Bayesian Active Learning for Maximal Information Gain on Model Parameters

Kasra Arnavaz*†, Aasa Feragen‡*, Oswin Krause* and Marco Loog§*
kasra@di.ku.dk, afhar@dtu.dk, oswin.krause@di.ku.dk, m.loog@tudelft.nl
*Department of Computer Science, University of Copenhagen
†DanStem, University of Copenhagen
‡DTU Compute, Technical University of Denmark
§Pattern Recognition Laboratory, Delft University of Technology

***Abstract*—The fact that machine learning models, despite their advancements, are still trained on randomly gathered data is proof that a lasting solution to the problem of optimal data gathering has not yet been found. In this paper, we investigate whether a Bayesian approach to the classification problem can provide assumptions under which one is guaranteed to perform at least as good as random sampling. For a logistic regression model, we show that maximal expected information gain on model parameters is a promising criterion for selecting samples, assuming that our classification model is well-matched to the data. Our derived criterion is closely related to the maximum model change. We experiment with data sets which satisfy this assumption to varying degrees to see how sensitive our performance is to the violation of our assumption in practice.**

I. Introduction

The default procedure to train a machine learning model is to learn from randomly gathered data. Active learning investigates whether we could reach at least the same performance as random sampling with fewer samples if we had the control over which samples to gather. While it might seem intuitive that the answer to the previous question should be positive, a consistent solution has been elusive so far [1], [2] . There have been several heuristics that propose sampling strategies based on common sense [3], and while they do outperform random sampling on occasion, it is not clear ahead of time if they will. What is missing is the set of conditions under which one is guaranteed to perform at least no worse than random sampling. This has been a hurdle which has prevented active learning strategies from replacing random sampling altogether.

In this paper, we employ a Bayesian framework for both model fitting as well as active learning to investigate the possibility of optimal data gathering—at least under certain assumptions. Parameters of our classification model are inferred through the approximation of the posterior distribution of parameters by a Gaussian distribution. We, in turn, look for data which maximally reduces the expected entropy of this Gaussian posterior. Our derivation follows that of MacKay [4] for regression problems. We show that in the case of logistic regression, our sampling strategy has a nice, analytical form, which is closely related to maximum model change [5]. Moreover, in the limit of infinite data our sampling strategy behaves similarly to the "decision boundary sampling" [6]. We illustrate the behavior of our sampling strategy on a number of data sets, and in this context discuss how our derivations rely on our model being well-matched to the data. Having said that, we have no Bayesian way of quantifying the extent to which the model is well-matched to the data. Bayesian model testing is only able to compare different proposed models, and is in principle unable to detect if all those models are far from the truth. We conclude with a discussion of how this problem might be alleviated by extending the logistic regression model to a neural network.

The outline of this paper is as follows: Section II reviews how inference and active learning fit into the data modeling process, which we derive in detail in Section III and Section IV, respectively, following the analogous derivations of [7] and [4] for regression models. Particularly, in Subsection III-A we apply Laplace's method to Bayesian inference of model parameters and in Subsection III-B we address inference of a hyperparameter in an evidence maximization framework. In Section IV, we briefly review measures of information gain and introduce an active learning strategy to gain maximal information on model parameters. A performance criterion that incorporates parameter uncertainty is given in Section V. In Section VI, we experimentally validate our active learning scheme on a range of data sets which satisfy our data set assumptions to varying degrees. Finally, Section VII draws on our findings to discuss if any guarantees can be given about the optimality of our proposed method. We end by suggesting how the same line of thought could potentially lead to a better solution.[1]

II. Data Modeling

Before we delve into details, we find it of great value to remind the reader of the role of inference and data gathering in data modelling process [8]. We, as the data scientist, typically have a few candidate models (hypotheses), one of which we postulate underlies the data fairly well. These models might also include some parameters which we would like to set. Bayesian inference is the tool that enables us to reliably compare models and fit their parameters, taking into

[1]The code is available at https://github.com/kasra-arnavaz/Bayesian-Active-Learning.

account the information produced by the data and our prior knowledge about which model or parameter is more likely. As a result, one could think of the following scenario for optimal data gathering [7]: Firstly, we may look for data that gives us maximal expected information on the plausibility of our candidate models. We may use this information to come up with new models we now find likely. We may continue this iteration until we have narrowed down our hypotheses to one. Once we have decided on one particular model, we would then aim to gather data which gives us maximal expected information on the parameters of that particular model. A criterion that maximizes the discrimination of two models has been given in [4].

In this paper, we assume we have identified the so-called 'true model', and thus look for data that reduces the expected uncertainty of parameters the most. This assumption will incur consequences which we will discuss in Section VII.

III. Bayesian inference

Active learning goes hand in hand with uncertainty quantification. Bayesian inference provides a coherent basis for uncertainty quantification, where uncertainties are expressed by probability distributions. We refer the reader to [7] for an in-depth review of Bayesian inference for regression problems. Below, we will specifically work with the logistic regression model. This choice is made in part because it leads to analytical derivations, and in part because it is commonly used as the final output of more flexible deep learning classification networks.

A. Inference of model parameters $\mathbf{w}$

In discriminative modeling, we are interested in finding a mapping from the input space to the target space which generalizes well to unseen data.

Suppose we have observed N input-target pairs as $\mathcal{D} = \{\mathbf{x}_n, t_n\}$, where $\mathbf{x}_n \in \mathbb{R}^k$, $t_n \in \{0, 1\}$, and $n = 1, ..., N$. A parametric model with parameters $\mathbf{w} \in \mathbb{R}^k$ is specified by its functional form $y(\mathbf{x}; \mathbf{w})$, the likelihood distribution $P(\{t_n\}|\{\mathbf{x}_n\}, \mathbf{w})$[2], and the prior distribution $P(\mathbf{w})$[3]. The functional form specifies a space of functions $\mathcal{F}$ through which one can wander by changing the parameters. The likelihood determines which functions in $\mathcal{F}$ fit the observed N samples better than the others. Prior establishes our prior belief about the plausibility of functions in $\mathcal{F}$. As our observed samples might also include outliers, prior usually takes a form which favors smooth functions to prevent the model from fitting to outliers and facilitate generalization. Our posterior belief of parameters would then be given by Bayes' theorem as

$$P(\mathbf{w}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{P(\mathcal{D})}. \tag{1}$$

[2]For notational convenience, from here onwards, we will write $P(\{t_n\}|\{x_n\}, \mathbf{w})$ as $P(\mathcal{D}|\mathbf{w})$, taking into account that in discriminative models we model targets given inputs.

[3]The prior could also depend on nuisance parameters, which we are momentarily assuming to be already integrated out.

One common prior which gives a varying degree of smoothness is a zero-mean Gaussian prior with variance $1/\alpha$ written as

$$P(\mathbf{w}|\alpha) = \frac{1}{Z_E} \exp\left(-\alpha E(\mathbf{w})\right), \tag{2}$$

where

$$E(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} \tag{3}$$

and

$$Z_E = \sqrt{(2\pi/\alpha)^k}. \tag{4}$$

For small α, the variance would be large, making the prior more like a uniform distribution and thus prior takes a neutral position in the posterior belief of the parameters in (1). In this case, parameters which fit the observed data better would be more plausible. Conversely, large α leads to a small variance, leaving the posterior to be dominated by the prior. In this case, over-smooth functions would be more plausible which might not fit the observed data well. In Subsection III-B, we will apply Bayes' theorem to find which values of hyperparameter α are more likely.

We let the output of our model to estimate the probability of the positive class i.e. $P(t_n = 1|\mathbf{x}_n, \mathbf{w}) = y(\mathbf{x}_n; \mathbf{w})$. Consequently, we would get $P(t_n = 0|\mathbf{x}_n, \mathbf{w}) = 1 - y(\mathbf{x}_n; \mathbf{w})$. Therefore the probability of observed data would be given by

$$\begin{aligned} P(\mathcal{D}|\mathbf{w}) &= \prod_n y_n(\mathbf{w})^{t_n}(1 - y_n(\mathbf{w}))^{1-t_n} \\ &= \exp(-G(\mathbf{w})), \end{aligned} \tag{5}$$

where $y_n(\mathbf{w}) \equiv y(\mathbf{x}_n; \mathbf{w})$ and

$$G(\mathbf{w}) = -\sum_n t_n \log y_n(\mathbf{w}) + (1 - t_n)\log(1 - y_n(\mathbf{w})), \tag{6}$$

which is referred to as the binary-cross entropy loss.

By Bayes' theorem[4]

$$P(\mathbf{w}|\mathcal{D}, \alpha) = \frac{P(\mathcal{D}|\mathbf{w})P(\mathbf{w}|\alpha)}{P(\mathcal{D}|\alpha)}, \tag{7}$$

the posterior distribution over $\mathbf{w}$ can be written as

$$P(\mathbf{w}|\mathcal{D}, \alpha) = \frac{1}{Z_M} \exp(-M(\mathbf{w})), \tag{8}$$

where

$$M(\mathbf{w}) = G(\mathbf{w}) + \alpha E(\mathbf{w}), \tag{9}$$

and Z_M is the normalizing constant given by

$$Z_M = \int \exp(-M(\mathbf{w}))d^k\mathbf{w}. \tag{10}$$

This integral is intractable and we need the value of Z_M to infer α. To get around this problem, we can substitute $M(\mathbf{w})$ by its quadratic Taylor approximation around $\mathbf{w}_{\text{MAP}} = \text{argmin}M(\mathbf{w})$ as

$$M(\mathbf{w}) \simeq M(\mathbf{w}_{\text{MAP}}) + \frac{1}{2}(\Delta\mathbf{w}^T\mathbf{A}\Delta\mathbf{w}), \tag{11}$$

[4]Data's dependence on α is only through $\mathbf{w}$ i.e. $P(\mathcal{D}|\mathbf{w}, \alpha) = P(\mathcal{D}|\mathbf{w})$.

where

$$\Delta \mathbf{w} = \mathbf{w} - \mathbf{w}_{\text{MAP}}, \tag{12}$$

and

$$\mathbf{A} = \mathbf{H}_M(\mathbf{w}_{\text{MAP}}), \tag{13}$$

which is the Hessian matrix of $M(\mathbf{w})$ computed at $\mathbf{w}_{\text{MAP}}$. Consequently, the posterior distribution would be a Gaussian as

$$P(\mathbf{w}|\mathcal{D}, \alpha) = \frac{1}{Z_M} \exp(-\frac{1}{2}\Delta \mathbf{w}^T \mathbf{A} \Delta \mathbf{w}), \tag{14}$$

where

$$Z_M = \mathrm{e}^{-M(\mathbf{w}_{\text{MAP}})} \sqrt{(2\pi)^k / \det(\mathbf{A})}. \tag{15}$$

For a logistic regression model defined by

$$y_n(\mathbf{w}) := \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x}_n)}, \tag{16}$$

it can be verified that

$$\mathbf{A} = \sum_n y_n(\mathbf{w}_{\text{MAP}}) \left[1 - y_n(\mathbf{w}_{\text{MAP}})\right] \mathbf{x}_n \mathbf{x}_n^T + \alpha \mathbf{I}_k. \tag{17}$$

In summary, we approximated our posterior distribution by a Gaussian distribution with mean $\mathbf{w}_{\text{MAP}}$ and covariance matrix $\mathbf{A}^{-1}$.

B. Inference of hyperparameter α

The mean and the covariance matrix of the posterior distribution over $\mathbf{w}$ depend on α. To address this issue, we must take the expectation of $P(\mathbf{w}|\mathcal{D}, \alpha)$ when α's are drawn from their own posterior distribution $P(\alpha|\mathcal{D})$, or simply marginalize $P(\mathbf{w}, \alpha|\mathcal{D})$ over α i.e.

$$P(\mathbf{w}|\mathcal{D}) = \int P(\mathbf{w}|\mathcal{D}, \alpha) P(\alpha|\mathcal{D}) \; d\alpha. \tag{18}$$

The posterior over α is determined by

$$P(\alpha|\mathcal{D}) \propto P(\mathcal{D}|\alpha) P(\alpha). \tag{19}$$

Assuming a uniform prior over $\log \alpha$—since α appeared as an exponent in (2)—posterior belief over α is completely determined by its likelihood function $P(\mathcal{D}|\alpha)$ which is also known as the evidence for α. Evidence $P(\mathcal{D}|\alpha)$ appeared as the normalizing constant in (7) and is thus given by $P(\mathcal{D}|\alpha) = Z_M / Z_E$ which using (4) and (15) results in

$$\log P(\mathcal{D}|\alpha) = -M(\mathbf{w}_{\text{MAP}}) - \frac{1}{2} \log\det \mathbf{A} + \frac{k}{2} \log \alpha. \tag{20}$$

In the above equation, $M(\mathbf{w}_{\text{MAP}})$ and $\mathbf{A}$ depend on α according to (9) and (17). If we rewrite (17) as $\mathbf{A} = \mathbf{B} + \alpha \mathbf{I}_k$ and plug that together with (9) into (20), we get

$$\begin{aligned} \log P(\mathcal{D}|\alpha) = &- G(\mathbf{w}_{\text{MAP}}) - \alpha E(\mathbf{w}_{\text{MAP}}) \\ &- \frac{1}{2} \log\det(\mathbf{B} + \alpha \mathbf{I}_k) + \frac{k}{2} \log \alpha. \end{aligned} \tag{21}$$

We apply the same Gaussian approximation technique in the previous subsection to $P(\alpha|\mathcal{D})$, with the difference that we replace the Gaussian distribution by a delta function at its peak and keep track of the variance only as a measure of how dominant the peak is.

If $P(\alpha|\mathcal{D})$ has a dominant peak[5] at α_{MAP}, we can approximate (18) by

$$P(\mathbf{w}|\mathcal{D}) \simeq P(\mathbf{w}|\mathcal{D}, \alpha_{\text{MAP}}). \tag{22}$$

If we take the derivative of (21) w.r.t $\log \alpha$ and set it to zero, we find α_{MAP} will satisfy

$$2\alpha_{\text{MAP}} E(\mathbf{w}_{\text{MAP}}) + \alpha_{\text{MAP}} \text{Trace}((\mathbf{B} + \alpha_{\text{MAP}} \mathbf{I}_k)^{-1}) - k = 0. \tag{23}$$

We will solve this equation numerically to find α_{MAP}. Taking the second derivative of (21) w.r.t $(\log \alpha)^2$ computed at α_{MAP} shows how dominant the peak is and is given by [9]

$$\sigma_{\log \alpha|\mathcal{D}} \simeq \sqrt{\frac{2}{k - \alpha \text{Trace}(\mathbf{A}^{-1})}}. \tag{24}$$

With few labeled samples, $\alpha \text{Trace}(A^{-1})$ would be close to its maximum value k, which in turn leads to a large $\sigma_{\log \alpha|\mathcal{D}}$, so the peak would not be dominant and approximation (22) would not hold.

To conclude this section, our posterior distribution over model parameters, which also represents our uncertainty of parameters, is approximated to $P(\mathbf{w}|\mathcal{D}, \alpha_{\text{MAP}})$. Details on how we actually compute the posterior will be given in Section VI.

IV. Bayesian Active Learning

Having gathered N input-target pairs, we would like to select the next input $\mathbf{x}_{N+1}$ such that we expect maximal information gain on model parameters once we receive target t_{N+1}. We will introduce two measures of information gain, both of which depend on entropy.

Entropy was originally introduced by Shannon [10] for discrete random variables. It measures how uncertain a discrete random variable is. For example, if we have a bent coin with bias p, the entropy is zero when $p = 0$ or $p = 1$ and is maximum when $p = 0.5$. For continuous random variables, entropy on its own is incompetent of conveying anything meaningful [11], mainly due to the fact that it is scale variant. However, change in entropy can be one measure of information gain (or loss).

Let us denote the probability distributions of parameters before and after we receive the target t_{N+1} by $P_N(\mathbf{w})$ and $P_{N+1}(\mathbf{w})$, respectively. Then information gain would mean a positive $\Delta S = S_N - S_{N+1}$, where

$$S_N = \int P_N(\mathbf{w}) \log \frac{1}{P_N(\mathbf{w})} \; d^k \mathbf{w} \tag{25}$$

is the entropy of the probability distribution of parameters before receiving t_{N+1}. Since the value of t_{N+1} is unknown to us when selecting $\mathbf{x}_{N+1}$, we will be working with the expectation over $P(t|\mathbf{x}, \mathcal{D})$ of our selected information gain measure. Another measure for information gain is the cross entropy between $P_N(\mathbf{w})$ and $P_{N+1}(\mathbf{w})$ defined as

$$C = \int P_{N+1}(\mathbf{w}) \log \frac{P_N(\mathbf{w})}{P_{N+1}(\mathbf{w})} \; d^k \mathbf{w}. \tag{26}$$

[5]Empirically, if the model is well-matched to the data, this distribution is unimodal [7].

It is shown in [4] that change in entropy and cross entropy are equivalent in expectation i.e. $\mathbb{E}[\Delta S] = \mathbb{E}[C]$. Merely out of convenience, we will be working with change in entropy moving forward.

If we denote our entire training set by Q, from which N samples have been labeled, then our next query to label would be

$$\mathbf{x}_{N+1} = \underset{\mathbf{x}\in Q}{\operatorname{argmax}} \left(\mathbb{E}_{P(t|\mathbf{x},\mathcal{D})} \left[S_N - S_{N+1} \right] \right). \tag{27}$$

We approximated our posterior distribution over $\mathbf{w}$ by a Gaussian in (14). It can be shown that the entropy of a k-dimensional Gaussian distribution with covariance matrix $\mathbf{A}^{-1}$ is [4]

$$S = \frac{k}{2}(1 + \log 2\pi) + \frac{1}{2}\log(\det \mathbf{A}^{-1}). \tag{28}$$

Therefore change in entropy would equal to

$$\Delta S = \frac{1}{2}\log \frac{\det \mathbf{A}_{N+1}}{\det \mathbf{A}_N}. \tag{29}$$

Due to (17), the relationship between $\mathbf{A}_{N+1}$ and $\mathbf{A}_N$ is

$$\mathbf{A}_{N+1} = \mathbf{A}_N + y_{N+1}(\mathbf{w}_{\text{MAP}}) \left[1 - y_{N+1}(\mathbf{w}_{\text{MAP}}) \right] \mathbf{x}_{N+1}\mathbf{x}_{N+1}^T. \tag{30}$$

For a scalar β and a vector $\mathbf{x}$, determinant has the property $\det \left[\mathbf{A} + \beta \mathbf{x}\mathbf{x}^T \right] = (\det \mathbf{A})(1 + \beta \mathbf{x}^T \mathbf{A}^{-1}\mathbf{x})$. Applying this property to (30), we can rewrite (29) as

$$\Delta S = \frac{1}{2}\log(1 + m), \tag{31}$$

where

$$m = y_{N+1}(\mathbf{w}_{\text{MAP}}) \left[1 - y_{N+1}(\mathbf{w}_{\text{MAP}}) \right] \mathbf{x}_{N+1}^T \mathbf{A}_N^{-1} \mathbf{x}_{N+1}. \tag{32}$$

Note that this expression is independent of t_{N+1}, and as a result $\mathbb{E}(\Delta S) = \Delta S$. This is a mere consequent of choosing a logistic regression model, and might not hold for other models.

We refer to the criterion defined by (31) and (32), as maximal expected information gain or just information gain for short. This criterion has a nice property that $y_{N+1}(\mathbf{w}_{\text{MAP}}) \left(1 - y_{N+1}(\mathbf{w}_{\text{MAP}}) \right)$ favors the points close to the decision boundary while $\mathbf{x}_{N+1}^T \mathbf{A}_N^{-1} \mathbf{x}_{N+1}$ favors the points on the far end of data space. In particular, for small N, where we have a high uncertainty over parameters, $\mathbf{x}_{N+1}^T \mathbf{A}_N^{-1} \mathbf{x}_{N+1}$ would be the dominating term, so the criterion would select points which have a larger norm and lie close to the decision boundary. As we gather more data, eigenvalues of the covariance $\mathbf{A}_N^{-1}$ would get smaller, and $y_{N+1}(\mathbf{w}_{\text{MAP}}) \left(1 - y_{N+1}(\mathbf{w}_{\text{MAP}}) \right)$ would be the dominating term in the criterion. At this point, maximal expected information gain behaves close to decision boundary sampling in [6]. Figure 1 illustrates this intuition of the two sampling strategies for $N = 2, 7, 12, 17$ and 22.

Lastly, information gain criterion is closely related to maximum model change criterion in [5] for a logistic regression model. Inspired by the gradient descent update rule, which updates parameters in the opposite direction of the gradient of the loss function, the authors in [5] propose a sampling strategy which maximizes the expected gradient length of the loss function.

V. Bayesian prediction

By the application of probability rules, Bayesian prediction takes the uncertainty of parameters into account when predicting new targets. For an input $\mathbf{x}$, our prediction for its corresponding target t to belong to the positive class is the expected value of our model output when the parameters are drawn from the posterior distribution i.e.

$$P(t = 1|\mathbf{x}, \mathcal{D}) = \int P(t = 1|\mathbf{x}, \mathbf{w}) P(\mathbf{w}|\mathcal{D}) \; d^k\mathbf{w}. \tag{33}$$

The term $P(t = 1|\mathbf{x}, \mathcal{D})$ is written in shorthand by $y(\mathbf{x})$ and is referred to as marginalized output.

Under our current assumptions $P(\mathbf{w}|\mathcal{D})$ is a Gaussian distribution according to (14), and $P(t = 1|\mathbf{x}, \mathbf{w})$ is a sigmoidal function according to (16), which render the above integral intractable. Maximum a Posteriori (MAP) method estimates this integral by

$$P(t = 1|\mathbf{x}, \mathcal{D}) \simeq P(t = 1|\mathbf{x}, \mathbf{w}_{\text{MAP}}), \tag{34}$$

which is equivalent of replacing the posterior $P(\mathbf{w}|\mathcal{D})$ by a delta function at its peak $\mathbf{w}_{\text{MAP}}$ (similar to what we did for the inference of α). A better approximation has been in suggested [12] resulting in a predictive distribution as [6]

$$P(t = 1|\mathbf{x}, \mathcal{D}) = \frac{1}{1 + \exp(-\mathbf{w}_{\text{MAP}}^T \mathbf{x} / \sqrt{1 + \frac{\pi}{8}\mathbf{x}^T \mathbf{A}^{-1}\mathbf{x}})}. \tag{35}$$

This equation comes with a nice interpretation: When the uncertainty of parameters are low i.e. the eigenvalues of the covariance $\mathbf{A}^{-1}$ are small, we get $\mathbf{x}^T \mathbf{A}^{-1}\mathbf{x} \to 0$, which implies $P(t = 1|\mathbf{x}, \mathcal{D}) \to P(t = 1|\mathbf{x}, \mathbf{w}_{\text{MAP}})$, meaning MAP estimation is valid. On the other hand, when the eigenvalues (uncertainties) are large, giving $\mathbf{x}^T \mathbf{A}^{-1}\mathbf{x} \to \infty$, we see that $P(t = 1|\mathbf{x}, \mathcal{D}) \to 0.5$, meaning that we have low confidence in our predictions due to high uncertainty in parameters. MAP estimation is no longer valid in this case. In conclusion, the term $\mathbf{x}^T \mathbf{A}^{-1}\mathbf{x}$ prevents overconfident predictions when the model has been exposed to little data and is uncertain about the parameters as a result.

Having settled our prediction, we would seek to define measures for out-of-sample performance on a given test set $\{\mathbf{x}_m, t_m | m = 1, \cdots, M\}$. Accuracy defined by (37) is ignorant of prediction uncertainty and is not preferred as a result.

$$\text{accuracy} = \frac{1}{M}\sum_m \mathbb{1}(t_m = \lfloor y_m \rceil), \tag{36}$$

where $\mathbb{1}(\cdot)$ outputs one if its argument is True and zero if False, $\lfloor \cdot \rceil$ rounds its argument to closest integer, and $y_m \equiv y(\mathbf{x}_m)$.

[6] This approximation becomes inaccurate when $\mathbf{w}_{\text{MAP}}^T \mathbf{x} \gg \mathbf{x}^T \mathbf{A}^{-1}\mathbf{x} \gg 1$.

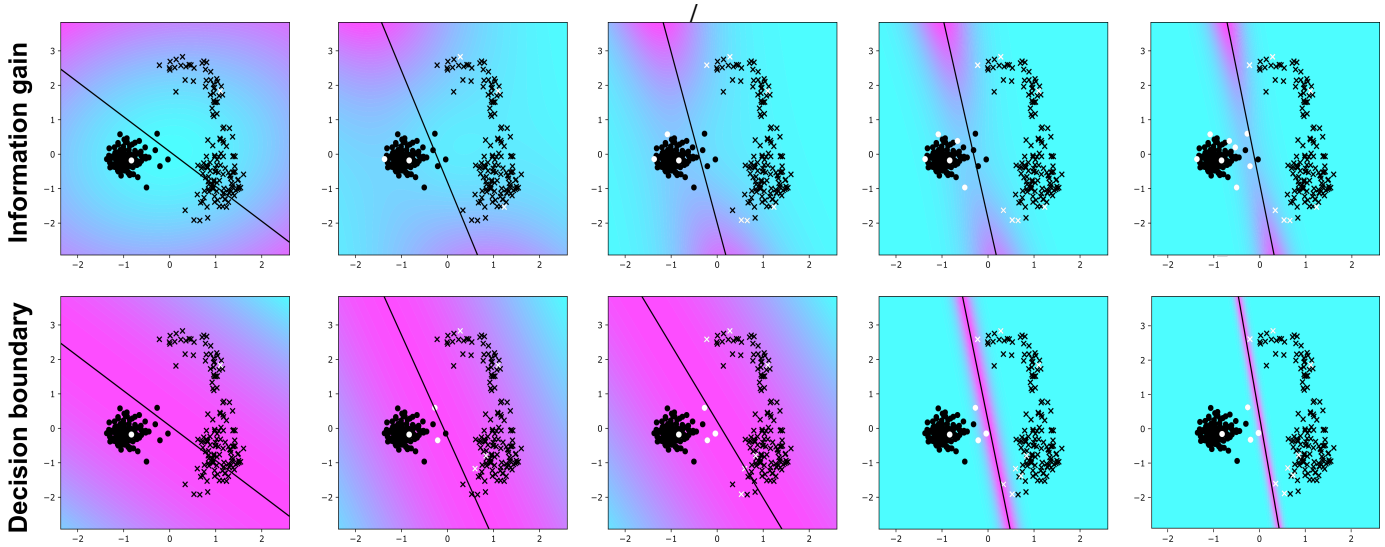

Fig. 1. Here, we show the sampling procedure for maximal expected information gain versus decision boundary sampling for a 2D version of the Digits data set obtained by projecting it onto its first two principal components. We show the points drawn at iteration 0, 5, 10, 15, and 20, respectively. The black dots are unlabeled samples and the white ones are labeled samples. The color indicates the value (pink=high, blue=low) of the property being maximized, i.e. the information gain (top) and the distance to decision boundary (bottom). Note how samples are drawn near pink areas.

An alternative would be the binary cross entropy loss between the targets t_m and marginalized outputs y_m, which we call marginalized loss, and is given by

$$\text{marginalized loss} = -\frac{1}{M}\sum_m t_m \log y_m + (1-t_m)\log(1-y_m). \tag{37}$$

Due to the intuition given for marginalized output, a model which makes a mistake in the classification of a test point but is unsure about its prediction incurs a lower marginalized loss, than the one which makes a mistake and is confident about its prediction.

With marginalized loss as our preferred measure of performance on unseen data, we head to experiment.

VI. Experiments

We compare our maximal information gain strategy with two other sampling strategies using logistic regression; namely random sampling and decision boundary sampling [6], where the latter picks the closest possible sample to the decision boundary under the assumption that these are maximally uncertain. We perform experiments on the following 10 classification data sets: AB and ABA are synthetic data sets sampled from 2 and 3 2-dimensional Gaussian distributions, respectively, to produce data sets which are linearly separable (AB) and not linearly separable (ABA). The Breast Cancer Wisconsin (Diagnostic) [13], Optical Recognition of Handwritten Digits [14], Statlog (Heart), Haberman's Survival [15], Parkinson's [16], and Ionosphere [17] data sets come from the UCI repository [18]; here, for Digits only the first two classes were used. DD [19], [20] and AIDS [21], [22] are benchmark graph learning data sets [23], where each graph was given a vector representation via its node degree histogram. The details of the 10 data sets are shown in Table I.

TABLE I
Data set details

Name	AB	Heart	Digits	AIDS	Cancer
# Samples	1000	270	360	2000	569
# Features	2	13	64	10	30

Name	Parkinson's	DD	Haberman	Ion	ABA
# Samples	195	1178	306	351	1000
# Features	22	19	3	34	2

When training logistic regression, we pick the initial value of $\log\alpha$ according to its prior, $\mathcal{U}(10^{-3}, 10)$ in our case, each time a new training point is drawn. With this value of α, we minimize $M(\mathbf{w})$, and then update α according to (23). We plug in this new α into the equation for $M(\mathbf{w})$ and repeat this process until convergence, which is guaranteed if optimizations of $M(\mathbf{w})$ at each stage are not far off [9]. The final values for $\mathbf{w}_{\text{MAP}}$ and $\mathbf{A}$ will be used for the prediction.

When using the maximal expected information gain sampling, we start the experiments by randomly revealing one labeled data point from each class, and add in one sample at a time that maximizes (31). We repeat this setup 20 times changing the two revealed data points each time. A similar procedure is applied to the decision boundary sampling and random sampling strategies; these are initialized with the same two data points per run as the maximal information gain sampling. Samples are selected with replacement among all strategies i.e. relabeling of the same input is possible. Otherwise, all sampling strategies would reach the same performance when all samples in Q are exhausted [1].

Figure 2 and Figure 3 show the mean and standard deviation of the accuracy and marginalized loss over the different data

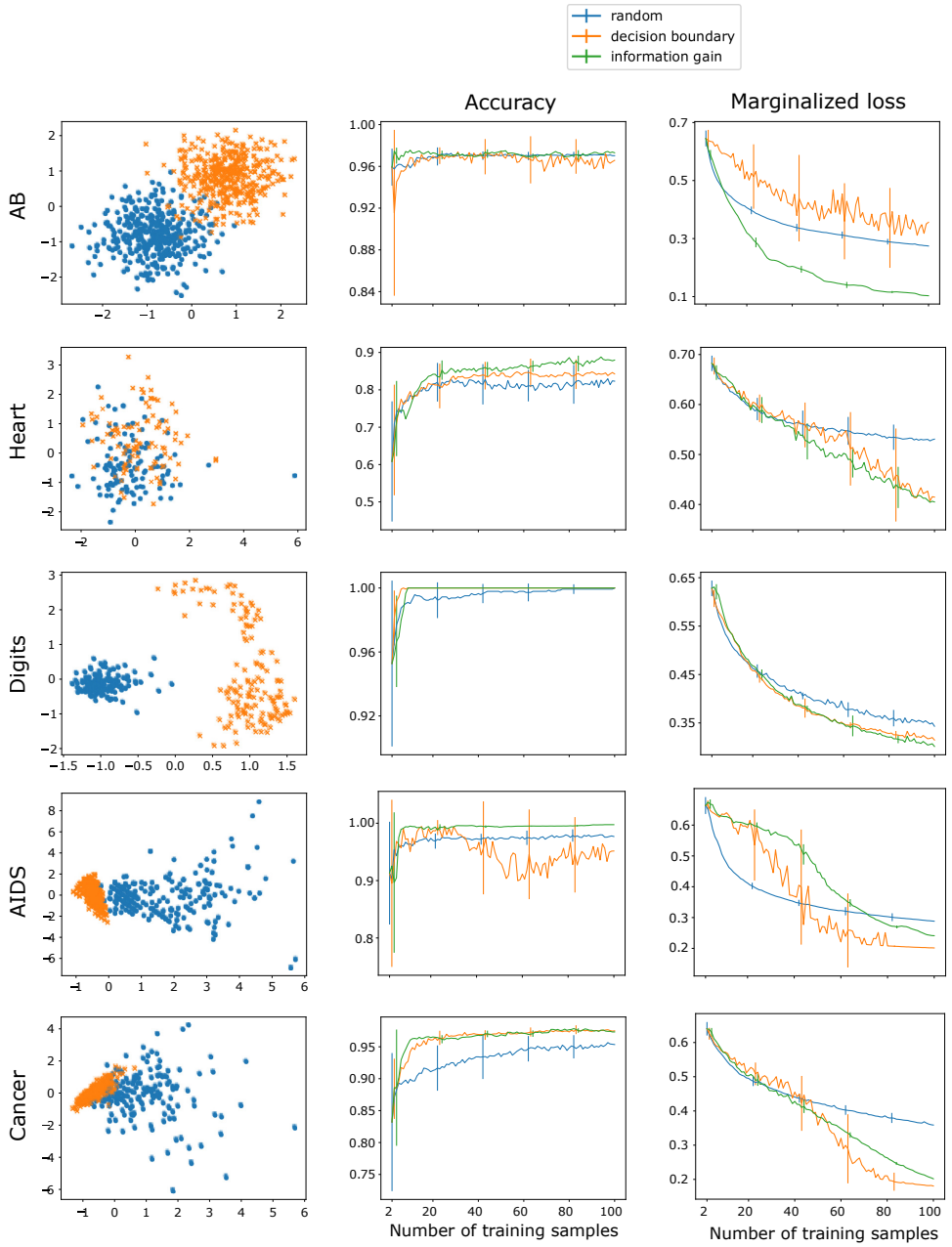

Fig. 2. For 5 roughly linearly separable data sets, we see (left) a visualization of the data set via projection onto the first two principal components, as well as the (middle) accuracy and (right) marginalized loss, both as a function of number of training samples seen. The plots contain the mean and standard deviations of 20 repeated runs of the sampling strategies. As the logistic regression model is well-matched to these data sets, the information gain criterion for gathering samples outperforms random sampling.

sets for the three sampling strategies. The figures also show (left column) the projection of the data onto the first two principal components of each data set, to give insight into data set properties.

Data sets in Figure 2 compose of two clusters for each class which are roughly separable by a line. Even Heart data set looks like two Gaussians whose means are close to each other. Looking at their corresponding accuracies and marginalized loss, we see that at least in the limit of high amount of data, information gain outperforms random sampling consistently. When the number of training samples is small, in the cases where we obtain inferior performance compared to random sampling, we speculate the reason to be a large $\sigma_{\log \alpha|\mathcal{D}}$ given in (24); not to mention that approximation (35) breaks when $\mathbf{w}_{\text{MAP}}^T \mathbf{x} \gg \mathbf{x}^T \mathbf{A}^{-1} \mathbf{x} \gg 1$. It is visually visible that for data sets in Figure 3, however, the classes are less linearly separable to the point that ABA clearly is not. In such scenarios, information gain sampling does not perform satisfactorily in neither accuracy nor marginalized loss. The reason is that the logistic regression model is not well-matched to these data

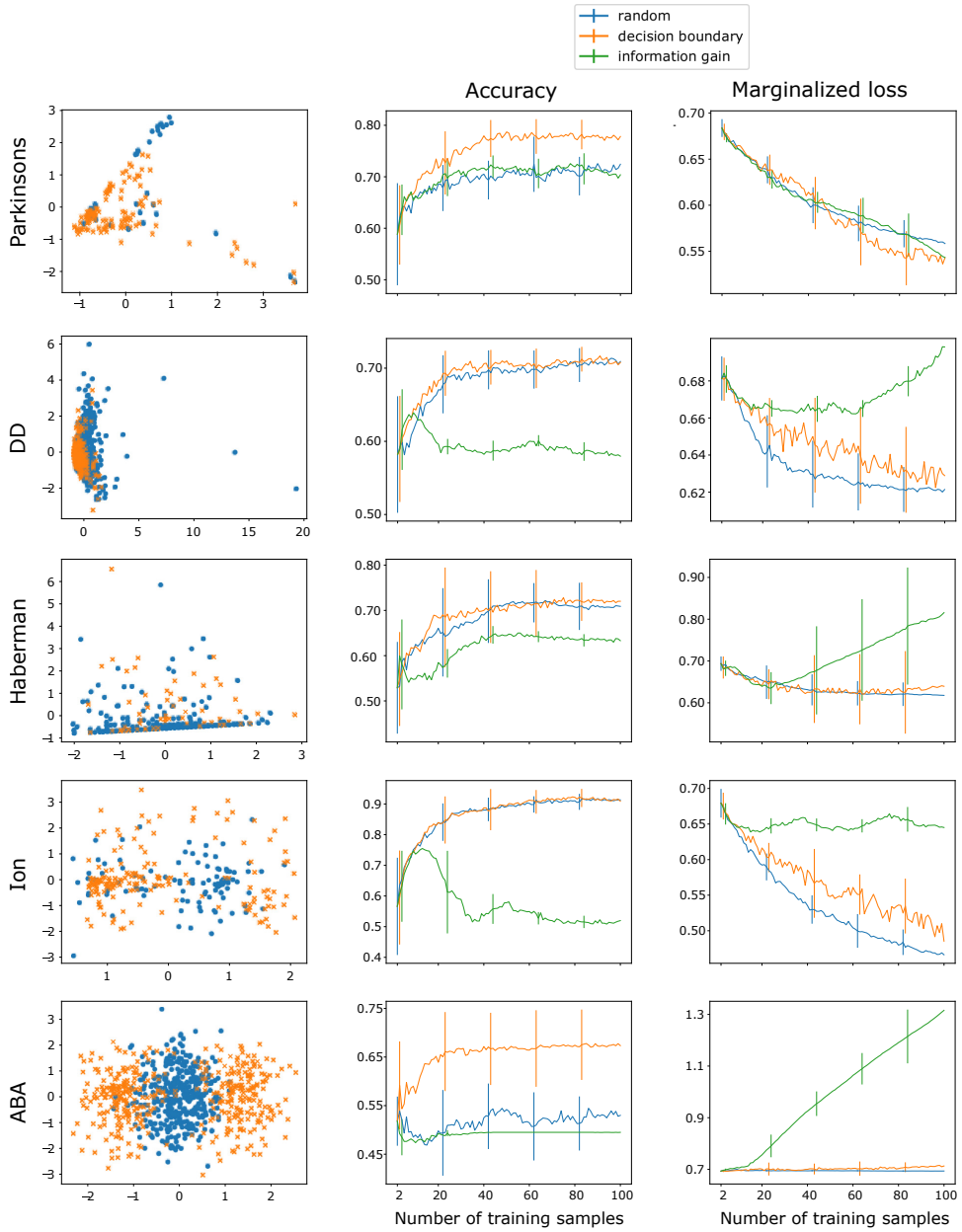

Fig. 3. For 5 not linearly separable data sets, we see (left) a visualization of the data set via projection onto the first two principal components, as well as the (middle) accuracy and (right) marginalized loss, both as a function of number of training samples seen. The plots contain the mean and standard deviations of 20 repeated runs of the sampling strategies. Since the assumption that our hypothesis space is correct no longer holds, the information gain criterion performs poorly.

sets. Information gain criterion works by taking uncertainty of parameters into account, but the input-target relationship in these data sets cannot be modeled by a logistic regression, so the covariance matrix A^{-1} we compute in (17) is far from anything meaningful. This is verified by the increasing marginalized loss in DD, Haberman, and ABA. We will resume this discussion in the next section.

VII. Discussion and Conclusion

We have derived an active learning sampling scheme for classification based on maximizing information gain on model parameters. Additionally, we have shown that in the case of logistic regression, this scheme takes a nice, interpretable form and, in particular, is closely related to the more ad hoc maximum model change [5]. Via our experiments we also see how the performance of the method depends whether our choice of classifier is well matched to the data set, an assumption implicitly underlying our analysis.

In particular, in Section II, we assumed that we have found the so-called true model, and as a result the purpose of data gathering is to infer the plausibility of model parameters with smallest possible amount of labeled data. Subsequently, we

derived a criterion under the Bayesian framework assuming that the model is well-matched to the data. In Bayesian language, our assumption is that the hypothesis space is correct. Our experiments confirmed that indeed, when the hypothesis space is correct, the information gain criterion could be a promising sampling strategy—perhaps with more accurate approximations using Monte Carlo methods [24]. That is reassuring except the fact that we have no Bayesian way of verifying that our hypothesis space actually is correct [25].

Hypothesis testing in the Bayesian framework is performed through model-comparison. Let's say we have several hypotheses $\mathcal{H}_i$ whose validity we want to investigate, and we have gathered data $\mathcal{D}$. Thanks to Bayes' theorem

$$P(\mathcal{H}_i|\mathcal{D}) \propto P(\mathcal{D}|\mathcal{H}_i)P(\mathcal{H}_i), \tag{38}$$

we can sort alternative hypotheses in order of their plausibility. They could all be completely far from the truth and we would get a ranking regardless. Therefore, Bayesian model comparison is also only viable if we are within the correct hypothesis space, where Bayes wouldn't prefer other hypotheses to the truth [7].

It seems inevitable that one must err on the side of a larger hypothesis space. Logistic regression is essentially a classification neural network with no hidden layer, which has the nice property that its posterior over parameters with a Gaussian prior is unimodal. For a neural network with hidden layers, the posterior could be multimodal, and even if we can find the global maximum of the posterior, fitting a Gaussian around that point is not an acceptable substitute for the entire posterior distribution. However, a solution has been given in [26] which is to fit a Gaussian distribution around each local maximum of the posterior and treat each maximum separately. One can then use Bayes' theorem to compare them to one another. Although this might not be a permanent solution, since neural networks of a fixed width and depth are not still universal approximators, they nevertheless specify a larger hypothesis space than logistic regression. Extending our analysis to more flexible, modern classifiers remains an important avenue for future research.

ACKNOWLEDGMENT

This work was supported by the Novo Nordisk Foundation grant NNF17OC0028360.

REFERENCES

[1] M. Loog and Y. Yang, "An empirical investigation into the inconsistency of sequential active learning," in *2016 23rd international conference on pattern recognition (ICPR)*. IEEE, 2016, pp. 210–215.

[2] Y. Yang and M. Loog, "A benchmark and comparison of active learning for logistic regression," *Pattern Recognition*, vol. 83, pp. 401–415, 2018.

[3] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.

[4] D. J. MacKay, "Information-based objective functions for active data selection," *Neural computation*, vol. 4, no. 4, pp. 590–604, 1992.

[5] W. Cai, Y. Zhang, Y. Zhang, S. Zhou, W. Wang, Z. Chen, and C. Ding, "Active learning for classification with maximum model change," *ACM Transactions on Information Systems (TOIS)*, vol. 36, no. 2, pp. 1–28, 2017.

[6] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *SIGIR'94*. Springer, 1994, pp. 3–12.

[7] D. J. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.

[8] G. E. Box and G. C. Tiao, *Bayesian inference in statistical analysis*. John Wiley & Sons, 2011, vol. 40.

[9] D. J. MacKay, "Comparison of approximate methods for handling hyperparameters," *Neural computation*, vol. 11, no. 5, pp. 1035–1068, 1999.

[10] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.

[11] C. Marsh, "Introduction to continuous entropy," *Department of Computer Science, Princeton University*, 2013.

[12] D. J. MacKay, "The evidence framework applied to classification networks," *Neural computation*, vol. 4, no. 5, pp. 720–736, 1992.

[13] O. L. Mangasarian and W. H. Wolberg, "Cancer diagnosis via linear programming," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 1990.

[14] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE transactions on systems, man, and cybernetics*, vol. 22, no. 3, pp. 418–435, 1992.

[15] S. J. Haberman, "Generalized residuals for log-linear models," in *Proceedings of the 9th international biometrics conference*, 1976, pp. 104–122.

[16] M. A. Little, P. E. McSharry, S. J. Roberts, D. A. Costello, and I. M. Moroz, "Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection," *Biomedical engineering online*, vol. 6, no. 1, p. 23, 2007.

[17] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker, "Classification of radar returns from the ionosphere using neural networks," *Johns Hopkins APL Technical Digest*, vol. 10, no. 3, pp. 262–266, 1989.

[18] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[19] P. D. Dobson and A. J. Doig, "Distinguishing enzyme structures from non-enzymes without alignments," *Journal of molecular biology*, vol. 330, no. 4, pp. 771–783, 2003.

[20] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels." *Journal of Machine Learning Research*, vol. 12, no. 9, 2011.

[21] K. Riesen and H. Bunke, "Iam graph database repository for graph based pattern recognition and machine learning," in *Structural, Syntactic, and Statistical Pattern Recognition*, N. da Vitoria Lobo, T. Kasparis, F. Roli, J. T. Kwok, M. Georgiopoulos, G. C. Anagnostopoulos, and M. Loog, Eds., 2008, pp. 287–297.

[22] D. Zaharevitz, "Aids antiviral screen data," 2015.

[23] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "Tudataset: A collection of benchmark datasets for learning with graphs," in *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*. [Online]. Available: www.graphlearning.io

[24] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[25] D. J. MacKay, "Bayesian methods for adaptive models," Ph.D. dissertation, California Institute of Technology, 1992.

[26] ——, "A practical bayesian framework for backpropagation networks," *Neural computation*, vol. 4, no. 3, pp. 448–472, 1992.