Bayesian Active Learning for Classification and Preference Learning

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, Máté Lengyel
Computational and Biological Learning Laboratory
University of Cambridge

December 30, 2011

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 tasks and with more complex models, such as classification with nonparametric models, the optimal solution is harder to compute. Current approaches make approximations to achieve tractability. We propose an approach that expresses information gain in terms of predictive entropies, and apply this method to the Gaussian Process Classifier (GPC). Our approach makes minimal approximations to the full information theoretic objective. Our experimental performance compares favourably to many popular active learning algorithms, and has equal or lower computational complexity. We compare well to decision theoretic approaches also, which are privy to more information and require much more computational time. Secondly, by developing further a reformulation of binary preference learning to a classification problem, we extend our algorithm to Gaussian Process preference learning.

1 Introduction

In most machine learning systems, the learner passively collects data with which it makes inferences about its environment. In active learning, however, the learner seeks the most useful measurements to be trained upon. The goal of active learning is to produce the best 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 of storage facilities, vast quantities of unlabelled data have become available, but it can be costly to obtain labels. Finding the most useful data in this vast space calls for efficient active learning algorithms.

Two approaches to active learning are to use decision and information theory [Kapoor et al., 2007, Lindley, 1956]. The former minimizes the expected losses encountered after making decisions based on the data collected i.e. minimize the Bayes posterior risk [Roy and McCallum, 2001]. 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 [Kapoor et al., 2007, Zhu et al., 2003] for classification are in general expensive to compute. Furthermore, one may not know the loss function or test distribution in advance, or may want the model to perform well on a variety of loss functions. In extreme scenarios, such as exploratory data analysis, or visualisation, losses may be very hard to quantify.

This motivates information theoretic approaches to active learning, which are agnostic to the decision task at hand and particular test data, this is known an inductive approach. They seek to reduce the number of feasible models as quickly as possible, using either heuristics (e.g. margin sampling [Tong and Koller, 2001]) or by formalising uncertainty using well studied quantities, such as Shannons entropy and the KL-divergence [Cover et al., 1991]. Although the latter approach was proposed several decades ago [Lindley, 1956, Bernardo, 1979], it is not always straightforward to apply the criteria to complicated models such as nonparametric processes with infinite parameter spaces. As a result many algorithms exist which compute approximate posterior entropies, perform sampling, or work with related quantities in non-probabilistic models.

We return to this problem, presenting the full information criterion and demonstrate how to apply it to Gaussian Processes Classification (GPC), yielding a novel active learning algorithm that makes minimal approximations. GPC is a powerful, non-parametric kernel-based model, and poses an interesting problem for information-theoretic active learning because the 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 apply it to GPC, and show how to extended our method to preference learning. In Section 4 we review other approaches and how they compare to our algorithm. We take particular care to contrast our approach to the Informative Vector Machine, that addresses data point selection for GPs directly. We present results on a wide variety of datasets in Section 5 and conclude 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 $y \in \mathcal{Y}$ on an input variable $\boldsymbol{x} \in \mathcal{X}$. The key idea in active learning is that the learner chooses the input queries $\boldsymbol{x}_i \in \mathcal{X}$ and observes the system's response y_i, rather than passively receiving $(\boldsymbol{x}_i y_i)$ pairs.

Within a Bayesian framework we assume existence of some latent parameters, $\boldsymbol{\theta}$, that control the dependence between inputs and outputs, $p(y|\boldsymbol{x}, \boldsymbol{\theta})$. Having observed data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, a posterior distribution over the parameters is inferred, $p(\boldsymbol{\theta}|\mathcal{D})$. The central goal of information theoretic active learning is to reduce the number possible hypotheses maximally fast, i.e. to minimize the uncertainty about the parameters using Shannon's entropy [Cover et al., 1991]. Data points $\mathcal{D}'$ are selected that satisfy $\arg\min_{\mathcal{D}'} \mathrm{H}[\boldsymbol{\theta}|\mathcal{D}'] = -\int p(\boldsymbol{\theta}|\mathcal{D}') \log p(\boldsymbol{\theta}|\mathcal{D}') \mathrm{d}\boldsymbol{\theta}$. Solving this problem in general is NP-hard; however, as is common in sequential decision making tasks a myopic (greedy) approximation is made [Heckerman et al., 1995]. It has been shown that the myopic policy can perform near-optimally [Golovin and Krause, 2010, Dasgupta, 2005]. Therefore, the objective is to seek the data point $\boldsymbol{x}$ that maximises the decrease in expected posterior entropy:

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

Note that expectation over the unseen output y is required. Many works e.g. [MacKay, 1992, Krishnapuram et al., , Lawrence et al., 2003] propose using this objective directly. However, parameter posteriors are often high dimensional and computing their entropies is usually intractable. Furthermore, for nonparametric processes the parameter space is infinite dimensional so Eqn. (1) becomes poorly defined. To avoid gridding parameter space (exponentially hard with dimensionality), or sampling (from which it is notoriously hard to estimate entropies without introducing bias [Panzeri and Petersen, 2007]), these papers make Gaussian or low dimensional approximations and calculate the entropy of the approximate posterior. A second computational difficulty arises; if $N_{\boldsymbol{x}}$ data points are under consideration, and N_y responses may be seen, then $\mathcal{O}(N_{\boldsymbol{x}} N_y)$, potentially expensive, posterior updates are required to calculate Eqn. (1).

An important insight arises if we note that the objective in Eqn. (1) is equivalent to the conditional mutual information between the unknown output and the parameters, $\mathrm{I}[\boldsymbol{\theta}, y|\boldsymbol{x}, \mathcal{D}]$. Using this insight it is simple to show that the objective can be rearranged to compute entropies in y space:

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

Eqn. (2) overcomes the challenges we described for Eqn. (1). Entropies are now calculated in, usually low dimensional, output space. For binary classification, these are just entropies of Bernoulli variables. Also $\boldsymbol{\theta}$ is now conditioned only on $\mathcal{D}$, so only $\mathcal{O}(1)$ posterior updates are required. Eqn. (2) also provides us with an interesting intuition about the objective; we seek the $\boldsymbol{x}$ for which the model is marginally most uncertain about y (high $\mathrm{H}[y|\boldsymbol{x}, \mathcal{D}]$), but for which individual settings of the parameters are confident (low $\mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\mathcal{D})} \left[\mathrm{H}[y|\boldsymbol{x}, \boldsymbol{\theta}]\right]$). This can be interpreted as seeking the $\boldsymbol{x}$ for which the parameters under the posterior disagree about the outcome the most, so we refer to this objective as Bayesian Active Learning by Disagreement (BALD). We present a method to apply Eqn. (2) directly to GPC and preference learning. We no longer need to build our entropy calculation around the type of posterior approximation (as in [MacKay, 1992, Krishnapuram et al., , Lawrence et al., 2003]) but are free to choose from many of the available algorithms. Minimal additional approximations are introduced, and so, to our knowledge our algorithm represents the most exact and fastest way to perform full information-theoretic active learning in non-parametric discriminative models.

3 Gaussian Processes for Classification and Preference Learning

In this section we derive the BALD algorithm for Gaussian Process classification (GPC). GPs are a powerful and popular non-parametric tool 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 calculate fully the relevant information quantities without having to work out entropies of infinite dimensional objects. The probabilistic model underlying GPC is as follows:

$$\begin{aligned} &f \sim \mathrm{GP}(\mu(\cdot), k(\cdot, \cdot)) \\ &y|\boldsymbol{x}, f \sim \mathrm{Bernoulli}(\Phi(f(\boldsymbol{x}))) \end{aligned}$$

The latent parameter, now called f is a function $\mathcal{X} \to \mathbb{R}$, and is assigned a Gaussian process prior with mean $\mu(\cdot)$ and covariance function or kernel $k(\cdot, \cdot)$. 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 Gaussian CDF. For further details on GPs see [Rasmussen and Williams, 2005].

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 [Rasmussen and Williams, 2005]. 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 one. In our derivation we will use $\overset{1}{\approx}$ to indicate where such an approximation is exploited.

The informativeness of a query $\boldsymbol{x}$ is computed 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:

$$\begin{aligned} \mathrm{H}[y|\boldsymbol{x}, f] &= \mathrm{h}\left(\Phi(f(\boldsymbol{x})\right) \\ \mathrm{h}(p) &= -p \log p - (1-p) \log(1-p) \end{aligned}$$

Expectations over the posterior need to be computed. 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 Eqn. (2) we have to compute two entropy quantities. The first term in Eqn. (2), $\mathrm{H}[y|\boldsymbol{x},\mathcal{D}]$ can be handled analytically for the probit case:

$$\begin{aligned}\mathrm{H}[y|\boldsymbol{x},\mathcal{D}] &\overset{1}{\approx} \mathrm{h}\left(\int \Phi(f_{\boldsymbol{x}})\mathcal{N}(f_{\boldsymbol{x}}|\mu_{\boldsymbol{x},\mathcal{D}},\sigma^2_{\boldsymbol{x},\mathcal{D}})df_{\boldsymbol{x}}\right)\\ &= \mathrm{h}\left(\Phi\left(\frac{\mu_{\boldsymbol{x},\mathcal{D}}}{\sqrt{\sigma^2_{\boldsymbol{x},\mathcal{D}}+1}}\right)\right)\end{aligned} \tag{3}$$

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

$$\begin{aligned}&\mathbb{E}_{f\sim p(f|\mathcal{D})}\left[\mathrm{H}[y|\boldsymbol{x},f]\right]\\ &\quad\overset{1}{\approx} \int \mathrm{h}(\Phi(f_{\boldsymbol{x}}))\mathcal{N}(f_{\boldsymbol{x}}|\mu_{\boldsymbol{x},\mathcal{D}},\sigma^2_{\boldsymbol{x},\mathcal{D}})df_{\boldsymbol{x}} \qquad (4)\\ &\quad\overset{2}{\approx} \int \exp\left(-\frac{f^2_{\boldsymbol{x}}}{\pi\ln 2}\right)\mathcal{N}(f_{\boldsymbol{x}}|\mu_{\boldsymbol{x},\mathcal{D}},\sigma^2_{\boldsymbol{x},\mathcal{D}})df_{\boldsymbol{x}}\\ &\quad= \frac{C}{\sqrt{\sigma^2_{\boldsymbol{x},\mathcal{D}}+C^2}}\exp\left(-\frac{\mu^2_{\boldsymbol{x},\mathcal{D}}}{2\left(\sigma^2_{\boldsymbol{x},\mathcal{D}}+C^2\right)}\right)\end{aligned}$$

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. (4) is intractable. By performing a Taylor expansion on $\ln\mathrm{h}(\Phi(f_{\boldsymbol{x}}))$ (see supplementary material) we can see that it can be approximated up to $\mathcal{O}(f^4_{\boldsymbol{x}})$ 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 - - -), yielding only a 0.27% error in the integral in Eqn. (4). 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 any standard approximate inference algorithm for GPCs (such as EP) 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:

$$\mathrm{h}\left(\Phi\left(\frac{\mu_{\boldsymbol{x},\mathcal{D}}}{\sqrt{\sigma^2_{\boldsymbol{x},\mathcal{D}}+1}}\right)\right)-\frac{C\exp\left(-\frac{\mu^2_{\boldsymbol{x},\mathcal{D}}}{2\left(\sigma^2_{\boldsymbol{x},\mathcal{D}}+C^2\right)}\right)}{\sqrt{\sigma^2_{\boldsymbol{x},\mathcal{D}}+C^2}} \tag{5}$$

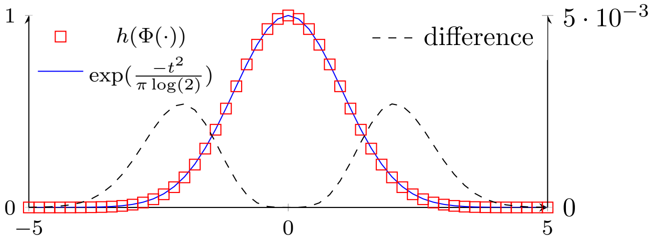

Figure 1: Analytic approximation ($\overset{1}{\approx}$) to the binary entropy of the error function (□) by a squared exponential (——). The absolute error (- - -) remains under $3 \cdot 10^{-3}$.

For most practically relevant kernels, the objective (5) is a smooth and differentiable function of $\boldsymbol{x}$, so gradient-based optimisation procedures can be used to find the maximally informative query.

3.1 Extension: Learning Hyperparameters

In many applications the parameter set $\boldsymbol{\theta}$ naturally divides into parameters of interest, $\boldsymbol{\theta}^+$, and nuisance parameters $\boldsymbol{\theta}^-$, i.e. $\boldsymbol{\theta} = \{\boldsymbol{\theta}^+, \boldsymbol{\theta}^-\}$. In such settings, the active learning may want to query points that are maximally informative about $\boldsymbol{\theta}^+$, while not caring about $\boldsymbol{\theta}^-$. By integrating Eqn. (1) over the nuisance parameters, $\boldsymbol{\theta}^-$, BALD's objective is re-derived as:

$$\begin{aligned} &\mathrm{H}\left[\mathbb{E}_{p(\boldsymbol{\theta}^+,\boldsymbol{\theta}^-|\mathcal{D})}\left[y|\boldsymbol{x},\boldsymbol{\theta}^+,\boldsymbol{\theta}^-\right]\right] \\ &\quad - \mathbb{E}_{p(\boldsymbol{\theta}^+|\mathcal{D})}\left[\mathrm{H}\left[\mathbb{E}_{p(\boldsymbol{\theta}^-|\boldsymbol{\theta}^+,\mathcal{D})}[y|\boldsymbol{x},\boldsymbol{\theta}^+,\boldsymbol{\theta}^-]\right]\right] \end{aligned} \tag{6}$$

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 $\boldsymbol{\theta}^-$ and use Eq. 6, or to include them in $\boldsymbol{\theta}^+$ and perform active learning over them as well. In certain cases, such as automatic relevance determination [Rasmussen and Williams, 2005], it may even make sense to treat hyperparameters as variables of primary interest, and the function f itself as nuisance parameter $\boldsymbol{\theta}^-$.

3.2 Preference Learning

Our active learning framework for GPC can be extended to the important problem of preference learning [Fürnkranz and Hüllermeier, 2003, Chu and Ghahramani, 2005]. In preference learning the 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 $(\boldsymbol{u}, \boldsymbol{v})$. We can view this as a special case of building a classifier on pairs of inputs h : $\mathcal{X}^2 \mapsto \{0, 1\}$. [Chu and Ghahramani, 2005] propose a Bayesian approach, using a latent preference function f, over which a GP prior is defined. The model predicts preference, $\boldsymbol{u}_i \succ \boldsymbol{v}_i$ whenever $f(\boldsymbol{u}_i) + \epsilon_{u_i} > f(\boldsymbol{v}_i) + \epsilon_{v_i}$, where $\epsilon_{u_i}, \epsilon_{v_i}$ denote additive Gaussian noise. Under this model, the likelihood of f becomes:

$$\begin{aligned}\mathbb{P}[y = 1|(\boldsymbol{u}_i, \boldsymbol{v}_i), f] &= \mathbb{P}[\boldsymbol{u}_i \succ \boldsymbol{v}_i|f] \\ &= \Phi\left(\frac{f(\boldsymbol{u}_i) - f(\boldsymbol{v}_i)}{\sqrt{2}\sigma_{noise}}\right) \qquad (7)\end{aligned}$$

By rescaling the latent function f, it can be assumed w.l.o.g. that $\sqrt{2}\sigma_{noise} = 1$. 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: $y|\boldsymbol{u}, \boldsymbol{v}, f \sim \text{Bernoulli}(\Phi(g(\boldsymbol{u}, \boldsymbol{v})))$. 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) as: $k_{\text{pref}}((\boldsymbol{u}_i, \boldsymbol{v}_i), (\boldsymbol{u}_j, \boldsymbol{v}_j)) = k(\boldsymbol{u}_i, \boldsymbol{u}_j) + k(\boldsymbol{v}_i, \boldsymbol{v}_j) - k(\boldsymbol{u}_i, \boldsymbol{v}_j) - k(\boldsymbol{v}_i, \boldsymbol{u}_j)$.

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 [Chu and Ghahramani, 2005], 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 3 for GPC can readily be applied to pairwise preference learning also.

4 Related Methodologies

There are a number of closely related algorithms for active classification which we now review.

The Informative Vector Machine (IVM): Perhaps the most closely related approach is the IVM [Lawrence et al., 2003]. This popular,and successful approach to active learning was designed specifically for GPs; it uses an information theoretic approach and so appears very similar to BALD. The IVM algorithm was designed for subsampling a dataset for training a GP, so it is privy to the y values before including a measurement; it cannot therefore work explicitly in output space i.e. with Eqn. (2). The IVM uses Eqn. (1), but parameter entropies are calculated approximately in the marginal subspace corresponding to the observed data points. The entropy decrease after inclusion of a new data point can then be calculated efficiently using the GP covariance matrix.

Although the IVM and BALD are motivated by the same objective, they work fundamentally differently when approximate inference is carried out. At any time both methods have an approximate posterior $q_t(\boldsymbol{\theta}|\mathcal{D})$, this can be updated with the likelihood of a new data point $p(y_{t+1}|f, \boldsymbol{x}_{t+1})$, yielding $\hat{p}_{t+1}(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{x}_{t+1}, y_{t+1}) = \frac{1}{Z}q_t(\boldsymbol{\theta}|\mathcal{D})p(y_{t+1}|f, \boldsymbol{x}_{t+1})$. If the posterior at $t+1$ is approximated directly one gets $q_{t+1}(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{x}_{t+1}, y_{t+1})$. BALD calculates the entropy difference between q_t and $\hat{p}_{t+1}$, without having to compute q_{t+1} for each candidate $\boldsymbol{x}$. In contrast, the IVM calculates the entropy change between q_t and q_{t+1}. The IVM's approach cannot calculate the entropy of the full infinite dimensional posterior, and requires $\mathcal{O}(N_{\boldsymbol{x}}N_y)$ posterior updates. To do these updates efficiently, approximate inference is performed using Assumed Density Filtering (ADF). Using ADF means that q_{t+1} is a direct approximation to $\hat{p}_{t+1}$, indicating that the IVM makes a further approximation to BALD. Since BALD only requires $\mathcal{O}(1)$ posterior updates it can afford to use more accurate, iterative procedures, such as EP.

Information Theoretic approaches: Maximum Entropy Sampling (MES) [Sebastiani and Wynn, 2000] explicitly works in dataspace (Eqn. (2)). MES was proposed for regression models with input-independent observation noise. Although Eqn. (2) is used, the second term is constant because of input indepedent noise and is ignored. One cannot, however, use MES for heteroscedastic regression or classification; it fails to differentiate between model uncertainty and observation uncertainty (about which our model may be confident). Some toy demonstrations show this '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. [Huang et al., 2010]. This is because MES is inappropriate in this domain; BALD distinguishes between observation and model uncertainty and eliminates these problems as we will show.

Mutual-information based objective functions are presented in [Ertin et al., , Fuhrmann, 2003]. They maximise the mutual information between the variable being measured and the variable of interest. Fuhrmann [Fuhrmann, 2003] applies this to linear Gaussian models and acoustic arrays, Ertin *et al.* [Ertin et al.,] to a communications channel. Although related, these objectives do not work with the model parameters and are not applied to classification. [Guestrin et al., 2005, Krause et al., 2006] also use mutual information. They specify interest points in advance and maximise the expected mutual information between the predictive distributions at these points and at the observed locations. Although this is a objective is promising for regression, it is not tractable for models with input-dependent observation noise, such as classification or preference learning.

Decision theoretic: We briefly mention decision theoretic approaches to active learning. Two closely related algorithms, [Kapoor et al., 2007, Zhu et al., 2003], seek to minimize the expected cost i.e. loss weighted misclassification probability on all seen and future data. These methods observe the locations of the test points and their objective functions become monotonic in the predictive entropies at the test points. [Kapoor et al., 2007] also includes an empirical error term

	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 2: Percentage approximation error (±1 s.d.) for different methods of approximate inference (*columns*) and approximation methods for evaluating Eqn. (4) (*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 [Kuss and Rasmussen, 2005]. For our experiments we use EP.

that can yield pathological behaviour (we investigate this experimentally). These approaches are computationally expensive, requiring $\mathcal{O}(N_{\boldsymbol{x}}N_y)$ posterior updates. Also, they must know the locations of the test data (and thus are transductive approaches); designing an inductive, decision-theoretic algorithm is an open, hard problem as it would require expensive integration over possible test data distributions.

Non-probabilistic Some non-probabilistic methods have close analogues to information theoretic active learning. Perhaps the most ubiquitous is active learning for SVMs [Tong and Koller, 2001, Seung et al., 1992], where the volume of Version Space (VS) is used as a proxy for the posterior entropy. If a uniform (improper) prior is used with a deterministic classification likelihood, the log volume of VS and Bayesian posterior entropy are in fact equivalent. Just as Bayesian posteriors become intractable after observing many data points, VS can become complicated. [Tong and Koller, 2001] proposes methods for approximating VS with a simple shapes, such as hyperspheres (their simplest approximation reduces to margin sampling). This closely resembles approximating a Bayesian posterior using a Gaussian distribution via the Laplace or EP approximations. [Seung et al., 1992] sidesteps the problem by working with predictions. The algorithm, Query by Committee (QBC), samples parameters from VS (committee members), they vote on the outcome of each possible $\boldsymbol{x}$. 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.

5 Experiments

Quantifying Approximation Losses: To obtain (5) 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 sampling, enabling us to compute

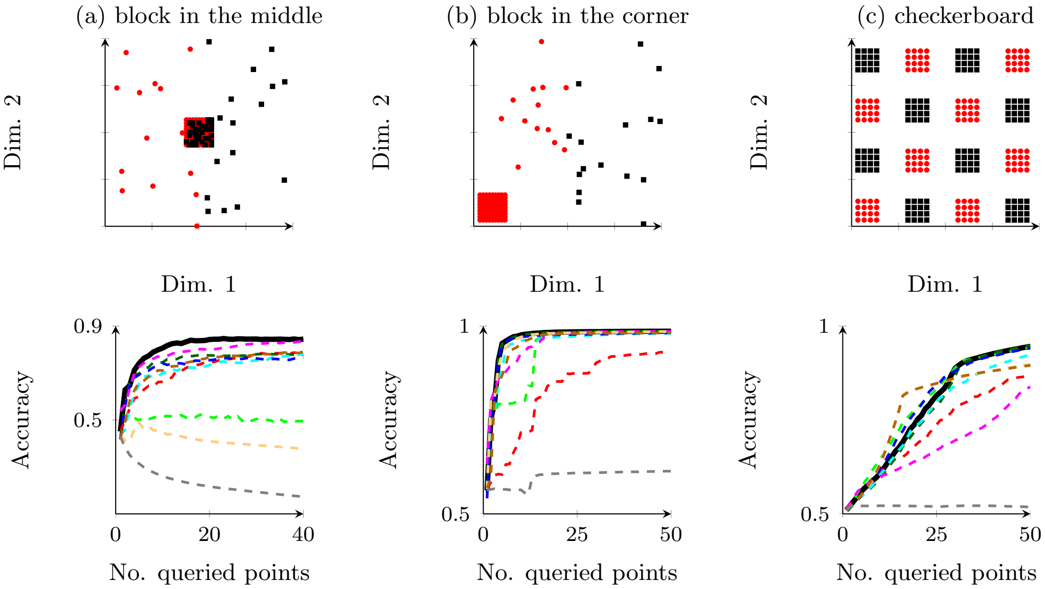

Figure 3: *Top:* Evaluation on artificial datasets. Exemplars of the two classes are shown with black squares (▪) and red circles (•). *Bottom:* Results of active learning with nine methods: random query (- - -), BALD(▬), MES (- - -), QBC with the vote criterion with 2 (- - -) and 100 (- - -) committee members, active SVM (- - -), IVM (- - -), decision theoretic: [Kapoor et al., 2007] (- - -), [Zhu et al., 2003] (- - -) and empirical error (- - -).

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:

$$\frac{\max_{\boldsymbol{x}\in\mathcal{P}} I(\boldsymbol{x}) - I(\arg\max_{\boldsymbol{x}\in\mathcal{P}} \hat{I}(\boldsymbol{x}))}{\max_{\boldsymbol{x}\in\mathcal{P}} I(\boldsymbol{x})} \cdot 100\% \tag{8}$$

where I is the objective computed using Monte Carlo, $\hat{I}$ is the approximate objective. The *cancer* UCI dataset was used, results and discussion are in Fig. 2.

Pool based active learning: We test BALDfor GPC and preference learning in the pool-based setting i.e. selecting $\boldsymbol{x}$ values from a fixed set of data-points. Although BALD can generalise to selecting continuous $\boldsymbol{x}$, this enables us to compare to algorithms that cannot. We compare to eight other algorithms: random sampling, MES, QBC (with 2 and 100 committee members), SVM with version space approximation [Tong and Koller, 2001], decision theoretic approaches in [Kapoor et al., 2007, Zhu et al., 2003] and directly minimizing

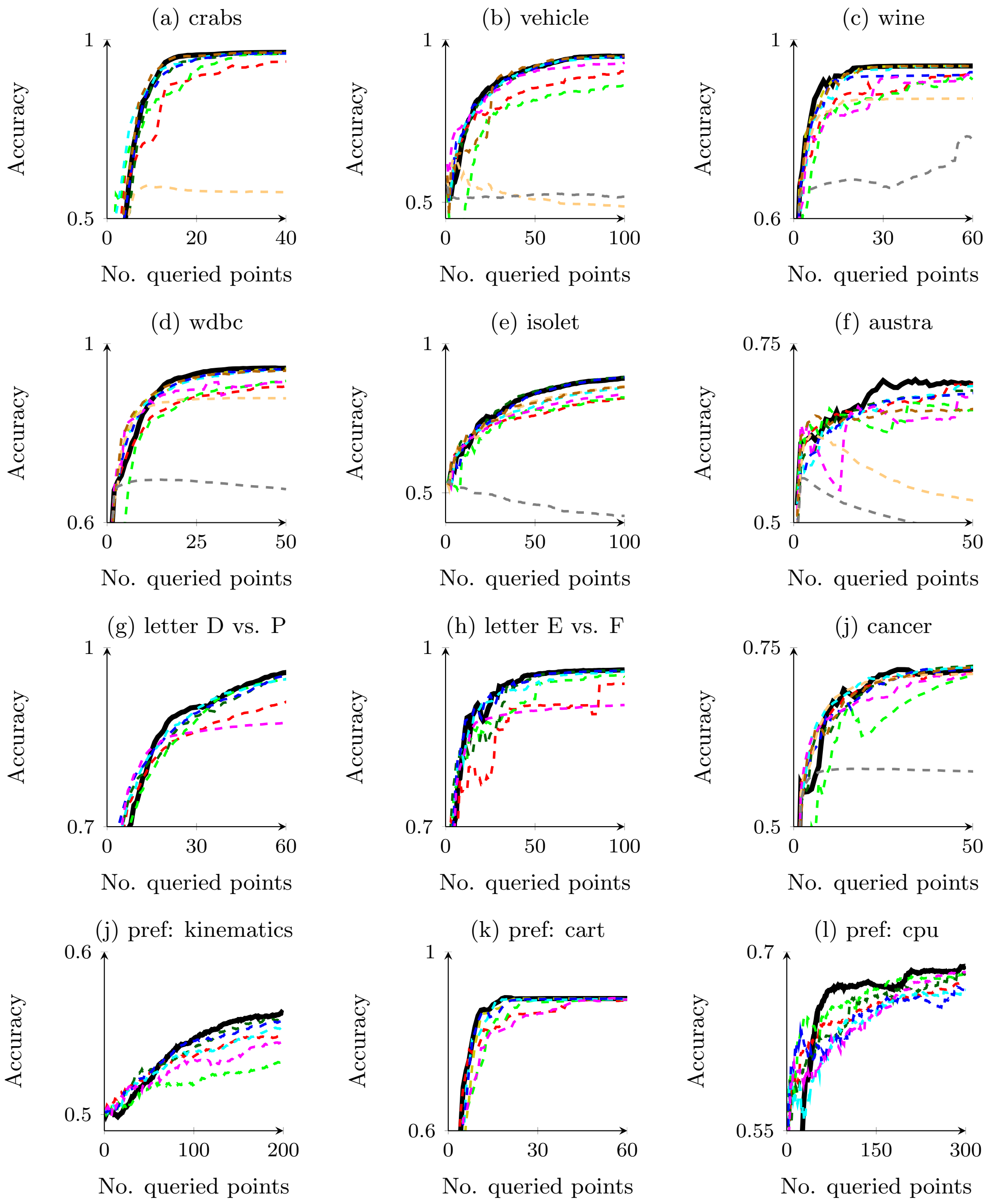

Figure 4: Test set classification accuracy on classification and preference learning datasets. Methods used are BALD(▬▬), random query (- - -), MES (- - -), QBC with 2 (QBC$_2$, - - -) and 100 (QBC$_{100}$, - - -) committee members, active SVM (- - -), IVM (- - -), decision theoretic [Kapoor et al., 2007] (- - -), decision theoretic [Zhu et al., 2003] (- - -) and empicial error (- - -). The decision theoretic methods took a long time to run, so were not completed for all datasets. Plots (a-i) are GPC datasets, (j-l) are preference learning.

.

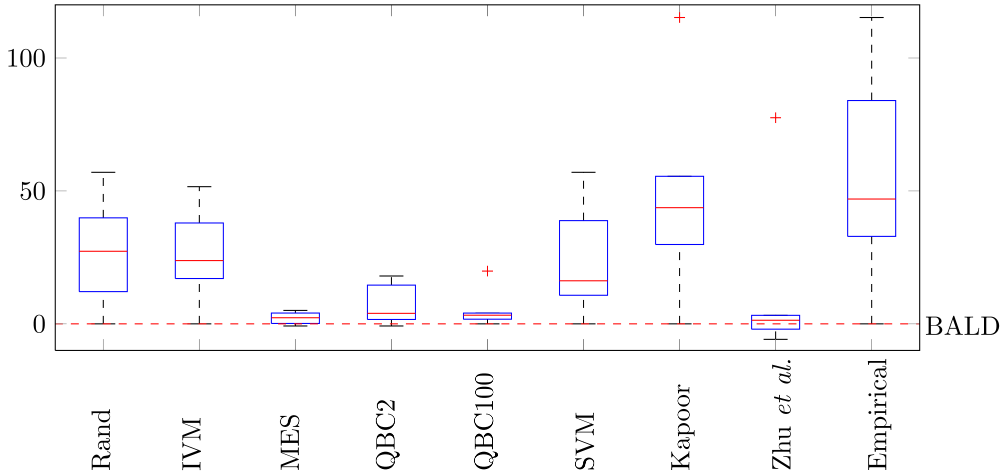

Figure 5: Summary of results for all classification experiments. y-axis denotes the number of additional data points, relative to BALD, required to achieve at least 97.5% of the predictive performance of the entire pool. The 'box' denotes 25th to 75th percentile, the red line denotes the median over datasets, and the 'whiskers' depict the range. The crosses denote outliers ($> 2.7\sigma$ from the mean). Positive values mean that the algorithm required more data points than BALD to achieve the same performance.

expected empirical error (the last is not a widely used method, but is included for analysis of [Kapoor et al., 2007]).

We consider three artificial, but challenging, datasets. The first of which, *block in the middle*, has a block of noisy points on the decision boundary, the second *block in the corner*, has a block of uninformative points far from the decision boundary: a strong active learning algorithm should avoid these uninformative regions. The third is similar to the *checkerboard* dataset in [Zhu et al., 2003], and is designed to test the algorithm's capabilities to find multiple disjoint islands of points from one class. The three datasets and results using each algorithm are depicted in Fig. 3.

Results are also presented on eight UCI classification datasets *australia, crabs, vehicle, isolet, cancer, wine, wdbc* and *letter*. *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 [Chu and Ghahramani, 2005]. Results are plotted in Fig. 4, and Fig. 5 depicts an aggregation of the results.

Discussion: Figs. 3 and 4 show that by using BALDwe make significant gains over naive random sampling in both the classification and preference learning domains. Relative to other active learning algorithms BALDis consistently the

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

best, or amongst the best performing algorithms on all datasets. On any individual dataset BALD's performance is often matched because we compare to many methods, and the more approximate algorithms can have good performance under different conditions. Fig. 5 reveals that BALD has the best overall performance; on average, all other methods require more data points to achieve the same classification accuracy. Zhu *et al.*'s decision theoretic approach is closest, the median increase in the number of data points required is 1.4 and zero (i.e. equivalent to BALD) is within the inter-quartile range. This algorithm, however, requires much more computational time and has access to the full set of test inputs, which BALD does not have. MES and QBC appear close in performance to BALD, but the zero line falls outside both of their inter-quartile ranges.

As expected, MES performs poorly on the noisy dataset (Fig. 3(a)) because it discards knowledge of observation noise. When there is zero observation noise it is equivalent to BALD e.g. Fig. 3(c). On many of the real-world datasets MES performs as well as BALD e.g. Fig. 4(b, e), indicating that these datasets are mostly noise-free.

The IVM performs well on Fig. 3(c), but pathologically on 3(a); this is due to the fact that it biases selection towards points from only one class in the noisy cluster, reducing the posterior entropy rapidly but artificially. However, it also performs significantly worse than BALD on noise-free (indicated by MES's strong performance) datasets e.g. Fig. 4(b). This implies that the IVM's posterior approximation or the ADF update are detrimental to the algorithm's performance.

QBC often yields only a small decrement in performance, the sampling approximation is often not too detrimental. However, it performs poorly on the noisy artificial dataset (Fig. 3(a)) because the vote criterion is not maintaining a notion of inherent uncertainty, like MES. The SVM-based approach exhibits variable performance (it does well on Fig. 4(d), but very poorly on 4(f)). The performance is greatly effected by the approximation used, for consistency we present here one that yielded the most consistent good performance.

Decision theoretic approaches sometimes perform well, on 3(c) they choose the first 16 points from the centre of each cluster as they are influenced by the surrounding unlabelled points. BALDdoes not observe the unlabelled points so may not pick points from the centres. Fig. 5 reveals that BALD is performing as well as the method in [Zhu et al., 2003], and outperforms the approach in [Kapoor et al., 2007], despite not having access to the locations of the test points and having a significantly lower computational cost. The objective in [Kapoor et al., 2007] can fail, this is because one term in their objective function is the empirical error. The weight given to this term is determined by the relative sizes of the training and test set (and the associated losses). Directly minimizing empirical error usually performs very pathologically, picking only 'safe' points. When the method in [Kapoor et al., 2007] assigns too much weight to this term, it can fail also.

Finally we note that BALD may occasionally perform poorly on the first few data points (e.g. Fig. 4(l)). This is may be because the hyperparameters are fixed throughout the experiments to provide a fair comparison to algorithms incapable of incorporating hyperparameter learning. This may mean that given little data the GP model overfits, leading to BALD selecting abnormal query locations. Maintaining a distribution over hyperparameters can be done using MCMC, although this significantly increases computational time. Designing a general method to do this efficiently is a subject of further work. In practice, a simple heuristic such as picking the first few points randomly, and optimising hyperparameters will usually suffice.

6 Conclusions

We have demonstrated a method that applies the full information theoretic active learning criterion to GP classification that makes, as far as the authors are aware, the smallest number of approximations to date, and has as good computational complexity. We extend the GPC model to develop a new preference learning kernel, which enables us to apply our active learning algorithm directly to this domain also. The method can handle naturally active learning of kernel hyperparameters, 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, the Laplace approximation, ADF or even sparse online learning, and thereby make the trade off between computational complexity and accuracy. Our experimental performance compares favourably to many other active learning methods for classification, and even decision theoretic methods that have access to the test data and require much greater computational time.

References

[Bernardo, 1979] Bernardo, J. (1979). Expected information as expected utility. *The Annals of Statistics*, 7(3):686–690.

[Chu and Ghahramani, 2005] Chu, W. and Ghahramani, Z. (2005). Preference learning with Gaussian processes. In *ICML*, pages 137–144. ACM.

[Cover et al., 1991] Cover, T., Thomas, J., and Wiley, J. (1991). *Elements of information theory*, volume 6. Wiley Online Library.

[Dasgupta, 2005] Dasgupta, S. (2005). Analysis of a greedy active learning strategy. In *NIPS*.

[Ertin et al.,] Ertin, E., Fisher, J., and Potter, L. Maximum mutual information principle for dynamic sensor query problems. In *Information Processing in Sensor Networks*, Lecture Notes in Computer Science.

[Fuhrmann, 2003] Fuhrmann, D. (2003). *Active Testing Surveillance Systems, or, Playing Twenty Questions with a Radar.* Defense Technical Information Center.

[Fürnkranz and Hüllermeier, 2003] Fürnkranz, J. and Hüllermeier, E. (2003). Pairwise preference learning and ranking. *Machine Learning: ECML 2003*, pages 145–156.

[Golovin and Krause, 2010] Golovin, D. and Krause, A. (2010). Adaptive submodularity: A new approach to active learning and stochastic optimization. In *COLT*.

[Guestrin et al., 2005] Guestrin, C., Krause, A., and Singh, A. P. (2005). Near-optimal sensor placements in Gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 265–272, New York, NY, USA. ACM.

[Heckerman et al., 1995] Heckerman, D., Breese, J., and Rommelse, K. (1995). Troubleshooting under uncertainty. *Communications of the ACM*, 38(3):27–41.

[Huang et al., 2010] Huang, S., Jin, R., and Zhou, Z. (2010). Active learning by querying informative and representative examples. *Advances in neural information processing systems*, 23:892–900.

[Kapoor et al., 2007] Kapoor, A., Horvitz, E., and Basu, S. (2007). Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *IJCAI*.

[Krause et al., 2006] Krause, A., Guestrin, C., Gupta, A., and Kleinberg, J. (2006). Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Proceedings of the 5th international conference on Information processing in sensor networks*, pages 2–10. ACM.

[Krishnapuram et al.,] Krishnapuram, B., Williams, D., Xue, Y., Hartemink, A., Carin, L., and Figueiredo, M. On semi-supervised classification. *NIPS*.

[Kuss and Rasmussen, 2005] Kuss, M. and Rasmussen, C. E. (2005). Assesing approximations for gaussian process classification. In *NIPS*. MIT Press.

[Lawrence et al., 2003] Lawrence, N., Seeger, M., and Herbrich, R. (2003). Fast sparse Gaussian Process methods: The informative vector machine. *Advances in neural information processing systems*, pages 625–632.

[Lindley, 1956] Lindley, D. (1956). On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005.

[MacKay, 1992] MacKay, D. (1992). Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604.

[Panzeri and Petersen, 2007] Panzeri, S., S. R. M. M. and Petersen, R. (2007). Correcting for the sampling bias problem in spike train information measures. *Journal of neurophysiology*, 98(3):1064.

[Rasmussen and Williams, 2005] Rasmussen, C. and Williams, C. (2005). *Gaussian Processes for Machine Learning*. The MIT Press.

[Roy and McCallum, 2001] Roy, N. and McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In *ICML*, pages 441–448.

[Sebastiani and Wynn, 2000] Sebastiani, P. and Wynn, H. (2000). Maximum entropy sampling and optimal Bayesian experimental design. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(1):145–157.

[Seung et al., 1992] Seung, H., Opper, M., and Sompolinsky, H. (1992). Query by committee. In *COLT*, pages 287–294. ACM.

[Tong and Koller, 2001] Tong, S. and Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66.

[Zhu et al., 2003] Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining.*

APPENDIX – SUPPLEMENTARY MATERIAL

Taylor Expansion for Approximation $\overset{2}{\approx}$

We perform a Taylor expansion on $\ln \mathrm{H}[\Phi(x)]$ as follows:

$$\begin{aligned}
f(x) &= f(0) + \frac{f'(0)x}{1!} + \frac{f''(0)x^2}{2!} + \dots \\
f(x) &= \ln \mathrm{H}[\Phi(x)] \\
f'(x) &= -\frac{1}{\ln 2}\frac{\Phi'(x)}{\mathrm{H}[\Phi(x)]}\left[\ln\Phi(x) - \ln(1-\Phi(x))\right] \\
f''(x) &= \frac{1}{\ln 2}\frac{\Phi'(x)^2}{\mathrm{H}[\Phi(x)]^2}\left[\ln\Phi(x) - \ln(1-\Phi(x))\right] \\
&\quad - \frac{1}{\ln 2}\frac{\Phi''(x)}{\mathrm{H}[\Phi(x)]}\left[\ln\Phi(x) - \ln(1-\Phi(x))\right] \\
&\quad - \frac{1}{\ln 2}\frac{\Phi'(x)^2}{\mathrm{H}[\Phi(x)]}\left[\frac{1}{\Phi(x)} + \frac{1}{(1-\Phi(x)})\right] \\
\therefore \ln \mathrm{H}[\Phi(x)] &= 1 - \frac{1}{\pi \ln 2}x^2 + \mathcal{O}(x^4)
\end{aligned}$$

Because the function is even, we can inspect that the x^3 term will be zero. Therefore, exponentiating, we make the approximation up to $\mathcal{O}(x^4)$:

$$\mathrm{H}[\Phi(x)] \overset{2}{\approx} \exp\left(-\frac{x^2}{\pi \ln 2}\right)$$

Preference Kernel

The mean μ_{pref}, and covariance function k_{pref} of the GP over g can be computed from the mean and covariance of $f \sim \mathrm{GP}(\mu, k)$ as follows:

$$\begin{aligned}
k_{\text{pref}}([\boldsymbol{u}_i, \boldsymbol{v}_i], [\boldsymbol{u}_j, \boldsymbol{v}_j]) &= Cov[g(\boldsymbol{u}_i, \boldsymbol{v}_i), g(\boldsymbol{u}_j, \boldsymbol{v}_j)] \\
&= Cov\left[\left(f(\boldsymbol{u}_i) - f(\boldsymbol{v}_i)\right), \left(f(\boldsymbol{u}_i) - f(\boldsymbol{v}_i)\right)\right] \\
&= \mathbb{E}\left[\left(f(\boldsymbol{u}_i) - f(\boldsymbol{v}_i)\right) \cdot \left(f(\boldsymbol{u}_i) - f(\boldsymbol{v}_i)\right)\right] \\
&\qquad - \left(\mu(\boldsymbol{u}_i) - \mu(\boldsymbol{v}_i)\right)\left(\mu(\boldsymbol{v}_j) - \mu(\boldsymbol{u}_i)\right) \\
&= k(\boldsymbol{u}_i, \boldsymbol{u}_j) + k(\boldsymbol{v}_i, \boldsymbol{v}_j) \\
&\qquad - k(\boldsymbol{u}_i, \boldsymbol{v}_j) - k(\boldsymbol{v}_i, \boldsymbol{u}_j) && (9) \\
\mu_{\text{pref}}([\boldsymbol{u}, \boldsymbol{v}]) &= \mathbb{E}\left[g([\boldsymbol{u}, \boldsymbol{v}])\right] = \mathbb{E}\left[f(\boldsymbol{u}) - f(\boldsymbol{v})\right] \\
&= \mu(\boldsymbol{u}) - \mu(\boldsymbol{v}) && (10)
\end{aligned}$$