Chapter 2

Bayesian Active Learning

This chapter introduces the Bayesian framework used to design the active learning algorithms in this thesis. This framework takes an information-theoretic approach to active learning. First, in Section 2.1 we introduce active learning and then describe the Bayesian information theoretic approach in Section 2.2. We discuss possible computational difficulties, and present our framework which can circumnavigate these, called Bayesian Active Learning by Disagreement (BALD), in Section 2.3. We discuss properties of the framework, such as computational complexity and submodularity. The chapter finishes with a review of related literature in active learning and experimental design that has developed in statistics, experimental science, social science, computer science and machine learning.

2.1 Introduction to Active Learning

We first introduce active learning, discuss when it is applicable, and present the different settings for active querying.

2.1.1 When Active Learning may be Applied

Data, particularly labelled data, can be expensive to obtain. Therefore, it is desirable to collect only the most useful points. To address this, active learning algorithms choose their training data. These are 'active' in the sense that they can adjust which points they choose in light of data collected so far.

A concern may arise: when choosing the data to learn from, can one bias the inferences made? Fortunately, this is not the case, provided that we condition our inferences on the actively selected variables. More concretely, consider the setting where we choose

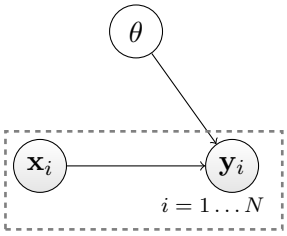

Figure 2.1: Graphical model for a discriminative learner. White nodes indicate latent (unobserved) variables and shaded nodes denote observed variables. Note that the distribution of the input $\mathbf{x}$ is assumed to be independent of the parameters θ.

a query $\mathbf{x} \in \mathcal{X}$, and observe a variable $\mathbf{y} \in \mathcal{Y}$ in response. These quantities shall be referred to as the *input* and *output* variables respectively.[1] Suppose that we also have a model with parameters $\theta \in \Theta$ that describes the dependence of the output on the input, $p(\mathbf{y}|\mathbf{x}, \theta)$. This setting is called *discriminative learning*, see Figure 2.1. Provided that we condition on $\mathbf{x}$, we may perform Bayesian computations without introducing inferential bias into posterior distribution $p(\theta|\mathbf{y}, \mathbf{x})$ or the predictive distribution $p(\mathbf{y}|\mathbf{x})$[2]. More specifically, when computing these quantities, we must compute the likelihood function $p(\mathbf{y}|\mathbf{x}, \theta)$. When computing the likelihood we implicitly integrate over all of the unobserved data. Ignoring this integral does not effect the likelihood provided that we condition on all observed data, regardless of how it was collected, and do not filter it based upon the output variable $\mathbf{y}$.

The alternative to discriminative learning is *generative learning* in which the entire data distribution $p(\mathbf{x}, \mathbf{y})$ is modelled [Ng & Jordan, 2002]. Learning a full generative model actively is not possible because one does not observe an unbiased sample from $p(\mathbf{x})$. However, this terminology may be confusing. For example, probabilistic models of matrix data are normally termed 'generative' as the data is a matrix $\mathbf{Y}$, and matrix models typically generate the entire dataset, they learn $p(\mathbf{Y}|\theta)$. However, active learning with matrices can be reconciled with the framework in Figure 2.1. Here, we select a row and column index and observe that entries' value. The location of each matrix entry corresponds to the input variable $\mathbf{x}$, and we do not model the entries' locations directly, but we condition upon them. However, this variable is usually implicit in the equations.

[1] $\mathcal{X}, \mathcal{Y}$ can be general input and output spaces. Unless stated otherwise, $\mathcal{X}$ will be a real-valued vector space $\mathcal{X} = \mathbb{R}^D$. $\mathcal{Y}$ will usually be uni-dimensional, but may be binary, real-valued, ordinal etc.

[2] In this context we use the term 'bias' loosely to mean that the distributions differ to those computed by integrating all of the unobserved data, not in the formal sense of a biased statistical estimator.

A second distinction that is often made is between *supervised* and *unsupervised* learning. In supervised learning there are clearly defined input and output variables, and the goal is to predict the output corresponding to new, unseen inputs. Supervised learning fits well into the discriminative setting in Figure 2.1. Supervised active learning is the focus of this thesis, although the methods could be extended to unsupervised learning by choosing a partition of the variables and learning the conditional distributions actively.

2.1.2 Query Scenarios

Active learning can be performed in three scenarios: *continuous sampling*, *pool based* and *stream based* learning. In continuous sampling any point in input space may be selected. This is appropriate when the input space can be queried to arbitrary precision. For example, in cognitive science where the queries may be computer generated stimuli with real-valued parameters, or when learning the kinematics of a robot arm by choosing joint angles and measuring hand coordinates [Cohn *et al.*, 1996]. However, in other regimes continuous querying is inappropriate. In a handwriting recognition system, Baum & Lang [1992] found that many generated query images contained no recognizable symbols and hence could not be labelled. In Chapter 4 we perform continuous sampling, adjusting real-valued parameters of a physics experiment to select continuous measurements.

In pool based active learning one has access to a set (pool) of unlabelled data from which to select points for annotation. For example, in a study on the effects of smoking on cancer rates, the pool contains people whose age, smoking habits etc. are known, and the scientist wants to select only the most informative few for expensive clinical trials. This is probably the most common scenario in machine learning, occurring in many applications including text classification [Tong & Koller, 2002], image classification [Tong & Chang, 2001], speech recognition [Tur *et al.*, 2005], cancer diagnosis [Liu, 2004] and recommendation systems [Jin & Si, 2004].

Stream based active learning is closely related to pool based learning, except that the pool is presented sequentially and the algorithm must decide online whether or not to collect the point's label. A famous example is 'triggering' on the CERN particle accelerator; there is insufficient storage capacity to store the vast stream of events, so only potentially interesting measurements are recorded [Aad *et al.*, 2012].

Active learning algorithms must assign a score, or utility, to each location in input space that may be queried. Continuous sampling is the most flexible, but requires

optimization of the utility function over input space. This may involve a hard high dimensional, multi-modal optimization problem. If the utility function is not differentiable, continuous optimization may be infeasible. In pool based active learning the utility is normally evaluated for every point in the pool, but if the pool is large, or the utility is expensive to evaluate, pruning may be required [Roy & McCallum, 2001]. We now present general strategies for active learning, these are usually applicable in any of the three scenarios, but the computational issues above should be considered in practice.

2.2 Information Theoretic Active Learning

Probabilistic active learning broadly divides into two categories, *decision* and *information* theoretic. We take the information-theoretic approach, which we first motivate.

2.2.1 Motivation

In Bayesian methods, it is common to separate learning from decision making. This permits general models and learning algorithms to be used in a variety of tasks. Learning, or inference[1], involves applying Bayes' rule (1.1) to compute the posterior distribution of the parameters of the model $p(\theta|\mathcal{D})$. Decision making involves choosing an action a, which incurs a particular loss that depends on the true parameters of the system $\hat{\theta}$, denoted $\mathcal{L}(\hat{\theta}, a)$. However, in practice the true parameters are unknown. Therefore, the optimal course of action, on average, is to minimize the expected loss given our beliefs about the parameters,

$$\mathcal{R}_{\mathcal{D}}(a) = \mathbb{E}_{p(\theta|\mathcal{D})}\mathcal{L}(\theta, a)\,. \tag{2.1}$$

$\mathcal{R}_{\mathcal{D}}(a)$ is called the Bayes posterior risk. Inference and decision making can be separated because the posterior is not a function of the loss. This can be beneficial because the same posterior can be used to solve different tasks.[2]

Active learning algorithms need to quantify how 'useful' datapoints are. To do this, they are equipped with a utility function $\mathcal{U}(\mathbf{x}) : \mathcal{X} \to \mathbb{R}$. The optimal utility

[1] Sometimes these terms are distinguished. Inference is often used to refer to computing the posterior distribution over variables local to each datapoint. Learning then refers to optimizing or computing the posterior over global model parameters. Unless context dictates otherwise, these terms will be used interchangeably.

[2] Recent work couples inference with the loss function in order to incur lower loss when the posterior can only be approximated [Abbasnejad *et al.*, 2013; Lacoste-Julien *et al.*, 2011].

function, from a loss minimization standpoint, minimizes the expected posterior risk after observing the output $\mathbf{y}$ corresponding to input $\mathbf{x}$,

$$\mathcal{U}(\mathbf{x}) = \mathbb{E}_{p(\mathbf{y}|\mathbf{x},\mathcal{D})} \operatorname*{argmin}_a \mathcal{R}_{\{\mathcal{D},\mathbf{x},\mathbf{y}\}}(a)\,. \tag{2.2}$$

Choosing samples to maximize Equation (2.2) is known as *decision theoretic* active learning [Kapoor *et al.*, 2007; Roy & McCallum, 2001].

Decision theoretic active learning, although theoretically optimal in terms of expected loss minimization, can be disadvantageous for two reasons. First, the utility function is tied to a particular task and loss function. These may not be known ahead of time, or we may desire a learning algorithm that can perform well with a variety of loss functions. Second, computing the expected change in risk can be expensive. This is because the utility function in Equation (2.2) includes both the learning and decision making processes. Thus, to compute Equation (2.2) one must calculate the expected change in beliefs and then compute the new optimal decision after including any new data $\{\mathbf{x}, \mathbf{y}\}$.

An alternative is to focus upon learning alone, and choose data that is most useful for inferring the parameters θ. This approach decouples learning from future decision making which is consistent with many Bayesian methods. To measure the utility of a datapoint, we must quantify its 'informativeness' about the parameters. Information theory is a natural tool for doing this, so this approach is called *information theoretic* active learning.

2.2.2 Information Theory

Before presenting information-theoretic active learning, a brief overview of elementary information theory is provided. For a complete introduction see Cover *et al.* [1994]. Information theory was founded by Claude Shannon who studied data transmission over noisy communication channels [Shannon, 1948]. Shannon derived a theoretical upper bound for the capacity of a channel, which is the maximum rate that a set of symbols $X = \{x_1 \dots x_N\}$ which have distribution $P(X)$ can be transmitted with zero reconstruction error. To do this he defined the *information content* in a symbol x, and the *entropy*, which is the average information content in the ensemble.

$$\text{Information content: } \mathrm{J}(x) = -\log P(x)\,, \tag{2.3}$$

$$\text{Entropy: } \mathrm{H}[P(X)] = -\sum_x P(x) \log P(x)\,. \tag{2.4}$$

The usual shorthand $P(x) \equiv P(X = x)$ will be used to denote the probability that X takes a particular value x when context makes the usage clear. When unambiguous, $\mathrm{H}[X]$ will be used to denote the entropy of the distribution $P(X)$. Entropy is a measure of uncertainty in a distribution that satisfies two intuitive desiderata. First, $\mathrm{H}[X] \geq 0$, and takes value zero only when all of the mass in $P(X)$ is concentrated on a single symbol. In this case the outcome of sampling from $P(X)$ is certain. Furthermore, $\mathrm{H}[X]$ is maximized when $P(X)$ is a uniform distribution. Intuitively, this corresponds to maximal uncertainty. The second desiderata is that entropy is additive for independent random variables. If X and Y are independent, then $\mathrm{H}[P(X, Y)] = \mathrm{H}[P(X)] + \mathrm{H}[P(Y)]$. Intuitively, if we have n units of uncertainty in X and m units of uncertainty in Y, then we have $n+m$ units of uncertainty in their joint distribution. The base of the logarithm in Equations (2.3) and (2.4) changes the quantities by a multiplicative constant, with base two the units of information are called bits. When measured in bits, the entropy may be interpreted as "The average number of binary questions that must be asked to determine the value of a sample from $P(X)$."

Shannon's entropy can be extended to distributions over continuous variables by replacing the sum in Equation (2.4) with an integral. This quantity is known as the differential entropy.[1]

$$\text{Differential entropy: } \mathrm{H}[p(X)] = -\int p(x) \log p(x) dx \,,$$

where X now denotes a continuous random variable and $p(X)$ is a continuous probability density function. Again, this quantity is maximized when $p(X)$ is a uniform distribution over the domain of X, and it is minimized when $p(X)$ is a Dirac delta function $\delta(x - x')$. However, in this case the differential entropy equals $-\infty$, because x can be specified to infinite precision and hence can carry an infinite amount of information.

Two further information theoretic quantities that occur frequently are the *mutual information*, and *Kullback-Leibler* (KL) divergence. The mutual information between

[1] It may concern some that, unlike entropy in Equation (2.4), the differential entropy is not a dimensionless quantity. However, it be thought of as the KL divergence, Equation (2.7), to an improper uniform distribution $u(x)$, $\mathrm{H}[p(x)] = -\int p(x) \log \frac{p(x)}{u(x)} dx$, which is dimensionless.

two random variables X and Y is

$$\text{mutual information: } \mathrm{I}[X,Y] = \mathrm{H}[p(X)] - \mathbb{E}_{p(Y)}\mathrm{H}[p(X|Y)] \tag{2.5}$$
$$= \mathrm{H}[p(Y)] - \mathbb{E}_{p(X)}\mathrm{H}[p(Y|X)]$$
$$= \mathrm{H}[p(X)] + \mathrm{H}[p(Y)] - \mathrm{H}[p(X,Y)]\,, \tag{2.6}$$

where $\mathbb{E}_{p(Y)}\mathrm{H}[p(X|Y)]$ is the *conditional entropy*, and is often denoted $\mathrm{H}[X|Y]$. Mutual information is a non-negative quantity that is symmetric in its arguments. It measures how much information X carries about Y, and vice versa. Shannon showed that the maximum possible capacity of a channel is given by the mutual information between the sent and received signals. The mutual information equals zero if and only if X and Y are statistically independent, that is $p(X,Y) = p(X)p(Y)$.

The KL divergence is a measure of dissimilarity between two probability distributions $p(X)$ and $q(X)$,

$$\text{KL divergence: } \mathrm{KL}[p(X)||q(X)] = \int p(x) \log \frac{p(x)}{q(x)} dx\,. \tag{2.7}$$

This divergence is non-negative, but asymmetric. It is equal to zero if and only if $p(X)$ is identical to $q(X)$. Intuitively, the KL divergence is the number of additional bits needed to transmit symbols with distribution $p(X)$, if our model of the distribution is $q(X)$.

Entropy is a well established characterization of uncertainty in a probability distribution and provides a natural metric for information theoretic active learning.

2.2.3 The Information Gain Utility Function

In information theoretic Bayesian active learning one is agnostic to future decision tasks and loss functions. The goal is to learn as quickly as possible about the model parameters θ. With Shannon's entropy (2.4) to quantify the uncertainty in a probability distribution, the natural objective is to minimize posterior entropy after collecting data. However, if we collect many datapoints in sequence, optimizing this quantity is NP-hard in the horizon length [Ko *et al.*, 1995; Krause & Guestrin, 2005]. Therefore, as is common in sequential decision making, we take a myopic (greedy) approach, selecting the next point as if it were the last. The implications of the myopic approximation are discussed in Section 2.3.6. The myopic expected information gain is given by

$$\mathcal{U}(\mathbf{x}) = \mathrm{H}[p(\theta|\mathcal{D})] - \mathbb{E}_{p(\mathbf{y}|\mathbf{x},\mathcal{D})}\mathrm{H}[p(\theta|\mathcal{D},\mathbf{x},\mathbf{y})]\,.^{1} \tag{2.8}$$

The expectation over $\mathbf{y}$ is taken because the output is unknown before the query has been made. Equation (2.8) was first proposed for the design of Bayesian experiments in Lindley [1956]. However, just as Bayes' rule itself is usually intractable, Equation (2.8) is difficult to compute with most interesting models. Therefore, much recent work has addressed information theoretic active learning; mathematical approximations and algorithmic techniques have been developed to apply Equation (2.8) to complex models.

As an aside: another intuitive information-theoretic objective is to maximize the KL-divergence between the current and next posterior, $\mathrm{KL}[p(\theta|\mathcal{D}, \mathbf{x}, \mathbf{y})||p(\theta|\mathcal{D})]$. However, after taking expectations with respect to $\mathbf{y}$ this utility function is equivalent to entropy decrease in Equation (2.8) [MacKay, 1992b].

2.3 Bayesian Active Learning by Disagreement

We now present an entirely equivalent formulation of Equation (2.8). This reformulation can provide substantial practical advantages that shall reappear throughout this thesis. We then discuss various properties of this approach to active learning including computational issues, extensibility, inductivity and submodularity. Inductivity and submodularity are general properties of information-theoretic active learning. The computational properties and extensions are specific to our method used to compute the utility, Bayesian Active Learning by Disagreement.

2.3.1 Symmetry in the Objective

An important insight arises if we note that Equation (2.8) is equivalent to the mutual information (2.5) between parameters and the unobserved output, conditioned upon the input and the observed data $\mathrm{I}[\theta; \mathbf{y}|\mathcal{D}, \mathbf{x}]$. Because mutual information is symmetric in its arguments, Equation (2.8) can be re-written as follows,

$$\begin{aligned}
\mathcal{U}(\mathbf{x}) &= \mathrm{H}[p(\theta|\mathcal{D})] - \mathbb{E}_{p(\mathbf{y}|\mathbf{x},\mathcal{D})}\mathrm{H}[p(\theta|\mathcal{D}, \mathbf{x}, \mathbf{y})] \\
&= \mathrm{I}[\theta, \mathbf{y}|\mathcal{D}, \mathbf{x}] && (2.9) \\
&= \mathrm{H}[p(\mathbf{y}|\mathbf{x}, \mathcal{D})] - \mathbb{E}_{p(\theta|\mathcal{D})}\mathrm{H}[p(\mathbf{y}|\mathbf{x}, \theta)]\,. && (2.10)
\end{aligned}$$

Equation (2.10) is equivalent to the expected information gain in Equation (2.8), but it provides a different intuition about the utility function. The first term in Equation (2.10) seeks the input $\mathbf{x}$ for which the model has high uncertainty about output

[1] Note that the first term is independent of $\mathbf{x}$, and is therefore constant, this term is included for clarity in subsequent sections.

$\mathbf{y}$. That is, the output has a high *marginal entropy*, $\mathrm{H}[p(\mathbf{y}|\mathbf{x}, \mathcal{D})]$. However, given any particular parameter value θ drawn from the posterior, Equation (2.10) seeks a datapoint with low expected *conditional uncertainty*, $\mathbb{E}_{p(\theta|\mathcal{D})}\mathrm{H}[p(\mathbf{y}|\mathbf{x}, \theta)]$. This relates to the two sources of uncertainty described in Section 1.1: parameter uncertainty and observation noise. Equation (2.10) will reward datapoints whose output has high entropy due to parameter uncertainty, which is captured by the marginal predictive distribution $p(\mathbf{y}|\mathbf{x}, \mathcal{D})$, but penalizes uncertainty due to inherent noise, which is modelled by the likelihood $p(\mathbf{y}|\mathbf{x}, \theta)$. Equivalently, Equation (2.10) can be interpreted as seeking the $\mathbf{x}$ for which the parameters under the posterior make confident predictions, but these predictions are highly diverse. That is, the parameters *disagree* about the output $\mathbf{y}$, hence we name this formulation Bayesian Active Learning by Disagreement (BALD).

The equivalence follows from straightforward application of the properties of Shannon's entropy and has been noted in the literature [Lindley, 1956; Shewry & Wynn, 1987]. A number of active learning algorithms are derived starting from the mutual information in Equation (2.9) between observed variables and 'variables of interest' [Caselton & Zidek, 1984; Ertin *et al.*, 2003; Krause *et al.*, 2008], and thus may compute this quantity in either direction. However, the use of BALD, Equation (2.10), as an general-purpose alternative to posterior entropy minimization is not widely discussed. We argue that this reformulation is a valuable tool for information theoretic active learning, and it has a number of practical advantages. Noting the direct equivalence to information gain allows us to relate and formalize a number of popular active learning methods (Section 2.4). By working in this framework we derive new algorithms for specific domains in the subsequent chapters. However, we highlight BALD in its general form as a starting point for active learning algorithms in many other possible domains.

2.3.2 Computational Advantages

The BALD reformulation in Equation (2.10) can provide a number of computational advantages over direct entropy minimization in Equation (2.8). We discuss the immediate advantages in this section. In the next section we investigate empirically a further computational advantage from improved sample complexity if Monte Carlo is used to estimate the utility function.

A principal difference between Equations (2.8) and (2.10), is that Equation (2.10) computes entropies in output space $\mathcal{Y}$ rather than parameter space Θ. Often parameter space is high dimensional and so posterior entropies are usually intractable. One possibility is to use a histogram estimator [Paninski, 2003], but the number of bins

scales exponentially with the dimensionality of parameter space. Another possibility is to sample from the posterior and estimate Equation (2.8) using these samples. However, estimating entropies directly from samples in an unbiased manner is notoriously hard [Panzeri *et al.*, 2007]. Therefore, the intractable posterior entropy is often approximated directly from approximations to the posterior [Herbrich *et al.*, 2002; Krishnapuram *et al.*, 2004; Lewi *et al.*, 2007; MacKay, 1992b].

In Bayesian nonparametric models, such as Gaussian processes (GPs), parameter space is infinite dimensional and so entropies in parameter space are ill-defined. However, output space is often simple. For example, in regression and classification, the output is usually a uni-dimensional real valued or binary variable, respectively. The entropy of these scalars is usually straightforward to compute, either analytically or with a one-dimensional grid. Therefore, BALD can often compute expected changes to posterior entropy exactly, even with infinite dimensional parameter spaces. Furthermore, because Equation (2.10) does not compute the entropy of the posterior distribution directly, estimating information gain in this way is not tied to a particular method for approximating the posterior. Section 2.3.3 presents an empirical study on a toy problem to demonstrate that $\mathcal{U}(\mathbf{x})$ can be estimated more accurately from posterior samples with BALD than with direct estimation using Equation (2.8).

A second advantage that BALD has over the original formulation is that the posterior does not need to be updated until *after* a query has been made. Equation (2.8) requires calculating the updated posteriors after including every possible new datapoint $p(\theta|\mathcal{D}, \mathbf{x}, \mathbf{y})$. However, only the current posterior $p(\theta|\mathcal{D})$ appears in Equation (2.10). Suppose we have a pool of N possible inputs, and can receive one of $|\mathcal{Y}|$ possible outputs – for continuous $\mathbf{y}$ output space could be partitioned into a grid. To compute Equation (2.8) we must calculate $N|\mathcal{Y}|$ new posterior distributions. This will usually require expensive approximate inference. BALD requires updating the posterior only once per sample, after labelling the datapoint. This is the same as number of updates required by *passive* online learning. If c^{inf} is the cost of inference, and h^θ is the cost to compute the entropy of the posterior, then the complexity to compute Equation (2.8) on N candidates is $\mathcal{O}(N|\mathcal{Y}|c^{\text{inf}} + N|\mathcal{Y}|h^\theta)$. With BALD, the cost is $\mathcal{O}(c^{\text{inf}} + Nc^{\text{pred}} + Nh^{\mathbf{y}})$, where c^{pred} is the cost to compute the predictive distribution, and $h^{\mathbf{y}}$ is the cost to compute the entropies in output space. For many models, $c^{\text{pred}} \ll c^{\text{inf}}$ and $h^{\mathbf{y}} \ll h^\theta$.

Like any algorithm, BALD does not offer a free lunch. If output space is complicated, such as in structured models [Tsochantaridis *et al.*, 2004], then direct entropy minimization may be easier. Even if this is not the case, the terms in Equation (2.10) may still be non-trivial to compute. In practice, the first term is usually straightforward

because the marginal predictive distribution $p(\mathbf{y}|\mathbf{x}, \mathcal{D})$ is central to most applications of Bayesian machine learning. Therefore techniques for computing this quantity have been developed for many models. However, the second term, the posterior mean conditional entropy $\mathbb{E}_{p(\theta|\mathcal{D})}\mathrm{H}[p(\mathbf{y}|\mathbf{x}, \theta)]$ is more unusual. As we shall see in the following chapters, computing this term is normally the main challenge when implementing BALD.

2.3.3 Simulation: Estimating the Utility from Samples

A toy linear-Gaussian model is used to investigate empirically the accuracy of estimating $\mathcal{U}(\mathbf{x})$ from posterior samples, either directly (2.8), or with BALD (2.10). The likelihood function is $p(y|\mathbf{x}, \theta) = \mathcal{N}(y; \theta^\top \mathbf{x}, \sigma^2)$, where the parameters θ and input $\mathbf{x}$ are 10-dimensional real valued vectors and the output y is a scalar: $\mathcal{X} = \mathbb{R}^{10}$,$\mathcal{Y} = \mathbb{R}$, $\Theta = \mathbb{R}^{10}$. $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \Sigma)$ denotes a (in general, multivariate) Gaussian distribution over $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance matrix Σ. The noise level σ^2 is fixed and known. With a Gaussian prior on θ, the posterior is tractable and Gaussian, so its entropy can be computed analytically to assess the quality of the sampling approximations. Furthermore, in this model the posterior entropy is independent of the future observations, so $\mathcal{U}(\mathbf{x})$ can be computed without simulating any y values.

Even a 10-dimensional parameter space is impractical to grid up, so we approximate Equation (2.8) using Monte Carlo samples from the posterior,[1]

$$\mathrm{H}[p(\theta|\mathcal{D})] \approx -\frac{1}{N}\sum_{i=1}^{N} \log p(\theta_i|\mathcal{D}), \quad \theta_i \sim p(\theta|\mathcal{D})\,, \tag{2.11}$$

where N is the number of samples. Note that we do not have to integrate over y as the output value does not influence the posterior entropy. We may also use these samples to estimate $\mathcal{U}(\mathbf{x})$ using BALD,

$$\mathcal{U}(\mathbf{x}) \approx \mathrm{H}\left[\frac{1}{N}\sum_{i=1}^{N} p(y|\mathbf{x}, \theta_i)\right] - \frac{1}{N}\sum_{i=1}^{N} \mathrm{H}[p(y|\mathbf{x}, \theta_i)]\,. \tag{2.12}$$

With this simple linear-Gaussian model the entropy of the second term in Equation (2.12) is constant with respect to $\mathbf{x}$. The first term is the entropy of a mixture of N one-dimensional Gaussians, which can be computed using a one-dimensional grid.

We define the approximation error as the normalized difference in utility between

[1] In practice posterior samples will have often been computed already from the approximate inference algorithm. This is the case in Chapter 4.

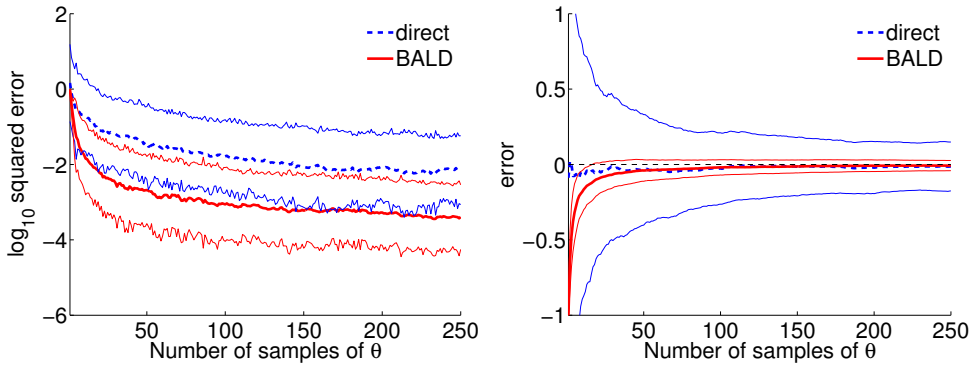

Figure 2.2: Error in the estimation of $\mathcal{U}(\mathbf{x})$ using samples directly (2.11) or with BALD (2.12). Thicker lines give the mean and thinner dash-dot lines indicate ± 1 s.d. over 200 repeats. The x-axis is the number of samples. *Left:* log squared error. *Right:* error, $\mathcal{U}_{\text{approx}}(\mathbf{x}) - \mathcal{U}_{\text{true}}(\mathbf{x})$.

the truth, computed analytically, and the Monte Carlo approximations,

$$\text{error} = \frac{\mathcal{U}_{\text{approx}}(\mathbf{x}) - \mathcal{U}_{\text{true}}(\mathbf{x})}{\mathcal{U}_{\text{true}}(\mathbf{x})} \,. \tag{2.13}$$

The experiment was repeated 200 times, resampling $\mathbf{x}$ each time. Figure 2.2 shows the distribution of errors over these runs as a function of the number of samples used to estimate $\mathcal{U}(\mathbf{x})$ by either method. The left-hand plot in Figure 2.2 shows that BALD provides a more accurate estimate of the information gain as the squared estimation error is over an order of magnitude smaller than with direct estimation of posterior entropy.

However, the right-hand plot in Figure 2.2 demonstrates that with very few samples (< 50) BALD tends to underestimate the entropy change. This is probably due to the mixture of Gaussians used to estimate the marginal predictive entropy (the term of Equation (2.12)). The true predictive distribution has infinitely many components. Each component has the same variance, and with few samples the finite mixture approximation is likely to yield an underestimate. In the extreme case of one sample, BALD will always underestimate this marginal entropy and will return a utility of zero. However, as indicated by the error bands, BALD yields an estimate with smaller variance and the bias is much smaller than the standard deviation of the direct estimate.

2.3.4 Extension: Nuisance Parameters and Focused Active Learning

In many settings some parameters are more important to learn than others. Here, in addition to the parameters of primary interest θ, the model has some other 'nuisance parameters' of lesser importance ϕ. For example, in Bayesian models, ϕ may correspond to hyper-parameters of the prior. Rather than collecting data that provides maximal information about the joint distribution over all of the parameters $p(\theta, \phi|\mathcal{D})$, we would like to focus learning efforts on the parameters of interest alone. In particular, we want to minimize the entropy of the marginal distribution $p(\theta|\mathcal{D})$. Equation (2.10) can be directly extended to this scenario,

$$\mathrm{I}[\theta; \mathbf{y}|\mathbf{x}, \mathcal{D}] = \mathrm{H}\left[\int p(\theta, \phi|\mathcal{D})d\phi\right] - \mathbb{E}_{p(\mathbf{y}|\mathbf{x},\mathcal{D})}\mathrm{H}\left[\int p(\theta, \phi|\mathbf{x}, \mathbf{y}, \mathcal{D})d\phi\right] \tag{2.14}$$

$$= \mathrm{H}[p(\mathbf{y}|\mathbf{x}, \mathcal{D})] - \mathbb{E}_{p(\theta|\mathcal{D})}\mathrm{H}[\mathbb{E}_{p(\phi|\theta,\mathcal{D})}p(\mathbf{y}|\mathbf{x}, \theta, \phi)]\,, \tag{2.15}$$

where Equation (2.14) is the information gain in the marginal written explicitly and Equation (2.15) is the computationally efficient rearrangement. The first term in this 'focused' BALD utility function is the entropy of the marginal predictive distribution, as in the original BALD formula (2.10). The second term differs, the integral over the nuisance parameters has moved inside the entropy. Intuitively, this only penalizes datapoints about whose output we are still uncertain if we know θ, but not ϕ. We only seek disagreement between the parameters of interest θ, and hence focus our efforts on refining the distribution over these parameters only.

In the next two sections we discuss two properties of posterior information-gain for active learning: inductivity and submodularity.

2.3.5 Inductive and Transductive Learning

In learning theory, *inductive* and *transductive* learning are distinguished [Vapnik, 2000]. Posterior information gain (2.8), and hence BALD, is an inductive approach. Inductive methods address the generic task of generalization from samples, whereas transductive methods minimize the expected loss on a particular test set or distribution. A transductive algorithm needs access to test-time information during training and exploits this information. Therefore, inductive algorithms seek good performance in a variety of possible test scenarios, but are not optimal for any single setting.

This distinction between inductive and transductive algorithms applies in active learning [Tong, 2001; Yu *et al.*, 2006]. It relates to the distinction between information

and decision theoretic methods. Decision theoretic algorithms are inherently transductive because they focus on loss minimization on a particular test distribution. However, information-theoretic transductive algorithms have also been proposed. These minimize the average *predictive* variance or entropy over regions of interest in input space [Cohn *et al.*, 1996; Krause *et al.*, 2008; MacKay, 1992b]. Note that this is not the same as BALD, which uses predictive entropies to minimize the *parameter* entropy.

These approaches have relative advantages and disadvantages. If the test inputs are known, transductive methods are likely to yield a smaller test loss because they focus on learning in that region. Inductive methods may choose to learn about parameters that have a small effect on the predictions in the interest region. However, transductive methods first require the interest region to be known, then a distribution over it to be defined. This is usually done using a set of reference points. These points may be placed in a grid [Krause *et al.*, 2008], but this is impractical in higher dimensions. Alternatively, samples from the pool may be used. However, in this case an inductive algorithm will naturally focus on the region of interest as well because the samples used to train the model are also from the pool.

An advantage of inductive methods is that information about future inputs is not required. Another advantage is by maximizing information gain in the parameters one can choose which parameters to learn actively using Equation (2.15). With transductive methods, the parameters are learnt selectively, but the algorithm designer cannot select them directly. Directly choosing parameters to focus learning on can be advantageous if: i) The values of particular parameters themselves are of primary interest. ii) Selective learning allows the algorithm to be improved in other ways. For example, in Chapter 3 we find that choosing the order in which parameters are actively learnt appropriately is crucial for robust active GP regression. In Chapter 7 we learn about a new user in a recommendation system who is represented by a few parameters in a much larger model.

2.3.6 Myopic Assumption and Submodularity

The utility functions presented so far have been *myopic* or *greedy*. That is, they seek the optimal $\mathbf{x}$ as if it were the last datapoint to be selected. However, often one has a budget to collect a set of L samples, X, and receive a set of labels Y. In this case we want to maximize information gain over the entire set,

$$\mathcal{U}(X) = \mathrm{H}[\theta|\mathcal{D}] - \mathrm{H}[\theta|Y, X, \mathcal{D}]\,. \tag{2.16}$$

Unfortunately, optimizing Equation (2.16) is, in general, NP-hard in the horizon length L [Ko *et al.*, 1995; Krause & Guestrin, 2005]. Therefore, a common approach is to greedily maximize the utility with respect to the next sample alone [Dasgupta, 2005; Heckerman *et al.*, 1994]. Fortunately, under certain conditions, the greedy strategy is near-optimal. We now discuss when these conditions apply to information theoretic active learning.

The L-step utility (2.16) is a set function. A set function $F(Y) : 2^{\mathcal{Y}} \mapsto \mathbb{R}$ is a function whose input is a set Y and (usually) outputs a real value. Submodularity is an extension of convexity to set functions. Intuitively, submodular set functions exhibit 'diminishing returns'. More precisely, adding an element y to a set Y^{small} produces a greater increase to F than adding y to Y^{large}, where $Y^{\text{large}} \supset Y^{\text{small}}$. A set function is submodular if for all Y^{large} and Y^{small}, and every $y \notin Y^{\text{large}}$

$$F(Y^{\text{small}} \cup y) - F(Y^{\text{small}}) \geq F(Y^{\text{large}} \cup y) - F(Y^{\text{large}}) \,. \tag{2.17}$$

A set function is *non-decreasing* if $F(Y^{\text{large}}) \geq F(Y^{\text{small}})$. The key result of Nemhauser *et al.* [1978] is that if a set function is submodular, non-decreasing, and $F(\emptyset) = 0$, then greedy, sequential maximization is guaranteed to produce a solution whose function value is within $(1 - 1/e) \approx 63\%$ of the global optimum.

Equation (2.16) is a set function from labels Y to information gain. Due to the 'information never hurts' bound [Cover *et al.*, 1994], F is non-decreasing, and clearly $F(\emptyset) = 0$. Unfortunately, as noted in Krause *et al.* [2008] information gain is not in general submodular. For example, consider a collection of three variables (y_1, y_2, z). We wish to gain information about z by observing y_1 or y_2; z may be thought of as the 'parameter of interest' $z \equiv \theta$. The input variable $\mathbf{x}$ is simply the index 1 or 2. Suppose (y_1, y_2, z) are jointly Gaussian distributed as

$$p(y_1, y_2, z) = \mathcal{N}\left([y_1, y_2, z]^\top; \mathbf{0}, \Sigma\right) \,, \quad \text{where } \Sigma = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 0 \\ 1 & 0 & 2 \end{bmatrix} \,. \tag{2.18}$$

The information gain in observing set Y is $\mathrm{IG}(Y) = \mathrm{H}[z] - \mathrm{H}[z|Y]$. y_2 and z are independent, therefore $\mathrm{IG}(y_2) - \mathrm{IG}(\emptyset) = 0$. Now suppose that we have observed y_1, the joint distribution of the remaining variables given y_1 is

$$p\left(\begin{bmatrix} y_2 \\ z \end{bmatrix} \middle| y_1\right) = \mathcal{N}\left(\frac{1}{2}\begin{bmatrix} y_1 \\ y_1 \end{bmatrix}, \begin{bmatrix} 1.5 & -0.5 \\ -0.5 & 1.5 \end{bmatrix}\right) \,.$$

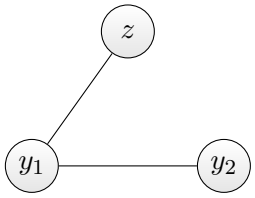

Figure 2.3: Graphical model for the joint Gaussian in Equation (2.18).

Now y_2 and z are conditionally dependent, y_2 provides information about z, $\mathrm{IG}(y_1 \cup y_2) - \mathrm{IG}(y_2) = \mathrm{H}[z|y_1] - \mathrm{H}[z|y_1, y_2] > 0$. Therefore, the increase in information gain when adding y_2 to y_1 is greater than when adding y_2 to $\emptyset$. Since $\emptyset \subset \mathbf{y}_1$, $\mathrm{IG}(Y)$ does not exhibit diminishing returns and is not submodular.

Fortunately, under certain conditional independence conditions, information gain will be submodular. The following theorem is given in Krause & Guestrin [2005].

Theorem 1. *Let S, U be disjoint subsets of (a finite set of random variables) V such that the variables in S are independent given U. Let 'information gain' be $IG(A) = H[U] - H[U \setminus A|A]$, where $A \subseteq W$, for any $W \subseteq S \cup U$. Then IG is submodular and non-decreasing on W, and $IG(\emptyset) = 0$.*

Theorem 1 seems a little daunting, but it has a simple intuition. We wish to learn about some variables U by observing some variables A chosen from a set W. W includes both the variables we are learning about U, or some extra variables S. Provided that these extra variables S are independent given the variables that we are learning U, then the information gain is a submodular set function. There is one further detail; if we are selecting A from U, then we only care about the information gain in the remaining variables in U, $\mathrm{IG}(A) = \mathrm{I}[A, U \setminus A]$.

Information theoretic active learning in the discriminative setting in Figure 2.1 is a special case of Theorem 1. Identifying U with the parameters θ, we never observe U directly, so $W = S$. Furthermore, since $S \cap U = \emptyset$ we have $U \setminus A = U$. S is the set of labels Y in Equation (2.16). The inputs X do not feature in the Theorem, they just index the set of labels. For $\mathrm{IG}(\theta)$ to be submodular and non-decreasing the variables in Y must be independent given θ. This is the case in Figure 2.1 as θ forms a Markov blanket between the variables $\mathbf{y}_i$.

So what is different in the joint Gaussian model in Equation (2.18)? The graphical model for Equation (2.18) is in Figure 2.3. Here, z does not form a Markov blanket between y_1 and y_2, so this figure cannot be mapped to the graphical model in Figure 2.1 and does not satisfy the conditional independence condition required by Theorem 1.

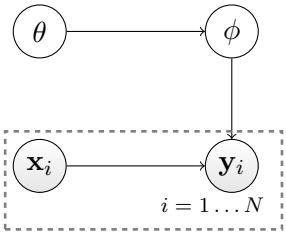

Figure 2.4: Graphical model for the discriminative learner with parameters θ and hyperparameters ϕ.

In summary, if the model can be mapped to the discriminative framework in Figure 2.1, then the information theoretic utility function in Equation (2.8) is submodular and so the myopic assumption only results in a small constant loss of utility.

However, for 'focused' information theoretic active learning this is not always the case. Consider a hierarchical Bayesian model in Figure 2.4. Suppose that the parameters of interest θ are the hyper-parameters and the 'lower level' parameters are considered to be nuisance parameters ϕ. For example, in Section 3.3 we want to learn the hyper-parameters of a GP kernel but do not care about the latent function. In this case BALD in Equation (2.15) integrates out the nuisance parameters. After integrating out ϕ, the outputs $\mathbf{y}_i$ are dependent, even when conditioned on θ. Therefore, with focused active learning, BALD may not be submodular. In practice we found that the myopic strategy performed well empirically, but investigating look-ahead techniques could produce further improvements here.

2.4 Literature Review

Active learning has been widely studied in machine learning, statistics and experimental and social sciences, where it is also referred to as *query learning* or *optimal experimental design* (OED). We outline some of the key related algorithms. It is not possible to cover the entire field, for comprehensive coverages see the textbooks Fedorov [1972] and Settles [2012].

2.4.1 Classical Optimal Experimental Design

Classical OED, developed in statistics, focuses primarily on linear regression,

$$y = \theta^\top \mathbf{x} + \epsilon \,. \tag{2.19}$$

The noise is usually Gaussian distributed with zero mean, $\epsilon \sim \mathcal{N}(0, \sigma^2)$. In this case the maximum likelihood estimator (MLE) of θ is equivalent to the least squares estimator. The MLE is $\hat{\theta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$, where $\mathbf{X}$ is the $N \times D$ 'design matrix' consisting of N input vectors $\mathbf{x}$, and $\mathbf{Y}$ is the vector containing N real valued outputs y. The variance of the MLE is given by $\sigma^2(\mathbf{X}^\top \mathbf{X})^{-1}$, the inverse of the variance is the Fisher Information (FI) matrix for this model. The FI provides an alternative to Shannon's entropy for measuring the information in a sample $\mathbf{x}$ about the unknown parameters θ. With linear-Gaussian models classical OEDs maximize the FI, or equivalently minimize the variance of the MLE, with respect to the next datapoint to be added to the design matrix. However, because the FI is a matrix quantity, not a scalar like the posterior entropy, a number of different utility functions have been proposed. These have alphabetized names, such as A-, D-, or E- optimality. We discuss a few of the most famous amongst these, others can be found in Atkinson [1988].

D-optimal design maximizes the determinant of $\mathbf{X}^\top \mathbf{X}$. This is equivalent to the Bayesian information gain criterion used by BALD (2.8) if a flat (non-informative) prior on θ is used. For example, if we have a zero mean Gaussian prior with covariance Σ, then the posterior $p(\theta|\mathbf{X}, \mathbf{Y})$ is Gaussian with covariance $\sigma^2(\mathbf{X}^\top \mathbf{X} + \Sigma^{-1})^{-1}$. The entropy of a Gaussian is proportional to the log determinant of its covariance matrix, and so with a non-informative prior, such as $\Sigma = \lim_{\epsilon \to 0} \frac{1}{\epsilon} I$, the determinant of the design matrix is a monotonic function of the posterior entropy. Bayesian D-optimality extends D-optimality to include a prior. In this case $|(\mathbf{X}^\top \mathbf{X} + \Sigma^{-1})^{-1}|$ is minimized, which is equivalent to posterior entropy minimization with the linear-Gaussian model (2.19). Bayesian equivalents of most classical OEDs are formed by replacing the covariance of the MLE $(\mathbf{X}^\top \mathbf{X})^{-1}$ with the posterior covariance matrix $(\mathbf{X}^\top \mathbf{X} + \Sigma^{-1})^{-1}$ [Chaloner & Verdinelli, 1995].

D-optimal design can be motivated using other utility functions on θ. Posterior entropy minimization is equivalent to minimizing expected log loss on the parameters. If the true parameter is $\hat{\theta}$ we can define a 'score' for the posterior distribution as $S(\hat{\theta}, p(\theta|\mathcal{D})) = \log p(\hat{\theta}|\mathcal{D})$. This score rewards posteriors with high density at the true parameter value. The expected score given the current posterior beliefs is

$$E_{p(\theta|\mathcal{D})}[S(\theta, p(\theta|\mathcal{D}))] = -\mathrm{H}[p(\theta|\mathcal{D})]\,.$$

Maximizing the expected score with respect to the data $\mathcal{D}$ is therefore equivalent to posterior entropy minimization. $S(\theta, p(\theta|\mathcal{D}))$ is an example of a proper scoring rule. These are scores that reward honest estimates of uncertainty. Proper scoring rules are

beyond the scope of this thesis, see Dawid [2007]; Huszár [2013] for details. Other scoring rules may be used, and with linear-Gaussian models, D-optimality is also equivalent to maximizing the Brier score [Brier, 1950]. However, unlike Shannon's information, which is induced by the log score, other scoring rules do not result in symmetric utility functions. Hence, computationally efficient rearrangements like BALD in Section 2.3.1 are not generally possible.

A-optimality minimizes the trace of $(\mathbf{X}^\top\mathbf{X})^{-1}$. The Bayesian equivalent, minimizing the trace of the posterior covariance, is motivated if a point estimate of the parameters $\hat{\theta}$ is sought. This is because A-optimality is equivalent to minimizing the expected squared Euclidean distance of an estimator $\hat{\theta}$ to the true parameters, $\mathcal{U}(\mathbf{x}) = -\mathbb{E}_{p(\mathbf{y},\theta|\mathbf{x},\mathcal{D})}||\theta - \hat{\theta}||_2^2$.[1] The trace operation minimizes the average variance of the estimator over the dimensions of θ. If this average is replaced with a minimax criterion, then the objective is called E-optimality. E-optimality minimizes the maximum posterior variance for any linear projection of θ, $\operatorname{argmax}_{\mathbf{c}:||\mathbf{c}||=\mathbf{1}} \mathbf{c}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{c}$.

E-optimality is mathematically similar to G-optimality which minimizes $\operatorname{argmax}_{\mathbf{x}^\star} \mathbf{x}^{\star\top}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}^\star$, where $\mathbf{x}^\star$ is any point in input space. The Bayesian equivalent has a decision-theoretic flavour: to minimize the greatest predictive variance at any point in input space. This arises because the variance of the Bayesian predictive distribution (1.2) for the linear-Gaussian model is $\operatorname{Var}[p(\mathbf{y}^\star|\mathbf{x}^\star,\mathcal{D})] = \sigma^2\mathbf{x}^{\star\top}(\mathbf{X}^\top\mathbf{X} + \Sigma^{-1})^{-1}\mathbf{x}^\star$.

These designs have been extended to nonlinear regression models, $y = f(\theta^\top\mathbf{x}) + \eta$, where f is a nonlinear function. In the linear-Gaussian model (2.19) most designs, such as D-optimality, are independent of the output $\mathbf{y}$. This is also true for other models such as the generalized linear model with Poisson likelihood and exponential link function [Lewi *et al.*, 2009]. With linear regression the design is also independent of the parameters. In these cases, the optimal sequence of measurements can be computed *a priori*. However, this is not the case with nonlinearities. Similar to active learning, classical designs often take an iterative approach. They estimate the parameter $\hat{\theta}$ and then select the next measurement using this estimate [Chaloner & Verdinelli, 1995]. In Bayesian OED, the posterior usually becomes intractable, so a number of approximations have been proposed. The most popular methods approximate the posterior with a Gaussian, whose mean is equal to the MLE or MAP estimate $\hat{\theta}$. The variance of the approximation is computed using the inverse of the FI matrix. In general, the FI matrix is the Hessian of the negative log likelihood surface, and is a function

[1] Note that an expectation of the unseen $\mathbf{y}$ is needed here because, although the posterior variance is not a function of $\mathbf{y}$, the estimator $\hat{\theta}$ may be.

	Cost Function	Bayesian Interpretation/Equivalence
D	$\lvert(\mathbf{X}^\top\mathbf{X})^{-1}\rvert$	max Shannon information gain in θ
A	$\mathrm{tr}[(\mathbf{X}^\top\mathbf{X})^{-1}]$	min variance of a point estimate for θ
E	$\mathrm{argmax}_{\mathbf{c}:\lvert\lvert\mathbf{c}\rvert\rvert=\mathbf{1}}\, \mathbf{c}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{c}$	min max post. variance for any projection of θ
G	$\mathrm{argmax}_{\mathbf{x}^\star}\, \mathbf{x}^{\star\top}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}^\star$	min max predictive variance at locations $\mathbf{x}^\star$

Table 2.1: A few classic optimal experimental designs. Cost is 'negative utility'. Bayesian equivalents are formed by replacing $\mathbf{X}^\top\mathbf{X}$ by $\mathbf{X}^\top\mathbf{X} + \Sigma^{-1}$, where Σ is the prior covariance of θ. More general equivalents for nonlinear regression models are given by replacing $\mathbf{X}^\top\mathbf{X}$ with the general form of the Fisher Information Matrix, $F(\hat{\theta})$.

of θ and $\mathbf{y}$, unlike for the linear model where it is simply equal to $\mathbf{X}^\top\mathbf{X}$. This is the Laplace approximation, widely used in machine learning and statistics [Kass & Raftery, 1995]. The classical design criteria can then be estimated for nonlinear designs using the approximate distribution over θ.

Table 2.1 summarizes the classical designs presented in this Section.

2.4.2 Information Theoretic Methods

Maximizing Shannon's information for Bayesian experimental design was originally proposed in Lindley [1956]. This paper studies the fundamental properties of this objective. These properties are studied further in Fedorov [1972]; MacKay [1992b]. In the latter, the posterior information gain for an arbitrary parametric model with a Gaussian likelihood is approximated using the Laplace approximation. As described in Section 2.3.2, performing active learning by computing parameter entropies can be expensive, even with a Laplace approximation. More recently, a fast algorithm that minimizes posterior entropies directly is proposed in Lewi *et al.* [2007].

Algorithms that work with predictive distributions tend to be more easily applicable, but do not necessarily optimize an objective, such as information gain. The most well known of these algorithms is Maximum Entropy (or uncertainty) Sampling (MES) [Shewry & Wynn, 1987]. MES selects datapoints with the largest predictive entropy. This corresponds to the first term in BALD, Equation (2.10). MES was originally proposed for regression models with constant observation noise, such as the linear-Gaussian model. Here, the second term in Equation (2.10) is constant with respect to $\mathbf{x}$, so it can be ignored. However, MES is not equivalent to BALD with more complex models, such as classification or heteroscedastic models. This is because it fails to differentiate between parameter uncertainty and observation noise. As a result

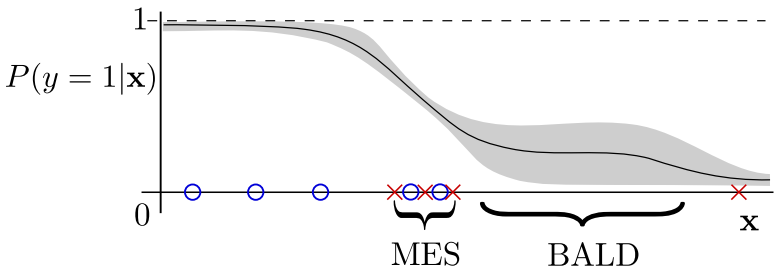

Figure 2.5: Toy example of classification with a 1D input. Circles and crosses denote labelled datapoints. The plot shows the mean and variance of the predictive distribution. Maximum Entropy Sampling (MES) selects data from regions of high marginal uncertainty. BALD seeks more informative data in the region of uncertainty due to high posterior variance.

MES over-samples in regions of noisy data. For example, points near a classification decision boundary may be very noisy, and hence provide no information about the model. BALD, on the other hand, directly maximizes the information gain so does not exhibit such pathologies. BALD will learn that a region has high inherent uncertainty thus will explore elsewhere. Figure 2.5 illustrates this point on a 1D classification example.

The mutual information between measured variables and 'variables of interest' has been proposed in various applications. These include tracking via Bayesian filtering [Ertin *et al.*, 2003]. Here, the hidden state is the variable of interest, and with a linear Gaussian model, the utility function reduces to MES. Fuhrmann [2003] applies mutual information in noisy communication channels with binary variables, in this case entropy computations are simple. With BALD, the interest variables are model parameters, the above approaches do not learn these.

Another application of mutual information is for monitoring environmental variables. Here, the mutual information between measured locations and interest locations is maximized [Caselton & Zidek, 1984]. This approach is transductive (see Section 2.3.5), it learns the function optimally just at the interest locations. This technique is applied with GP regression in Krause *et al.* [2008], where a grid of interest locations is specified. However, gridding the region of interest is impractical for higher dimensional problems.

The above mutual information based methods do not consider hyperparameter learning which we will address for GP regression in Chapter 3. Hyperparameter uncertainty has been considered in a similar setting to GPs, Empirical Kriging. Zimmerman

[2006] minimize the predictive variance at interest locations for active sensor placement with unknown kernel hyperparameters. A linearized approximation of the effect of hyperparameter uncertainty (measured using Fisher Information) on the predictive variance is added to the utility function. Focused BALD (2.15) minimizes hyperparameter uncertainty more directly.

2.4.3 Data Subsampling

Active learning is primarily used for collecting data when labelling is expensive. However, if the dataset is very large, or there is a continuous stream of data, and one does not have the resources to process all of the points then active learning can be used to subsample the dataset to a feasible size. This simple approach to 'compressing' a large dataset is referred to as the *subset of data* approximation [Quiñonero-Candela & Rasmussen, 2005]. Subset of data methods differ to active learning because the label $\mathbf{y}$ can be observed prior to sampling, therefore, although active learning techniques can address the subset of data problem, the reverse is not possible.

A popular dataset subsampling algorithm for GPs is the Informative Vector Machine (IVM) [Herbrich *et al.*, 2002].[1] Like BALD, the IVM maximizes information gain in the parameters. However, $\mathbf{y}$ is observed so the IVM does not work explicitly with predictive distributions, but works directly in parameter space. However, GPs have an infinite dimensional parameter, the latent function. The IVM computes parameter entropies in the marginal subspace that corresponds to the observed datapoints. Now, the entropy decrease after inclusion of a new point is calculated efficiently using the GP covariance matrix. Figure 2.6 shows the difference between this approach and BALD. As a result, the IVM has a transductive bias, the locations of the observed data defines the parameters whose entropy are minimized. BALD is inductive because it works implicitly with the full infinite-dimensional latent function.

Although the IVM and BALD are motivated by the same objective, they have different properties when approximate inference is carried out, such as in Gaussian process classification (GPC). At time t both methods have an approximate posterior $q_t(\theta|\mathcal{D})$. This can be updated with the likelihood of a new data point $p(y_{t+1}|f, \mathbf{x}_{t+1})$, yielding $\hat{p}_{t+1}(\theta|\mathcal{D}, \mathbf{x}_{t+1}, y_{t+1}) = \frac{1}{Z} q_t(\theta|\mathcal{D}) p(y_{t+1}|f, \mathbf{x}_{t+1})$. If the posterior at $t+1$ is approximated directly one gets $q_{t+1}(\theta|\mathcal{D}, \mathbf{x}_{t+1}, y_{t+1})$. BALD calculates the entropy difference between q_t and $\hat{p}_{t+1}$, without having to compute q_{t+1} for each candidate $\mathbf{x}_{t+1}$ and label y_{t+1}. In contrast, the IVM calculates the entropy change between q_t

[1] GPs are introduced in Section 3.1.

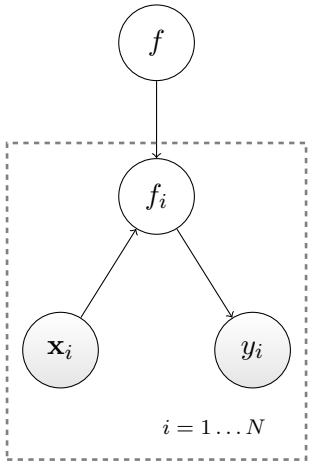

Figure 2.6: Graphical model for Gaussian processes. The infinite dimensional latent parameter f gives rise to scalar latent variables f_i at each input location, $\mathbf{x}_i$. The directed links from the top layer to the 2nd layer are deterministic marginalizations, not probabilistic dependencies. Given the latent function f_i the likelihood function generates the labels y_i. The IVM calculates entropies in the second layer, whereas BALD calculates entropy changes to the entire f at the top.

and q_{t+1}, and so must compute the new posterior for all possible queries. Methods that compute posterior entropies directly require $\mathcal{O}(N|\mathcal{Y}|c^{\text{inf}} + N|\mathcal{Y}|h^{\theta}))$ computations per sample since the posterior must be recomputed for all candidates.[1] Therefore, because c^{inf} enters the complexity per candidate, the IVM for GPC must perform fast approximate inference using assumed density filtering (ADF). BALD requires only one posterior update, so costs $\mathcal{O}(c^{\text{inf}} + Nc^{\text{pred}} + Nh^{\mathbf{y}})$. Therefore, more accurate iterative procedures, such as expectation propagation (EP) [Minka, 2001b], can be used. ADF is a single-pass version of EP, so this results in q_{t+1} being a direct approximation to $\hat{p}_{t+1}$ which BALD implicitly works with.

2.4.4 Decision Theoretic Methods

Decision theoretic methods directly minimize the expected loss, measured using the Bayes posterior risk (2.1). In many tasks, the loss is hard to quantify, but in classification, the misclassification rate provides a sensible loss function [Kapoor *et al.*, 2007; Roy & McCallum, 2001; Zhu *et al.*, 2003]. These methods assume knowledge of the location of the test data and then minimize the expected 0/1 classification loss. However, as

[1] For homoscedastic regression with fixed hyperparameters the posterior variance is independent of $\mathbf{y}$, so only $\mathcal{O}(Nc^{\text{inf}} + Nh^{\theta})$ updates are needed. With further approximations to the likelihood the IVM can be sped up further for regression [Seeger *et al.*, 2003].

Bayesian	Non-Probabilistic
parameter setting	hypothesis/hyperplane
posterior distribution	version space (VS)
posterior entropy	log volume of VS
Gaussian approx to posterior e.g. EP, Laplace	hypersphere approx to VS e.g. Tong & Koller [2002]
samples from posterior	committee members
MES	margin sampling
Equation (2.8)	directly minimizing VS volume
BALD	QBC with an infinite committee and probabilistic voting

Table 2.2: Analogies between Bayesian active learning and non-probabilistic active learning methods for SVMs.

noted in Roy & McCallum [2001], a limitation of decision theoretic methods is the computational cost. In general, with T test points they require $\mathcal{O}(N|\mathcal{Y}|c^{\text{inf}} + TN|\mathcal{Y}|c^{\text{pred}})$ computations to calculate the posterior risk on the test set for all possible new datapoints. Incremental re-training and re-classification can speed up computations with certain models [Roy & McCallum, 2001]. Further computational savings require approximations such as pruning the query search space or sub-sampling the test set.

2.4.5 Non-Probabilistic Methods

Non-probabilistic methods have analogies with Bayesian active learning. The most well known non-probabilistic active learning methods are for Support Vector Machines (SVMs) [Seung *et al.*, 1992; Tong & Koller, 2002]. SVMs are a sparse non-probabilistic model mostly used for binary classification. SVMs classify the data using a hyperplane with maximal separation from each class.[1] The set of possible hyperplanes that correctly classifies the training data is called version space (VS). An introduction to SVMs can be found in Burges [1998].

A popular approach to SVM active learning minimizes the number of possible hypotheses (hyperplanes), which is done by minimizing the volume of VS. VS can be interpreted as a deterministic equivalent to the posterior distribution, and its volume is analogous to the posterior entropy. If a uniform (improper) prior and a deterministic likelihood are used then the log volume of VS is equivalent to the entropy of the posterior. Each observed datapoint defines a plane in VS and hypotheses consistent

[1] SVMs are not limited to linear decision planes, they can use kernels to produce complex decision boundaries [Schölkopf & Smola, 2002].

with the data lie on one side of this plane. The goal of active learning with SVMs is to sample a datapoint that cuts VS in half. Thus, after receiving the label, at worst almost half of the hypotheses are eliminated.

However, just as Bayesian posterior distributions can be intractable, VS can become complex after observing many datapoints, and the relevant volumes hard to compute. Therefore, Tong & Koller [2002] propose approximating a complex VS with simpler shapes, such as hyperspheres. Their simplest approximation fits the largest possible hypersphere into VS and selects the point whose dual hyperplane falls closest to the centre of this sphere. This is equivalent to margin sampling, which chooses the point closest to the decision boundary [Campbell *et al.*, 2000]. Similar to MES, margin sampling can over-sample noisy data beside the boundary. This approximation of VS using a simple shape is similar to Bayesian inference methods that approximate the posterior with a simple distribution, such as a Gaussian.

Query by Committee (QBC) sidesteps complex version spaces and, like BALD, works directly with predictions [Seung *et al.*, 1992]. QBC samples parameters from VS 'committee members', each of whom makes a prediction, 'votes', on the label of $\mathbf{x}$. The $\mathbf{x}$ with the most balanced vote is selected, this is termed the 'principle of maximum disagreement'. If Equation (2.10) is approximated by sampling θ, then BALD resembles QBC, but with a probabilistic measure of disagreement. By discarding confidence estimates QBC can exhibit the same pathologies as MES, namely over-sampling noisy data. QBC was proposed for classification, and to extend this algorithm directly to other scenarios, such as regression, the disagreement measure needs to be re-designed [Burbidge *et al.*, 2007]. McCallum & Nigam [1998]; Melville *et al.* [2005] replace the deterministic voting in QBC with the Jensen-Shannon (JS) divergence between a finite number of predictive distributions. These works do not uncover the link to information gain in the parameters; BALD also minimizes the JS-divergence of predictive distributions, but from infinitely many committee members drawn from the posterior. Table 2.2 summarizes the analogies between Bayesian and non-probabilistic methods for active learning.

2.4.6 Summary

We have presented a framework for Bayesian information theoretic active learning called BALD. This framework directly exploits the rearrangement of parameter entropies to predictive entropies. There are many approaches to active learning, but BALD can be advantageous for a number of reasons:

- It is inductive, so does not make any assumptions about a future decision task, loss or test set.
- With many models, the utility function is smooth and so BALD may be applied to both continuous sampling and pool-based active learning.
- The utility function is often, but not always, submodular and hence greedy maximization is near-optimal.

The above advantages are specific to the classical information gain objective function in Equation (2.8). The potential advantages of the rearrangement that BALD exploits (2.10) and the extension in Equation (2.15) are:

- The required computations are not inherently tied to a particular model or inference method.
- If output space is 'simpler' than parameter space, as is often the case, then the required entropies are more straightforward to compute.
- The number of (approximate) posterior updates is reduced from one per *possible* datapoint to one per *observed* datapoint.
- One can focus upon learning particular variables in the model.

However, computation of the utility in Equation (2.10) may still be non-trivial. Whether BALD is computationally useful depends on the particular task and model. Whether BALD is practically useful is a matter of empirical performance. In the following chapters we apply this framework in machine learning and scientific domains to yield efficient algorithms with strong practical performances.