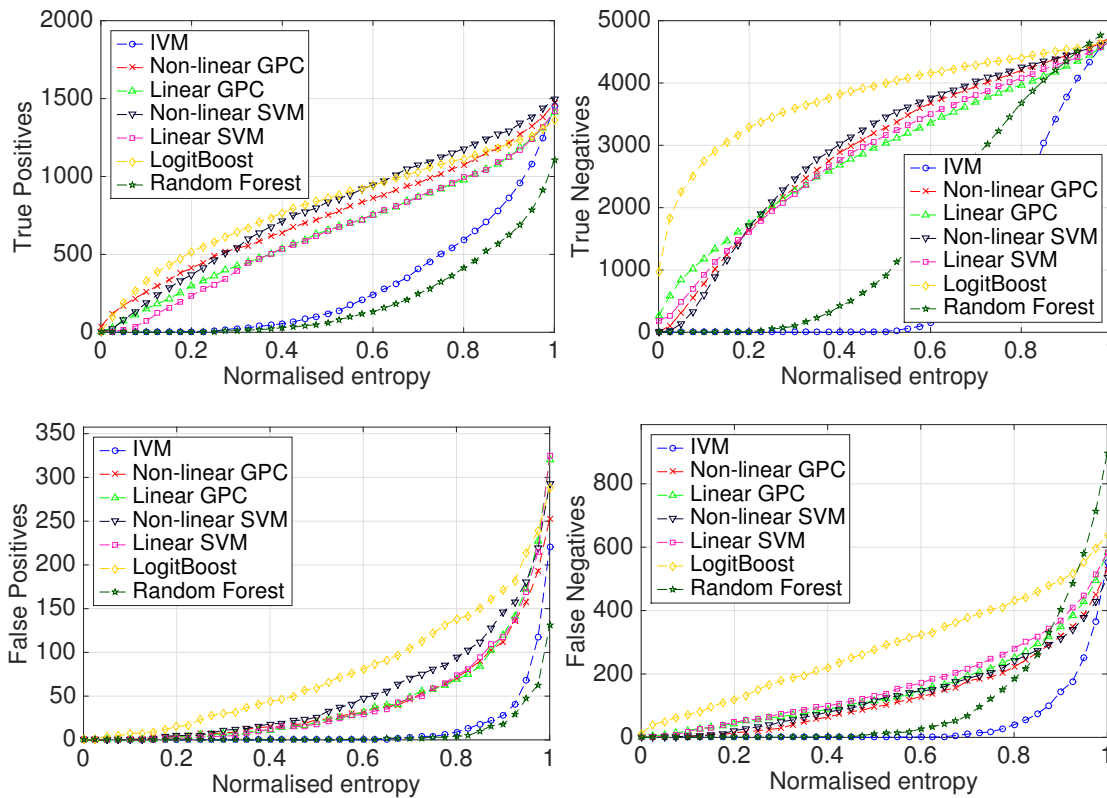


Fig. 9: Cumulative frequency plots of classification confusion (true positives, true negatives, false positives, and false negatives) against normalised entropy (uncertainty), using the KITTI data set. The classifiers are trained on 200 and 500 instances of pedestrians and background respectively, and are tested on 2,000 and 5,000 of those classes. See the caption for Figure 7 for more detail. (Best viewed in colour.)

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. As a result of this, model-averaging classifiers make better decisions than single-discriminant classifiers like SVMs, and thus will cause fewer catastrophic accidents despite appearing worse in terms of F-measure.

### VIII. ACKNOWLEDGEMENTS

This work is funded under the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement Number 269916 (V-CHARGE) and by the UK EPSRC Grant Number EP/J012017/1.

### REFERENCES

- D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 169–176, 2005.
- L-P Berczi, I. Posner, and T. D. Barfoot. Learning to Assess Terrain from Human Demonstration Using an Introspective Gaussian Process Classifier. In *IEEE International Conference on Robotics and Automation (ICRA)*, To appear 2015.
- C. M. Bishop. *Pattern Recognition and Machine Learning*, volume 4. Springer New York, 2006.
- L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, October 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL <http://dx.doi.org/10.1023/A:1010933404324>.
- C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- C-C Chang and C-J Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.
- 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.
- M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila. Multi-Cue Pedestrian Classification with Partial Occlusion Handling. 2010.
- M. Enzweiler, M. Hummel, D. Pfeiffer, and U. Franke. Efficient Stixel-Based Object Recognition. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1066–1071. IEEE, 2012.
- N. Fairfield and C. Urmson. Traffic Light Mapping and Detection. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5421–5426. IEEE, 2011.
- J. Friedman, T. Hastie, and R. Tibshirani. Additive Logistic

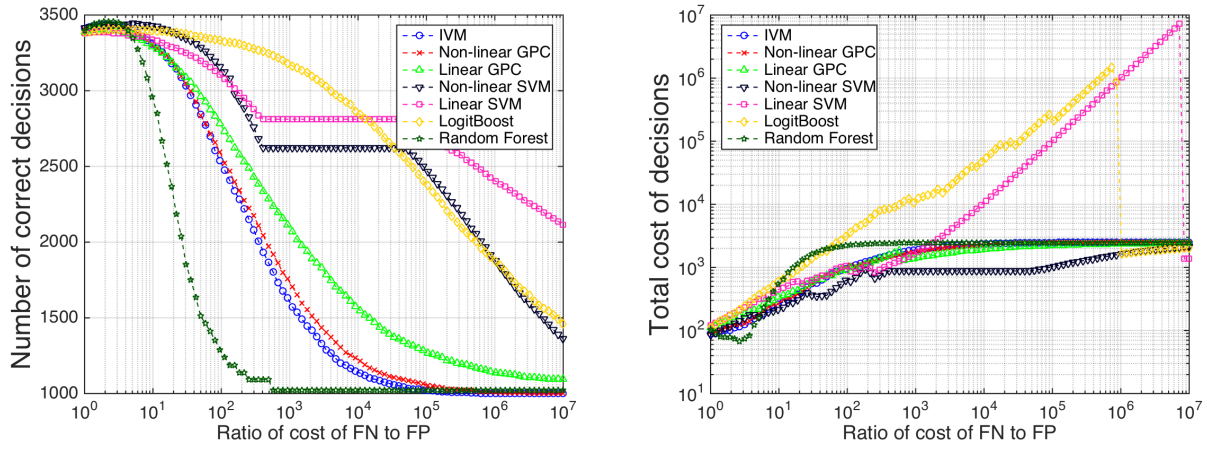


Fig. 10:

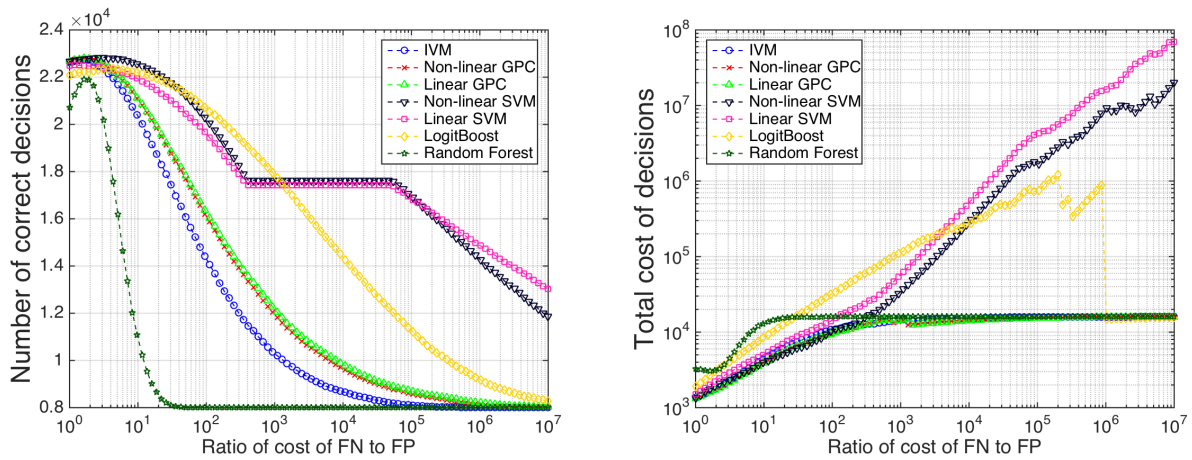


Fig. 11:

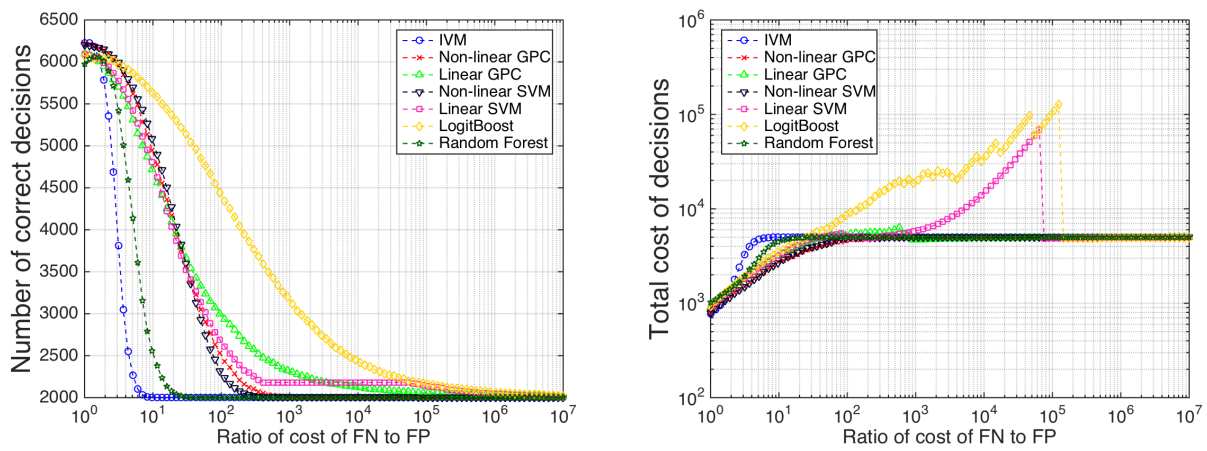


Fig. 12:

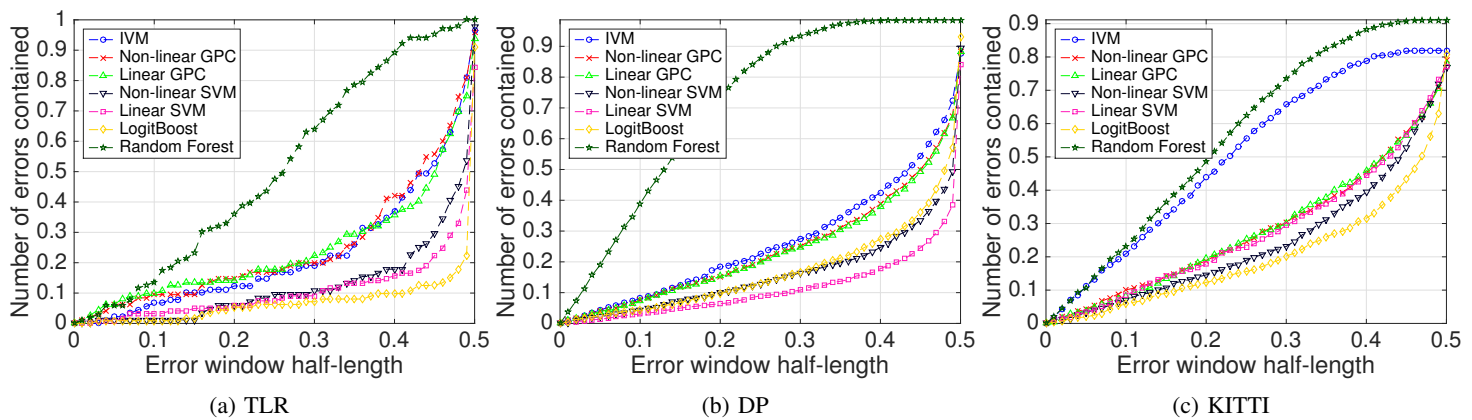


Fig. 13: Following Figure 4, we show the errors contained about  $p(C_2) = 0.5$  as we increase the size of the orange box. To generate these curves, we randomly sampled 1,000 positive and 1,000 negative test data and count the number of errors. Note that a classifier which is uncertain when it makes mistakes will be closer to the top left of each plot. This together with the confidence of correct decisions show the two sides of the introspective coin.

- Regression: a Statistical View of Boosting. *Annals of Statistics*, 28:2000, 1998.
- A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- H. Grimmert, R. Paul, R. Triebel, and I. Posner. Knowing When We Don't Know: Introspective Classification for Mission-Critical Decision Making. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- T. J. Hastie and R. J. Tibshirani. *Generalized additive models*. Chapman & Hall/CRC, 1990.
- A. Holub, P. Perona, and M. C. Burl. Entropy-based active learning for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–8. IEEE, 2008.
- T. Hospedales, S. Gong, and T. Xiang. Finding Rare Classes: Active Learning with Generative and Discriminative Models. *Transactions on Knowledge and Data Engineering, IEEE Transactions on*, 25(2):374–386, 2013.
- A. Huang and S. Teller. Probabilistic Lane Estimation using Basis Curves. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.
- A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2372–2379, 2009.
- A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Gaussian processes for object categorization. *International Journal of Computer Vision*, 88(2):169–188, 2010.
- S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at <http://planning.cs.uiuc.edu/>.
- N. D. Lawrence. MLTOOLS. <http://staffwww.dcs.shef.ac.uk/people/N.Lawrence/mltools/>.
- N. D. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. *Advances in Neural Information Processing Systems*, 15: 609–616, 2002.
- N. D. Lawrence, J. C. Platt, and M. I. Jordan. Extensions of the Informative Vector Machine. In *Proceedings of the First International Conference on Deterministic and Statistical Methods in Machine Learning*, pages 56–87. Springer-Verlag, 2005. URL [http://dx.doi.org/10.1007/11559887\\_4](http://dx.doi.org/10.1007/11559887_4).
- L. Li, M. L. Littman, and T. J. Walsh. Knows what it knows: A framework for self-aware learning. In *Proceedings of the 25th International Conference on Machine Learning, ICML '08*, pages 568–575, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390228. URL <http://doi.acm.org/10.1145/1390156.1390228>.
- Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. A Note on Platt's Probabilistic Outputs for Support Vector Machines. *Journal of Machine Learning*, 68(3):267–276, Oct 2007. ISSN 0885-6125. doi: 10.1007/s10994-007-5018-6. URL <http://dx.doi.org/10.1007/s10994-007-5018-6>.
- O. Martínez-Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard. Supervised semantic labeling of places using information extracted from sensor data. *Robotics and Autonomous Systems*, 55(5):391–402, 2007. ISSN 0921-8890. doi: <http://dx.doi.org/10.1016/j.robot.2006.12.003>.
- P. Matikainen, R. Sukthankar, and M. Hebert. Model Recommendation for Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2256–2263, 2012.
- D. Meger, P. E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, and D. G. Lowe. Curious george: An attentive semantic robot. *Robotics and Autonomous Systems*, 56(6):503–511, 2008.
- T. P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.
- A. Niculescu-Mizil and R. Caruana. Predicting Good Probabilities with Supervised Learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632. ACM, 2005.

- Robotics Centre of Mines ParisTech. Traffic Lights Recognition (TLR) data set, 2010. [www.lara.prd.fr/benchmarks/traffilightrecognition](http://www.lara.prd.fr/benchmarks/traffilightrecognition).
- 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.
- J. C. Platt. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances In Large Margin Classifiers*, pages 61–74. MIT Press, 1999.
- I. Posner, M. Cummins, and P. Newman. A generative framework for fast urban labeling using spatial and temporal context. *Autonomous Robots*, 2009. doi: 10.1007/s10514-009-9110-6. URL <http://dx.doi.org/10.1007/s10514-009-9110-6>.
- A. Pronobis and P. Jensfelt. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3515–3522, 2012.
- C. E. Rasmussen and H. Nickisch. Gaussian Processes for Machine Learning (GPML) Toolbox. *J. Mach. Learn. Res.*, 11:3011–3015, December 2010. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=1756006.1953029>.
- C. E. Rasmussen and C. K. I. Williams. Gaussian Processes for Machine Learning. *The MIT Press, Cambridge, MA, USA*, 2006.
- 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.
- J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 2012. ISSN 0893-6080. doi: 10.1016/j.neunet.2012.02.016. URL <http://www.sciencedirect.com/science/article/pii/S0893608012000457>.
- S. Tellex, P. Thaker, R. Deits, T. Kollar, and N. Roy. Toward Information Theoretic Human-Robot Dialog. In *Robotics: Science and Systems*, 2012.
- 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*, 23(9):661–692, 2006. ISSN 1556-4967. doi: 10.1002/rob.20147.
- S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.
- A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing Visual Features for Multiclass and Multiview Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):854–869, 2007. ISSN 0162-8828.
- A. Vedaldi and B. Fulkerson. VLFeat - an open and portable library of computer vision algorithms. In *ACM International Conference on Multimedia*, 2010.
- J. Velez, G. Hemann, A. S. Huang, I. Posner, and N. Roy. Planning to perceive: Exploiting mobility for robust object detection. In *Proc. ICAPS*, 2011.
- C. K. I. Williams and D. Barber. Bayesian classification with Gaussian processes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1342–1351, 1998.
- W. Zhang, X. Stella, and Y. S. Teng. Power SVM: Generalization with Exemplar Classification Uncertainty. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012.



# Corrigendum: Introspective Classification for Robot Perception

Hugo Grimmert Rudolph Triebel Rohan Paul Ingmar Posner

## I. INTRODUCTION

The authors wish to revise of the conclusions drawn in Section V of the original paper [Grimmett et al., 2016] in light of a recently emerged peculiarity of the probabilistic calibration. Due to particular choices of optimisation bounds, the SVMs never return probabilities (or henceforth in the corrigendum, measurements  $z$ ) in the range  $[\epsilon, 0.0025]$ , where  $\epsilon < 10^{-6}$ . Some of the measurements which should be made within this range are moved to a point less than  $\epsilon$ , and thus an incorrect decision may appear more confident, and incur a greater cost.

This finding explains the flat sections in the left-hand graphs of Figures 10, 11, and 12 in the original paper. The rapidly increasing nature of the SVMs on the right-hand graphs indicated that they were accruing a cost due to high-confidence false negative errors. This behaviour can now be partly attributed to this probabilistic calibration, although we will see that the SVMs continue to produce some catastrophic mistakes. In this corrigendum we have changed the bounds of the optimisation, thereby removing this ‘blind spot’, and allowing the SVMs to make better decisions.

The authors wish to point out that while this affects Figures 1, 10, 11, and 12 (replaced by Figure 5 below), the changes only affect the most confident of decisions, and so the concept of introspection, the reasoning behind it, and any conclusions up to that point are unaffected.

In the remainder of this corrigendum we present updated results and conclusions, and contextualise these against two idealised classifiers which clearly demonstrate the effects of introspection in decision making. The idealised classifiers, detailed in the next section, extend the idea of the ideal introspective classifier introduced in Figure 4 of the original paper.

We confirm that introspection in decision making is crucial, and that there are differences in the introspective capacities of the real classifiers benchmarked as part of this study. However, we newly conclude that these differences are not sufficient to consistently affect their decision-making abilities in the high-confidence decision-making experiment presented.

## II. IDEALISED CLASSIFIERS

In Figures 4a-c of the original paper we demonstrate the merit of making mistakes with high uncertainty. Instead of considering the error functions in Figure 4a, let us consider two new idealised classifiers, each defined by a pair of probability density functions  $f_1(z)$  and  $f_2(z)$ . These two density functions define the response of a classifiers for each of two

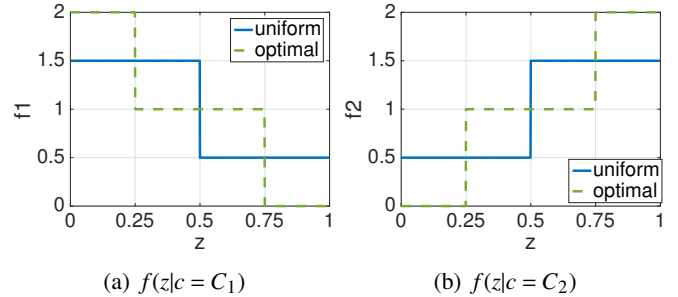


Fig. 1: The probability density functions which define two idealised classifiers, called *uniform* and *optimal*. (a) shows the distribution of the measurement  $z$  which classifiers give for the background class, and (b) shows the same for the positive class (e.g. foreground). These two pairs of probability density functions are chosen because they generate the error functions in Figure 2 for balanced data sets. We draw samples from these distributions to simulate the classifiers’ behaviours.

classes (with  $f_1(z)$  for class  $C_1$ , etc), and are shown in Figure 1 of this corrigendum.

We define making an error as either returning a measurement  $z < 0.5$  for an instance of the positive class, or returning a measurement  $z > 0.5$  for an instance of the negative class. If we plot the probability of error given the measurement  $z$  for these two classifiers applied to a balanced data set, shown in Figure 2, we see that the *uniform* classifier does not correlate confidence with correctness, because it has the same probability of error whatever the value of  $z$ , and thus is not introspective. In contrast, the *optimal* classifier makes mistakes only with high uncertainty (in the region around  $z = 0.5$ ). This correlation between correctness and confidence makes the *optimal* classifier more introspective than the *uniform* classifier.

We note that the expected error rate is  $p(e) = 0.25$  for both classifiers, and that the difference in their behaviour arises only from where in the range of  $z$  they make mistakes.

Considering these two classifiers to be at opposing ends of a spectrum, where one is indifferent to uncertainty (the *uniform* classifier) and one is always correct below a given uncertainty threshold (the *optimal* classifier), we can compare them with the real classifiers.

For the results in this corrigendum, we simulate the idealised classifiers’ responses to 5,000 negative examples and 2,000 positive examples by drawing them from  $f_1(z)$  and  $f_2(z)$ , respectively. This process is repeated ten times. These numbers are chosen to be consistent with the experiments conducted on

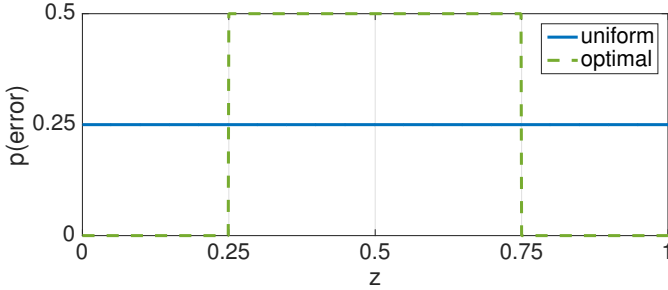


Fig. 2: The likelihood of making an error given some measurement  $z$  (or *error function*) for two idealised classifiers, given by Figure 1. The *uniform* classifier is non-introspective because the likelihood of error is not correlated with classification confidence  $H_N(z)$  (see (4) from the original paper), and thus is not introspective. The *optimal* classifier has the same overall error rate, but makes all of its mistakes with high uncertainty, and is therefore more introspective.

real classifiers in the original paper and below.

### III. RESULTS

In Figure 3 we show the number of each type of outcome (true positives and negatives, false positives and negatives) for the decisions made with confidence greater than a given threshold, this time for the idealised classifiers. The intuition here is that if we have an additional *safe* action such as waiting and gathering more data or asking a human for guidance, we might use it for all test data that the robot perceives to be above a certain threshold of uncertainty. In that case, we are looking at how many correct and incorrect classifications (and therefore decisions) the robot considered safe.

Figure 3 complements Figures 7, 8, and 9 in the original paper, and serves as comparison between real and idealised classifiers. Note how all the real classifiers, with the exception of the random forest, are more confident about both true and false outcomes than the *optimal* classifier up to an uncertainty threshold of 0.8. Considering only the false outcomes (bottom row), most real classifiers lie somewhere between the two idealised classifiers, with the linear SVM and logitboost being more similar to the less introspective *uniform* classifier. Overall, the GPC-based classifiers are closest to the *optimal* behaviour.

The revised graphs for the original Figures 10, 11, and 12 are shown in Figures 5a, 5b, and 5c of this corrigendum. Notice that the SVM curves in pink and yellow on the left-hand-side graphs are now smooth. Removing the ‘blind spot’ in the SVM probabilities improves their high-confidence decisions, particularly in the case of the non-linear SVM.

In Figure 5a we see that the decisions made by the linear SVM are improved, but it still incurs higher costs than all classifiers save logitboost. In Figure 5b the two SVMs still make bad decisions, but not as catastrophically as in the original paper. Figure 5c shows the same bad decisions from the linear SVM as before, but as a result of including more runs, we uncover some catastrophic decisions by the

linear GPC. Overall, each of the three linear classifiers makes catastrophic decisions in at least one of the three data sets.

We show the high-confidence decision-making capabilities of the idealised classifiers in Figure 5d. The *uniform* classifier makes catastrophic mistakes much as the real classifiers do, because it too is capable of making high-confidence errors. The *optimal* classifier, however, incurs a lower (or equal) total cost than the less introspective *uniform* classifier regardless of the choice of cost ratio. The costs asymptote to the maximum possible number of false positives, each worth a cost of 1. The more introspective *optimal* classifier makes decisions which are truer to the chosen loss function.

In Figure 4 we show the replacements for Figure 13 in the original paper, with the addition of the comparison with the idealised classifiers. This figure serves to show whether errors are made with high uncertainty, as is desirable for an introspective classifier. We see that the multi-discriminant classifiers do make their mistakes with higher uncertainty than the others in each of the three data sets, where the KITTI data set is the most challenging. This is consistent with the multi-discriminant GPC-based classifiers making poor high-confidence decisions in the KITTI data set alone. The idealised classifiers demonstrate that the single-discriminant classifiers are overconfident relative to the non-introspective *uniform* classifier.

Overall, no single real classifier behaves like the *optimal* idealised classifier, avoiding all high-cost errors. The random forest avoids high-cost errors only by being uncertain about all decisions. These findings indicate that while the multi-discriminant classifiers appear to be more introspective than single-discriminant classifiers, none presented here are consistently introspective across all three data sets and the differences do not seem to make a tangible difference in decision making.

### IV. CONCLUSIONS

The third-class experiment presented in the original paper and extended here with the introduction of the idealised classifiers indicates that the multi-discriminant GPCs are slightly more introspective than the SVMs, on the basis that the SVMs are overconfident. However, the difference in behaviour in the decision-making experiments is not overwhelming. No real classifier benchmarked as part of this study is capable of consistently avoiding catastrophic outcomes in all three data sets presented. This could be due to contributory effects which we do not control in this experiment, for instance the probabilistic calibrations used and how they are trained, or the choice of kernel. Evaluating the relevance of these factors is a vein for further investigation.

The use of idealised classifiers serve as baselines for the behaviour of the real classifiers. We find that most real classifiers behave somewhere between the more introspective *optimal* and the non-introspective *uniform* idealisations. In Figure 4 the single-discriminant classifiers such as the SVMs and logitboost appear to be more over-confident than the multi-discriminant GPCs and random forests.

We conclude from the idealised classifiers that introspection is crucial in decision making, but that none of the real

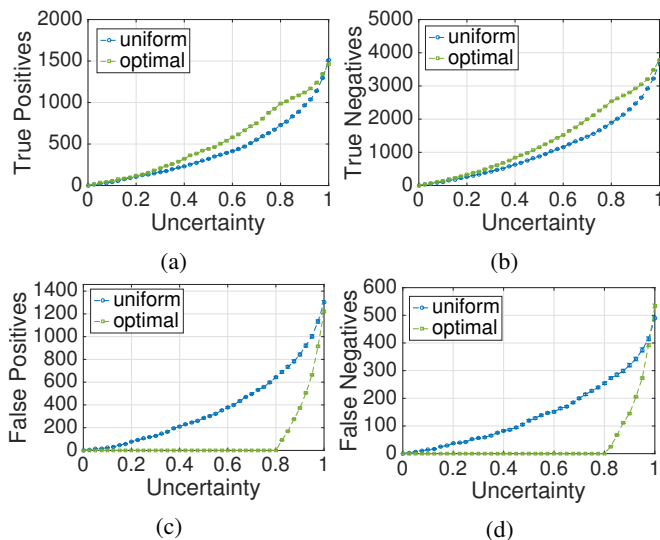


Fig. 3: Idealised classifiers: cumulative frequency plots of classification confusion (true positives, true negatives, false positives, and false negatives) against classification uncertainty. A more introspective classifier is one that simultaneously exhibits higher uncertainty when processing difficult instances (bottom right corner for false positives and negatives) and is more confident when it is correct (top left corner for true positives and negatives). Here we confirm that the *optimal* is the more introspective of the two idealised classifiers. For each run, the classifiers generated 2,000 positive and 5,000 negative measurements. We show the mean and standard error over 10 independent runs.

classifiers benchmarked here are introspective enough to allow them to avoid catastrophic decisions in all three data sets. The benchmarking of other frameworks (e.g. deep architectures) remains further work. In addition, we are also exploring the notion of whether a framework can be designed or trained specifically such that its introspective capacity is improved.

#### REFERENCES

H. Grimmer, R. Triebel, R. Paul, and I. Posner. Introspective Classification for Robot Perception. *International Journal of Robotics Research (IJRR)*, 57:000–000, June 2016.

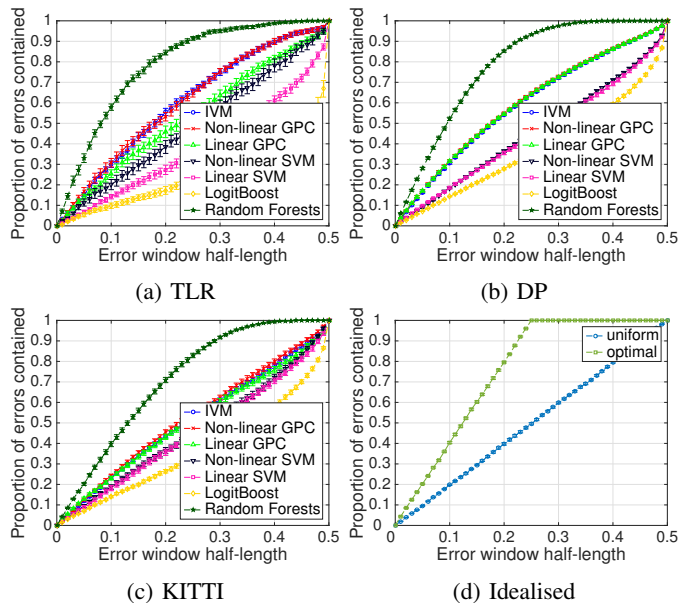
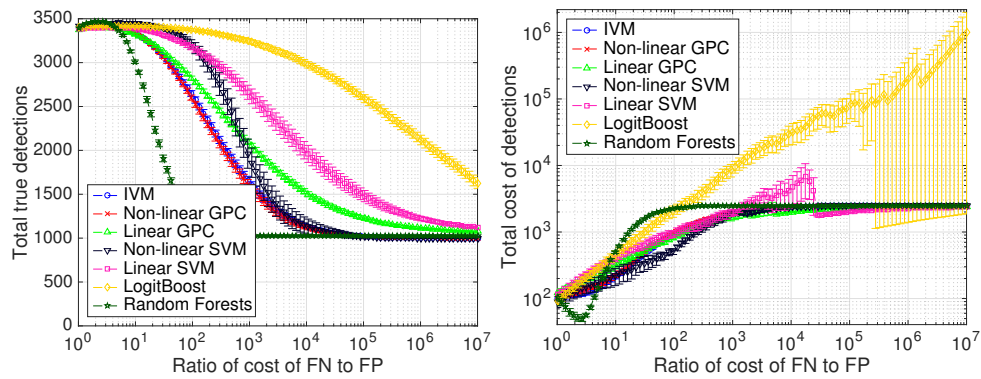
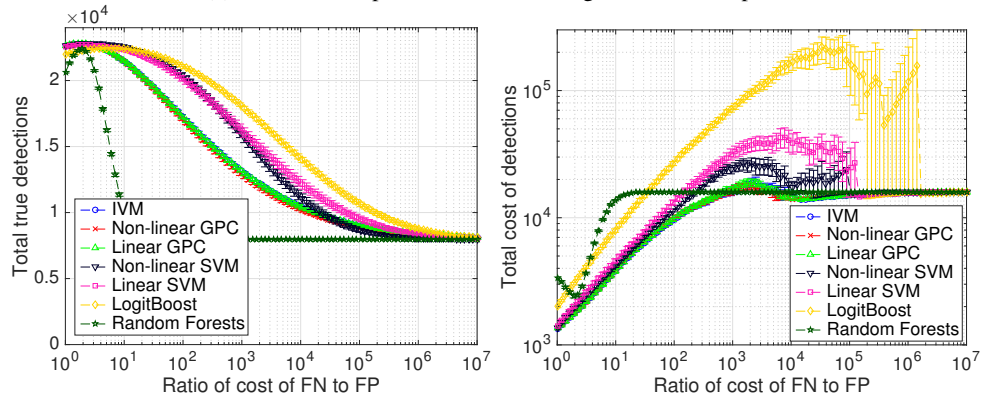


Fig. 4: We show the proportion of errors made by a classifier contained within a region around  $p(C_2) = 0.5$ . The  $x$ -axis shows the size of that region, the orange window as described in Figure 4 of the original paper. To generate these curves, we randomly sample 1,000 positive and 1,000 negative test data and count the number of errors within a certain window. Note that a classifier which is uncertain when it makes mistakes will be closer to the top left of each plot. The random forest performs very well in this respect, although it makes *all* decisions with large uncertainty. This shows one side of the introspective coin. We show the mean and standard error over 10 independent runs.

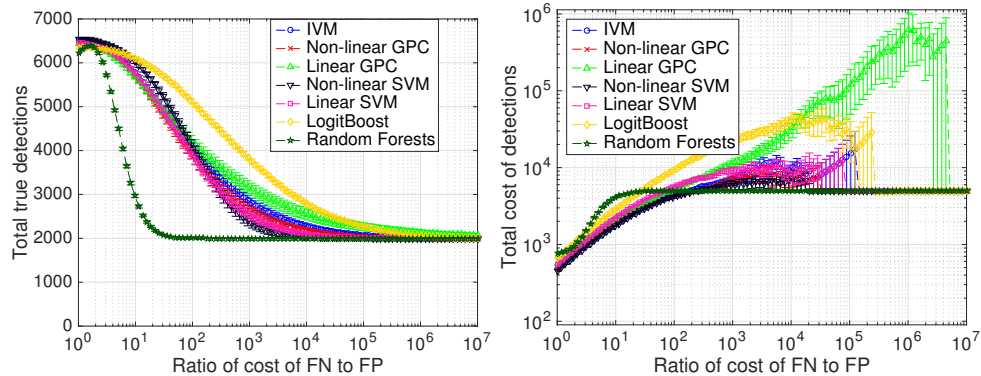




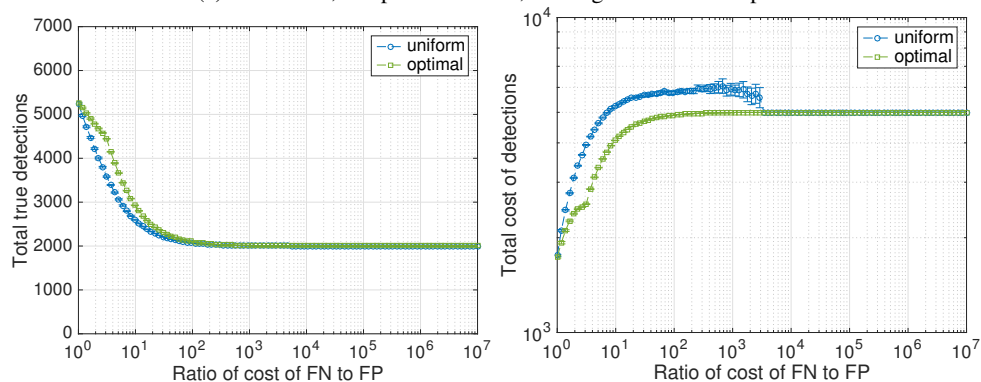
(a) TLR - 1,000 positive and 2,500 negative test examples.



(b) Daimler Pedestrian - 8,000 positive and 16,000 negative test examples.



(c) KITTI - 2,000 positive and 5,000 negative test examples.



(d) Idealised classifiers - 2,000 positive and 5,000 negative test examples.

Fig. 5: High-confidence decision making. The  $x$ -axes represent the cost of a false negative error (FN, e.g. missing a traffic light, pedestrian, etc), while the cost of a false positive error (FP) is held at 1. On the left we show the number of correct decisions made (positive and negative) as we vary the cost of a false negative, and on the right we show the total cost of all decisions (correct decisions carry 0 cost). Ideal behaviour on the left is to smoothly become more conservative (which requires making fewer correct decisions) as the cost increases, and on the right the ideal is to incur minimal total cost at every point on the  $x$ -axis. We show the mean and standard error over 10 independent runs.