Information Theoretic Active Learning for Classification and Preference Modelling

Anonymous Author(s)
Affiliation
Address
email

Abstract

Information theoretic active learning has been widely studied for probabilistic models. For simple regression an optimal myopic policy is easily tractable. However, for other common tasks, such as classification, the optimal solution is more complex. Many approaches have been proposed that include computing approximate posterior entropies, sampling, or using related quantities in non-probabilistic models. The contributions of this paper are threefold: Firstly, we propose an approach that expresses information gain in terms of predictive entropies and discuss the computational advantages this offers compared to other methods. Secondly, we propose a novel algorithm for active learning for the popular Gaussian Process classifier (GPC). Notably our algorithm works with all known approximate inference methods for GPC and allows for active learning of hyperparameters too. Finally we extend the the algorithm to Gaussian process-based binary preference learning.

1 Introduction

In most machine learning applications, the learner passively collects data with which it makes inferences about its environment. In active learning, however, the learner can seek out the most useful measurements to be trained on. The goal of active learning is to produce the most accurate model with the least possible data; this is closely related to the statistical field of optimal experimental design. With the advent of the internet and expansion in storage facilities vast quantities of unlabelled data have become available, but it is often costly to obtain labels; searching for the most useful data in this vast space calls for efficient active learning algorithms.

Two approaches to active learning utilise decision and information theory [**?**, **?**]. The former minimizes the expected losses encountered after making decisions based on the data collected i.e. minimize the Bayes posterior risk [**?**]. Maximising performance under test is the ultimate objective of most learners, however, evaluating this objective can be very hard. For example the methods proposed in [**?**, **?**] for classification are in general very expensive to compute. Furthermore, we may not know the loss function or test distribution in advance, we may want our model to perform well on a wide variety of loss functions. In extreme scenarios, such as exploratory data analysis, or visualisation, losses may be very hard to quantify. This motivates the use of information theoretic active learning which is agnostic to the decision task at hand, but tries to increase model certainty as quickly as possible, usually using Shannon's entropy as a measure of uncertainty.

In this paper we focus on probabilistic classification, we present an algorithm that applies the full information theoretic criterion to Gaussian Processes Classification (GPC). GPC is powerful, non-parametric kernel-based model, and poses an interesting problem for information-theoretic active learning because parameter space is infinite dimensional and the posterior distribution is analytically intractable. We present the information theoretic approach to active learning in Section 2. In Section 3 we briefly review other approaches and their suitability for classification relative to our information-

theoretic approach. We detail our algorithm and show how to extend our GPC approach to yield a novel method for active preference learning (Section 4). We present results on a variety of datasets in Section 5 and conclude the paper in Section 6.

2 Bayesian Information Theoretic Active Learning

We consider a fully discriminative model where the goal of active learning is to discover the dependence of some variable $\boldsymbol{y} \in \mathcal{Y}$ on an input variable $\boldsymbol{x} \in \mathcal{X}$ by interactively querying the system with inputs $\boldsymbol{x}_i \in \mathcal{X}$ and observing the system's response $\boldsymbol{y}_i$.

Within a Bayesian framework we assume existence of some latent parameters, $\boldsymbol{\theta}$, that control the dependence between inputs and outputs, $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$. After having observed data $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, we have a posterior distribution over the parameters, $p(\boldsymbol{\theta}|\mathcal{D})$. The goal of information theoretic active learning is to minimize the uncertainty about the parameters using the well studied Shannon's entropy [**?**], i.e. select a new set of datapoints $\mathcal{D}'$ that satisfy $\arg\min_{\mathcal{D}'} H[\boldsymbol{\theta}|\mathcal{D}'] = -\int p(\theta|\mathcal{D}') \log p(\theta|\mathcal{D}') \mathrm{d}\theta$. Unfortunately, solving this problem in general in NP-hard; however, it has been shown that a myopic policy can perform near-optimally [**?**, **?**]. The myopic strategy sequentially selects points to greedily minimize the objective. Therefore, the objective function, first proposed in [**?**] is to seek the a data point $\boldsymbol{x}$ that satisfies:

$$\arg\max_{\boldsymbol{x}} H[\boldsymbol{\theta}|\mathcal{D}] - \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y}|\boldsymbol{x}\mathcal{D})} \left[H[\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{x}, \mathcal{D}]\right] \tag{1}$$

Note that expectation over the unseen output $\boldsymbol{y}$ is required. Eqn. (1) poses two difficulties: firstly, if we search k potential queries, $\boldsymbol{x}$, and the output, $\boldsymbol{y}$, may take on l values, each kl posterior updates are required to compute the objective for each $\boldsymbol{x}$ in question. Secondly, calculating entropies in parameter space may be hard. Often we may only be able to estimate entropies in parameter space using samples from the posterior, which is notoriously difficult [**?**]; or by gridding up parameter space which scales exponentially with dimensionality of $\boldsymbol{\theta}$. Worse still, for non-parametric processes parameter space is infinite dimensional so Eqn. (1) becomes poorly defined. [**?**, **?**, **?**] use this objective but must make approximations to the complicated entropy term. However, if we note that the objective in Eqn. (1) is equivalent to the conditional mutual information between the unknown output and the parameters, $I[\theta, \boldsymbol{y}|\boldsymbol{x}, \mathcal{D}]$ then it is simple to show that the objective can be rearranged to compute entropies in $\boldsymbol{y}$ space:

$$\arg\max_{\boldsymbol{x}} H[\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}] - \mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\mathcal{D})} \left[H[\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}]\right] \tag{2}$$

Eqn. (2) overcomes the aforementioned challenges. Entropies are now calculated in, usually low dimensional, output space. Also $\boldsymbol{\theta}$ is now conditioned only on $\mathcal{D}$, so we do not need to update the posterior for every possible outcome, saving a factor of kl posterior updates. Equation (2) provides us with an intuition about the objective; we seek the $\boldsymbol{x}$ for which the model is marginally most uncertain about $\boldsymbol{y}$ (high $H[\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}]$), but for which individual setting of the parameters are confident (low $\mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\mathcal{D})} \left[H[\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}]\right]$). This can be interpreted as seeking the $\boldsymbol{x}$ for which the parameters under the posterior disagree about the outcome the most, so refer to this objective as Bayesian Active Learning by Disagreement (BALD). We present a way to apply Eqn. (2) directly to GPC and preference learning. This method is inductive, i.e. we do not assume anything about the test set as opposed to transductive algorithms which know the distribution of the test data.

3 Related Methodologies

In this section we briefly review some of the very many related algorithms that are applicable to classification and relate them to the full information theoretic objective (2).

Information Theoretic: Other work that uses rearrangement to data space (Eqn. (2)) include Maximum Entropy Sampling (MES) [**?**]. MES was proposed for regression models with input-independent observation noise. Although Eqn. (2) is used, the second term is constant because of

input independent noise. and so can be ignored. For heteroscedastic regression or classification, MES is inappropriate; it fails to differentiate between model uncertainty and observation uncertainty (about which our model may be confident). Many toy demonstrations show the 'information based' active learning criterion performing pathologically in classification by repeatedly querying points close the decision boundary or in regions of high observation uncertainty e.g. those presented in [**?**, **?**]. This is because MES is inappropriate, BALD distinguishes between observatio and model uncertainty and will eliminate these problems as we will show.

Further mutual-information based objective functions are presented in [**?**, **?**], who seek to maximise mutual information between the variable being measured and the variable of interest. Fuhrmann [**?**] applies this to linear Gaussian models and acoustic arrays, Ertin *et al.* [**?**] to a communications channel. Although closely related, these objectives do not work with the model parameters and are not applied to classification. [**?**, **?**, **?**] also use mutual information. They specify points of interest in advance and maximise the expected mutual information between the points of interest at the observed locations. Although this is a objective is promising for regression, it is not computable for models with input-dependent observation noise, such as classification; furthermore, it is not inductive.

Finally, the IVM [**?**] algorithm was designed for sub-sampling a dataset to be used to train a GP. It may not fall under the term 'active learning' because all $\boldsymbol{y}$ values are available a priori. Their objective is Eqn. (1), however the algorithm is not based on a rearrangement to data space (Eqn. (2)). Therefore, posterior entropy calculations are made approximately on the n dimensional subspace corresponding to the n observed datapoints using the GP covariance matrix and kl posterior updates are required ([**?**] proposes using Assumed Density Filtering to do this quickly).

We have briefly reviewed several information-theoretic based algorithms, but as far as the authors are aware our paper is the first to develop an efficient algorithm applying the fulll information theoretic criterion (Eqn. (2)) to probabilistic classification.

Decision theoretic: We briefly mention a few decision theoretic approaches to classification. Two closely related algorithms, [**?**, **?**], seek to minimize the expected misclassification probability on all seen and future data (sometimes with costs associated). These methods observe the locations of the test points and their objective functions are monotonic in the predictive entropies at the test points. [**?**] also includes an empirical error term that can yield pathological behaviour (we investigate this experimentally). These approaches are computationally expensive, requiring kl posterior updates. They are also they are transductive because they look at the locations of the test data; designing an inductive, decision-theoretic algorithm is an open, hard problem as it must require potentially expensive integration over possible test data distributions.

Non-probabilistic Certain non-probabilistic methods have close analogues to information theoretic active learning. Perhaps the most ubiquitous is active learning for SVMs [**?**, **?**] where the volume of version space is used as a proxy for the posterior entropy. If a uniform (improper) prior is used with a deterministic classification likelihood it is easy to show that the log volume of version space and Bayesian posterior entropy are equivalent. However, just as Bayesian posteriors become intractable after observing many datapoints, version space too can become very complicated. [**?**] proposes approximating version space with a simple shape, such as a hypersphere. This closely resembles approximating a Bayesian posterior using a Gaussian distribution via the Laplace or EP approximations. [**?**] sidesteps the problem by working with predictions. The algorithm, Query by Committee (QBC), samples parameters from version space (committee members), they vote on the outcome of each $\boldsymbol{x}$ in question. The $\boldsymbol{x}$ with the most balanced vote is selected; this is termed the 'principle of maximal disagreement'. If BALD is used with a sampled posterior, query by committee is implemented but with a probabilistic measure of disagreement. QBC's deterministic vote criterion discards confidence in the predictions and so can exhibit the same pathologies as MES. We present a summary of the methods discussed in this section in the Supplementary material.

4 Gaussian Processes for Classification and Preference Learning

Here we present the application of BALD to Gaussian process classification (GPC). GPs are powerful and highly popular non-parametric tools for regression and classification. GPC appears to be an especially challenging problem for information-theoretic active learning because the parameter space

is infinite, however, by using (2) we are able to fully calculate the relevant information quantities without having to work out entropies of infinite dimensional objects. The probabilistic model underlying GPC is as follows:

$$f \sim \mathrm{GP}(\mu(\cdot), k(\cdot,\cdot)) \qquad \boldsymbol{y}|\boldsymbol{x}, f \sim \mathrm{Bernoulli}(\Phi(f(\boldsymbol{x}))) \tag{3}$$

The latent parameter, now called f (previously denoted as $\boldsymbol{\theta}$), is a function $\mathcal{X} \to \mathbb{R}$, and is assigned a Gaussian process prior with mean $\mu(\cdot)$ and covariance function $k(\cdot,\cdot)$. We consider the probit case where given the value of f, y takes a Bernoulli distribution with probability $\Phi(f(\boldsymbol{x}))$, and Φ is the cumulative distribution function of the normal distribution. For further details on GPC see [**?**].

Inference in the GPC model is intractable; given some observations $\mathcal{D}$, the posterior over f becomes non-Gaussian and complicated. The most commonly used approximate inference methods – EP, Laplace approximation, Assumed Density Filtering and sparse methods – all approximate the posterior by a Gaussian [**?**]. Throughout this section we will assume that we are provided with such a Gaussian approximation from one of these methods, though the active learning algorithm does not care which. In our derivation we will use $\overset{1}{\approx}$ to indicate where such an approximation is exploited.

Now, we will compute the informativeness of a query $\boldsymbol{x}$ using Eqn. (2). The entropy of the binary output variable y given a fixed f can be expressed in terms of the binary entropy function h:

$$H[y|\boldsymbol{x}, f] = h\left(\Phi(f(\boldsymbol{x}))\right), \qquad h(p) = -p\log p - (1-p)\log(1-p) \tag{4}$$

We now have to compute expectations over the posterior. Using a Gaussian approximation to the posterior, for each $\boldsymbol{x}$, $f_{\boldsymbol{x}} = f(\boldsymbol{x})$ will follow a Gaussian distribution with mean $\mu_{\boldsymbol{x},\mathcal{D}}$ and variance $\sigma^2_{\boldsymbol{x},\mathcal{D}}$. To compute the two terms in Eqn. (2) we have to compute two entropy quantities. The first term in Eqn. (2), $H[y|\boldsymbol{x},\mathcal{D}]$ can be handled analytically:

$$H[y|\boldsymbol{x},\mathcal{D}] \overset{1}{\approx} h\left(\int \Phi(f_{\boldsymbol{x}})\mathcal{N}(f_{\boldsymbol{x}}|\mu_{\boldsymbol{x},\mathcal{D}}, \sigma^2_{\boldsymbol{x},\mathcal{D}})df_{\boldsymbol{x}}\right) = h\left(\Phi\left(\frac{\mu_{\boldsymbol{x},\mathcal{D}}}{\sqrt{\sigma^2_{\boldsymbol{x},\mathcal{D}}+1}}\right)\right) \tag{5}$$

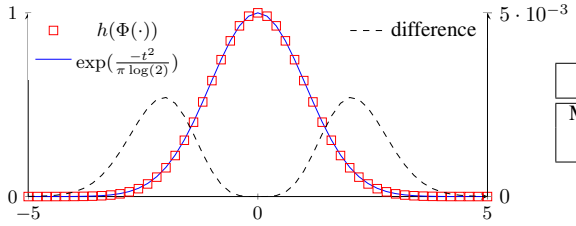
The second term, $\mathbb{E}_{f\sim p(f|\mathcal{D})}\left[H[\boldsymbol{y}|f]\right]$ can be computed approximately as follows

$$\begin{aligned}\mathbb{E}_{f\sim p(f|\mathcal{D})}\left[H[\boldsymbol{y}|f]\right] &\overset{1}{\approx} \int h(\Phi(f_{\boldsymbol{x}}))\mathcal{N}(f_{\boldsymbol{x}}|\mu_{\boldsymbol{x},\mathcal{D}}, \sigma^2_{\boldsymbol{x},\mathcal{D}})df_{\boldsymbol{x}} \\ &\overset{2}{\approx} \int \exp\left(-\frac{f^2_{\boldsymbol{x}}}{\pi\ln 2}\right)\mathcal{N}(f_{\boldsymbol{x}}|\mu_{\boldsymbol{x},\mathcal{D}}, \sigma^2_{\boldsymbol{x},\mathcal{D}})df_{\boldsymbol{x}} \\ &= \frac{C}{\sqrt{\sigma^2_{\boldsymbol{x},\mathcal{D}}+C^2}}\exp\left(-\frac{\mu^2_{\boldsymbol{x},\mathcal{D}}}{2\left(\sigma^2_{\boldsymbol{x},\mathcal{D}}+C^2\right)}\right)\end{aligned} \tag{6}$$

where $C = \sqrt{\frac{\pi\ln 2}{2}}$. The first approximation, $\overset{1}{\approx}$, reflects the Gaussian approximation to the posterior. The integral in the left hand side of Eqn. (6) is hard to compute; $h(\Phi(f_{\boldsymbol{x}}))$ must be integrated against a Gaussian distribution. However, if we perform a Taylor expansion on $\ln h(\Phi(f_{\boldsymbol{x}}))$ (see supplementary material) we can see that it can be approximated up to $\mathcal{O}(f^3_{\boldsymbol{x}})$ by a squared exponential curve, $\exp(-f^2_{\boldsymbol{x}}/\pi\ln 2)$. We will refer to this approximation as $\overset{2}{\approx}$. Now we can apply the standard convolution formula for Gaussians to finally get a closed form expression for both terms of Eqn. (2).

Fig. 1 depicts the striking accuracy of this simple approximation. The maximum possible error that will be incurred when using this approximation is if $\mathcal{N}(f_{\boldsymbol{x}}|\mu_{\boldsymbol{x},\mathcal{D}}, \sigma^2_{\boldsymbol{x},\mathcal{D}})$ is centred at $\mu_{\boldsymbol{x},\mathcal{D}} = \pm 2.05$ with $\sigma^2_{\boldsymbol{x},\mathcal{D}}$ tending to zero (see Fig. 1, absolute error - - -); even this yields only a 0.27% error in the integral in Eqn.(6). The authors are unaware of previous use of this simple and useful approximation in this context. In Section 5 we investigate experimentally the information lost from approximations $\overset{1}{\approx}$ and $\overset{2}{\approx}$ as compared to the golden standard of extensive Monte Carlo simulation.

To summarise, the BALD algorithm for Gaussian process classification consists of two steps. First it applies an approximate inference algorithm to obtain the posterior predictive mean $\mu_{\boldsymbol{x},\mathcal{D}}$ and $\sigma_{\boldsymbol{x},\mathcal{D}}$ for each point of interest $\boldsymbol{x}$. Then, it selects a query $\boldsymbol{x}$ that maximises the following objective function:



	MCMC	EP ($\overset{1}{\approx}$)	Laplace ($\overset{1}{\approx}$)
MC	0	7.51 ± 2.51	41.57 ± 4.02
$\overset{2}{\approx}$	0.16 ± 0.05	7.43 ± 2.40	40.45 ± 3.67

Figure 1: *Left:* Analytic approximation ($\overset{1}{\approx}$) to the binary entropy of the error function (□) by a squared exponential (——). The absolute error (- - -) remains under $3 \cdot 10^{-3}$. *Right:* Percentage approximation error (± 1 s.d.) for different methods of approximate inference (*columns*) and approximation methods for evaluating Eqn.(6) (*rows*). The results indicate that $\overset{2}{\approx}$ is a very accurate approximation; EP causes some loss and Laplace significantly more, which is in line with the comparison presented in [**?**].

$$\mathrm{h}\left(\Phi\left(\frac{\mu_{\boldsymbol{x},\mathcal{D}}}{\sqrt{\sigma^2_{\boldsymbol{x},\mathcal{D}}+1}}\right)\right) - \frac{C\exp\left(-\frac{\mu^2_{\boldsymbol{x},\mathcal{D}}}{2\left(\sigma^2_{\boldsymbol{x},\mathcal{D}}+C^2\right)}\right)}{\sqrt{\sigma^2_{\boldsymbol{x},\mathcal{D}}+C^2}} \tag{7}$$

For most practically relevant kernels, the objective (7) is smooth, and differentiable function of $\boldsymbol{x}$, so gradient-based optimisation procedures can be used to find the maximally informative query.

4.1 Learning and Exploring Hyperparameters

Suppose we want to perform active learning by minimising information about a subset of parameters $\boldsymbol{\theta}^+$ of central interest, but do not care about another set $\boldsymbol{\theta}^-$. By integrating Eqn. (1) over the nuisance parameters, θ^-, we may re-derive the following BALD objective function:

$$H\left[\mathbb{E}_{p(\boldsymbol{\theta}^+,\boldsymbol{\theta}^-|\mathcal{D})}\left[\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}^+,\boldsymbol{\theta}^-\right]\right] - \mathbb{E}_{p(\boldsymbol{\theta}^+|\mathcal{D})}\left[H\left[\mathbb{E}_{p(\boldsymbol{\theta}^-|\boldsymbol{\theta}^+,\mathcal{D})}[\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}^+,\boldsymbol{\theta}^-]\right]\right] \tag{8}$$

In the context of GP models, hyperparameters typically control the smoothness or spatial length-scale of functions. If we maintain a posterior distribution over these hyperparameters, which we can do e. g. via Hamiltonian Monte Carlo, we can choose either to treat them as nuisance parameters θ^- and use Eq. 8, or to include them in θ^+ and perform active learning over them as well. In certain cases, such as automatic relevance determination[**?**], it may even make sense to treat hyperparameters as variables of primary interest, and the function f itself as nuisance parameter θ^-.

4.2 Preference Learning

In preference learning our dataset consists for pairs of items $(\boldsymbol{u}_i, \boldsymbol{v}_i) \in \mathcal{X}^2$ with binary labels, $y_i \in \{0, 1\}$. $y_i = 1$ means instance $\boldsymbol{u}_i$ is preferred to $\boldsymbol{v}_i$, denoted $\boldsymbol{u}_i \succ \boldsymbol{v}_i$. The task is to predict the preference relation between any $(\boldsymbol{u}, \boldsymbol{v})$. Ultimately the problem is a special case of building a classifier $h : \mathcal{X}^2 \mapsto \{0, 1\}$. We now briefly review the Bayesian approach of Chu *et al.* [**?**] who use a latent preference function f, over which a zero-mean GP prior is defined. When predicting preference, $\boldsymbol{u}_i \succ \boldsymbol{v}_i$ whenever $f(\boldsymbol{u}_i) + \delta_{u_i} > f(\boldsymbol{v}_i) + \delta_{v_i}$, where $\delta_{u_i}, \delta_{v_i}$ denote additive Gaussian evaluation noise. Under this model, the likelihood of f becomes:

$$\mathbb{P}[y = 1|(\boldsymbol{u}_i, \boldsymbol{v}_i), f] = \mathbb{P}[\boldsymbol{u}_i \succ \boldsymbol{v}_i|f] = \Phi\left(\frac{f(\boldsymbol{u}_i) - f(\boldsymbol{v}_i)}{\sqrt{2}\sigma_{noise}}\right) \tag{9}$$

It can be assumed w.l.o.g. that $\sqrt{2}\sigma_{noise} = 1$. Observe, that the likelihood only depends on the difference between $f(\boldsymbol{u})$ and $f(\boldsymbol{v})$. We therefore define $g(\boldsymbol{u}, \boldsymbol{v}) = f(\boldsymbol{u}) - f(\boldsymbol{v})$, and do inference entirely in terms of g, for which the likelihood becomes the same as for probit classification:

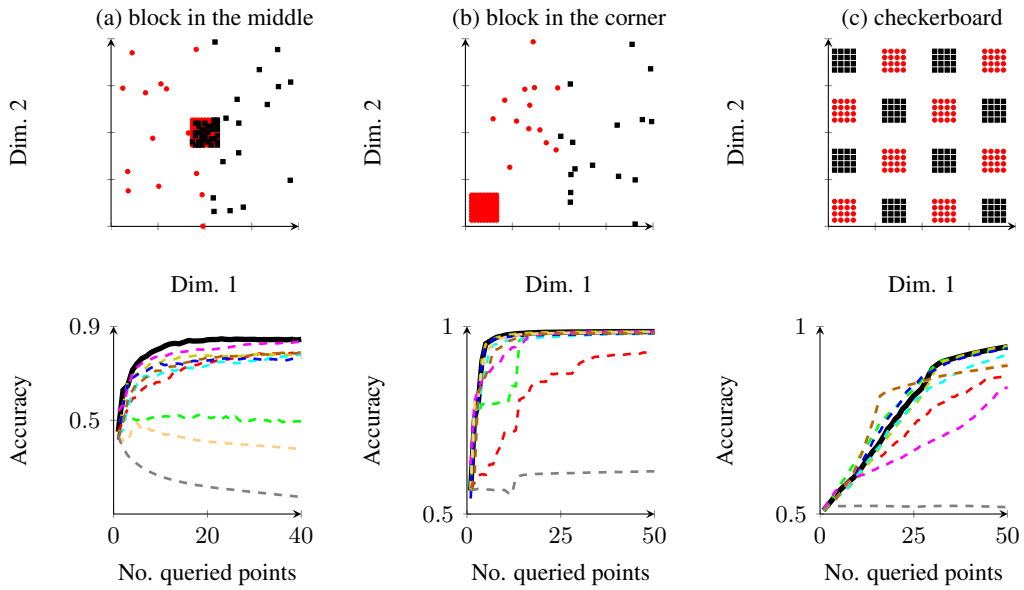


Figure 2: *Top:* Artificial datasets used in our evaluation of active learning methods. Exemplars of the two classes are shown with black squares (▪) and red circles (•). *Bottom:* Results of active learning with nine methods: random query (- - -), BALD (——), MES (- - -), QBC with the vote criterion with 2 (QBC_2, - - -) and 100 (QBC_{100}, - - -) committee members, active SVM (- - -), IVM (- - -), Kapoor *et al.* [**?**] (- - -), Zhu *et al.* [**?**] (- - -) and empirical error (- - -).

$y|\boldsymbol{u}, \boldsymbol{v}, f \sim \text{Bernoulli}(\Phi(g(\boldsymbol{u}, \boldsymbol{v})))$. We observe that a GP prior is induced on g because it is formed by performing a linear operation on f, for which we have a GP prior already $f \sim \text{GP}(0, k)$. We can derive the induced covariance function of g as (derivation in the Supplementary material):

$$k_{pref}((\boldsymbol{u}_i, \boldsymbol{v}_i), (\boldsymbol{u}_j, \boldsymbol{v}_j)) = k(\boldsymbol{u}_i, \boldsymbol{u}_j) + k(\boldsymbol{v}_i, \boldsymbol{v}_j) - k(\boldsymbol{u}_i, \boldsymbol{v}_j) - k(\boldsymbol{v}_i, \boldsymbol{u}_j) \tag{10}$$

Note, that this kernel k_{pref} respects the anti-symmetry properties desired for a preference learning scenario, i.e. the value $g(u, v)$ is perfectly anti-correlated with $g(v, u)$, ensuring $\mathbb{P}[\boldsymbol{u} \succ \boldsymbol{v}] = 1 - \mathbb{P}[\boldsymbol{v} \succ \boldsymbol{u}]$ holds. Thus, we can conclude that the GP preference learning framework of [**?**], is equivalent to GPC with a particular class of kernels, that we may call the *preference judgement kernels*. Therefore, our active learning algorithm presented in section 4 for GPC can readily be applied to pairwise preference learning as well.

5 Experiments

Quantifying Approximation Losses: To obtain (7) we made two approximations: we perform approximate inference ($\overset{1}{\approx}$), and we approximated the binary entropy of the Gaussian CDF by a squared exponential ($\overset{2}{\approx}$). Both of these can be substituted with Monte Carlo approximation, enabling us to compute an asymptotically unbiased estimate of the expected information gain. Using extensive Monte Carlo as the 'gold standard', we can evaluate how much we loose by applying these approximations. We quantify approximation error as:

$$\frac{\max_{\boldsymbol{x} \in \mathcal{P}} I(\boldsymbol{x}) - I(\arg\max_{\boldsymbol{x} \in \mathcal{P}} \hat{I}(\boldsymbol{x}))}{\max_{\boldsymbol{x} \in \mathcal{P}} I(\boldsymbol{x})} \cdot 100\% \tag{11}$$

where I is the objective computed using Monte Carlo, $\hat{I}$ is the approximate objective. These experiments were run on the *cancer* dataset, results are shown and discussed in Figure 1.

Pool based active learning: We test BALD for GPC and preference learning in the pool-based setting i.e. selecting x values from a fixed set of data-points. We compare to eight other algorithms discussed in this paper: random sampling, MES, QBC, SVM with version space approximation [**?**], decision theoretic approaches in [**?**, **?**] and directly minimizing expected empirical error (empirical error is not a widely used method, but is included for analysis of [**?**]).

We consider three artificial, but challenging, datasets. The first of which is similar to the *checkerboard* dataset used in [**?**], and is designed to test the algorithm's capabilities to find multiple disjoint islands of points from one class. The second, *block in the corner*, has a block of uninformative points far from the decision boundary, and the third, *block in the middle*, has a block of noisy points on the decision boundary: a strong active learning algorithm should avoid these uninformative regions. The three datasets and results using each algorithm are depicted in Fig. 2.

In addition to this, we present results on 6 UCI binary classification datasets *australia, crabs, vehicle, isolet, cancer* and *wdbc*. *Letter* is a multiclass dataset for which we select hard-to-distinguish letters E vs. F and D vs. P. For preference learning we use the *cpu, cart* and *kinematics* regression datasets [1] processed to yield a preference task as described in [**?**]. Results are plotted in Fig. 3.

We can see from Figs 2 and 3 that by using BALD we make significant gains over naive sampling in both the classification adn preference learning domains. Relative to other active learning algorithms BALD performs consistently well across all datasets, particularly when avoiding the block of points in Fig. 2 (a). Occasionally e.g. as in Fig. 3 (k,l), it performs poorly on the first couple of queries. In most reporter experiments we have fixed the hyperparamters a priori to the maximum likelihood estimate on the whole pool. This is of course cheating, as it uses information from the whole dataset before starting to select queries, but it provides us with a fair way of comparing various methods, that cannot handle hyperparameter learning. As shown in section 4.1, BALD can accommodate active learning of hyperparameters. For inference over hyperparameters we used Hybrid Monte Carlo, which is an expensive procedure, therefore we only tried it on a fewer number of datasets. On most datasets including hyperparameters in the BALD objective makes little difference, however, on the *cancer* dataset it helps to overcome the initial poor performance of BALD . This is shown in Fig. 3(l).

MES often performs as well as BALD e.g. on Fig. 2 (c), where there is zero noise. It never outperforms BALD though and on noisy datasets (e.g. Fig. 2(a)) performs poorly as expected. QBC provides a close approximation to BALD and usually provides a small decrement in performance. However, there is a large decrease in performance on the noisy artificial dataset caused by the vote criterion not maintaining a notion of inherent uncertainty, like MES. The IVM occasionally performs well, but often exhibits highly pathological behaviour; by observing $\boldsymbol{y}$ values in advance it actively chooses noisy or mislabelled points, thinking them informative. The SVM-based approach exhibits variable performance (it does extremely well on Fig. 3 (f), but very poorly on 2 (c)). Although we have only presented one possible version space approximation here, we find that the performance is greatly effected by the approximation used.

The decision theoretic approaches sometimes perform well, on 2(c) they choose the first 16 points from the centre of each cluster as they are influenced by the surrounding unlabelled points. BALD , being inductive, does not observe the unlabelled points so may not pick points from the centres. On the real datasets though BALD usually performs as well, if not better, despite not having access to the locations of the test points and having a significantly lower computational cost. The Kapoor *et al.* objective sometimes fails badly, this is likely to be because one term in their objective function is the empirical error. The weighting of this term is determined by the relative sizes of the training and test set. Directly minimizing empirical error usually performs very pathologically, picking only 'safe' points, when the Kapoor *et al.* objective assigns to much weight to this term it also fails.

6 Conclusions

We have presented a novel method for applying the full information theoretic active learning criterion to GPC. We present a neat trick that provides a highly accurate analytic approximation to the information theoretic objective. We extend the GPC model to develop a novel preference learning kernel, allowing us to apply our active learning algorithm directly to this domain also. We have shown that the method can naturally handle active learning of kernel hyperparameters, something

[1] Data sets at http://www.liacc.up.pt/ ltorgo/Regression/DataSets.html

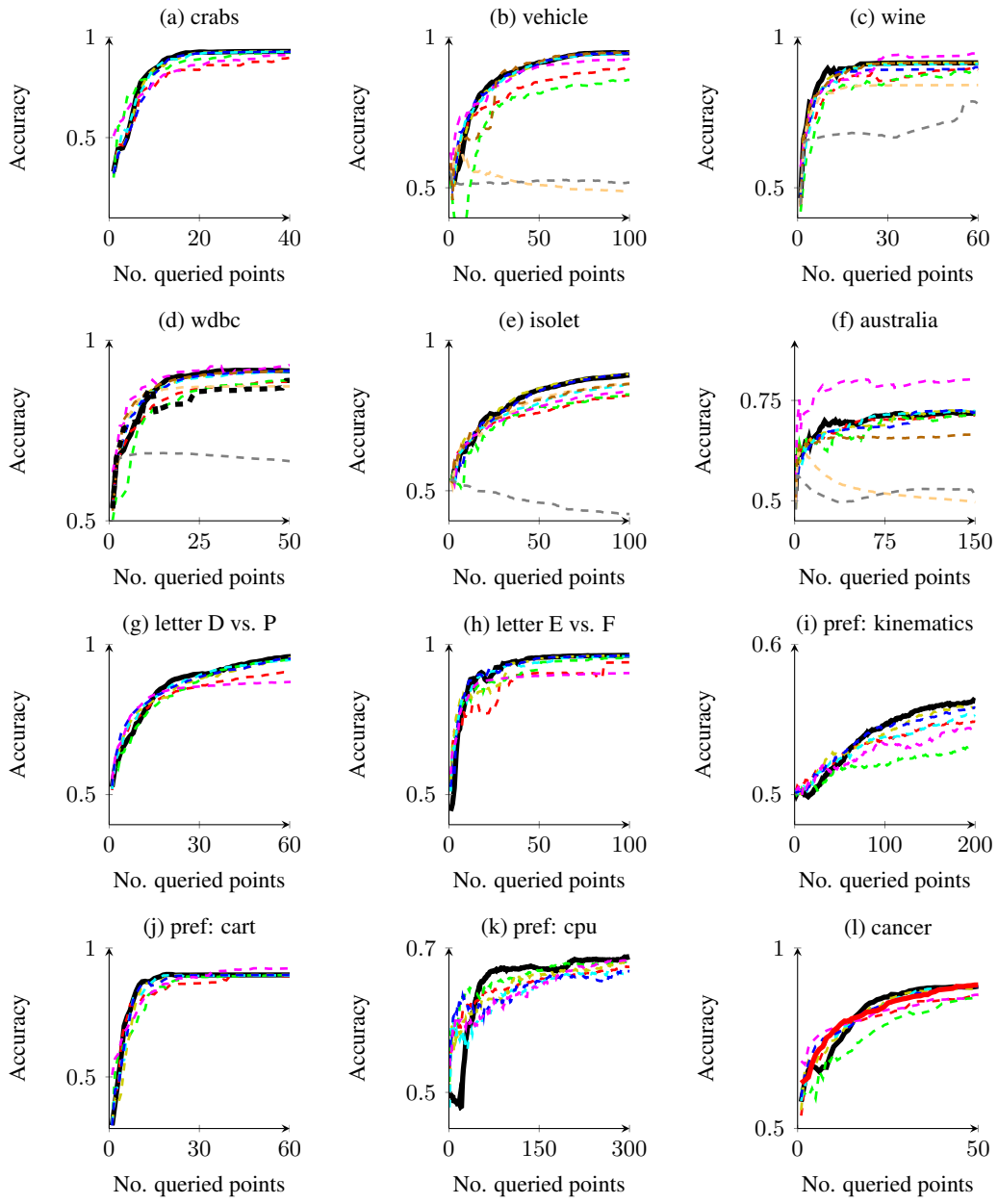


Figure 3: Classification accuracy on classification and preference learning datasets. Methods used are random query (- - -), BALD (—), MES (- - -), QBC with 2 (QBC_2, - - -) and 100 (QBC_{100}, - - -) committee members, active SVM (- - -), IVM (- - -), decision theoretic [**?**] (- - -), semi-supervised [**?**] (- - -) and empicial error (- - -). The decision theoretic methods took a long time to run, so were not completed for all datasets. Plots (a-h) are GPC datasets, (i-k) are preference learning, plot (l) includes BALD with hyperparameter learning (—)
.

which is a hard, mostly unsolved problem for example in SVM active learning. One notable feature of our approach is that it is agnostic to the approximate inference methods used. This allows us to choose from a whole range of approximate inference methods, including EP, Laplace approximation, ADF or even sparse online learning, and thereby to trade off between computational complexity and accuracy. We compare favourably to many other active learning methods for classification, even those that have access to the test data and require much greater computational time.