Active Learning with Weak Annotations for Gaussian Processes

Amanda Olmin  
amanda.olmin@liu.se  
Linköping University

Jakob Lindqvist  
jakob.lindqvist@chalmers.se  
Chalmers University of Technology

Lennart Svensson  
lassen.svensson@chalmers.se  
Chalmers University of Technology

Fredrik Lindsten  
fredrik.lindsten@liu.se  
Linköping University

Abstract

Annotating data for supervised learning can be costly. When the annotation budget is limited, active learning can be used to select and annotate those observations that are likely to give the most gain in model performance. We propose an active learning algorithm that, in addition to selecting which observation to annotate, selects the precision of the annotation that is acquired. Assuming that annotations with low precision are cheaper to obtain, this allows the model to explore a larger part of the input space, with the same annotation costs. We build our acquisition function on the previously proposed BALD objective for Gaussian Processes, and empirically demonstrate the gains of being able to adjust the annotation precision in the active learning loop.

1 Introduction

Supervised learning requires annotated data and sometimes a vast amount of it. In situations where input data is abundant or the input distribution known, but annotations are costly, we can use the annotation budget wisely and optimise model performance by selecting and annotating those instances that are most useful for the model. This is often referred to as active learning. Specifically, when the instances to annotate are (iteratively) selected from a larger pool of unannotated observations, it is referred to as pool-based active learning (Settles, 2010).

A common goal of active learning is to achieve the best possible model performance given a fixed annotation budget. As mentioned, the strategy used to accomplish this is to carefully select which instances to annotate and add to the training set. However, while the focus of most active learning algorithms lies on this type of instance selection, there might be other factors that can be controlled in order to use the annotation budget to its fullest.

In this paper, we consider the setting in which it is possible to control the precision of acquired annotations. In the proposed setting, we can include the annotation
precision in the selection strategy of the active learning algorithm. Under such circumstances, gains can be made in model performance by affording to collect several noisy, in place of a few precise, annotations. This is based on the assumption that noisy annotations are cheaper to acquire than precise annotations, but could still provide useful information to the model.

In some contexts, the task of jointly selecting the data point to annotate and its precision could be reduced to selecting the most appropriate annotator, considering both annotator accuracy as well as cost. This is related to a wide range of previous work on annotator modelling, starting with early work on modelling annotator accuracy (Dawid and Skene, 1979) to weighting multiple labels based on annotator expertise (Guan et al., 2018). Most closely related to the contributions of this paper, in this regard, is previous work on simultaneously selecting instances and annotators in an active learning setting; mainly those that specifically consider annotator costs, e.g. (Huang et al., 2017; Gao and Saar-Tsechansky, 2020; Chakraborty, 2020), but also those that do not, e.g. (Yan et al., 2011; Herde et al., 2020).

Although annotator selection is a special case of the setting that we consider, we emphasise that the setting is more general than that. It also includes, for instance, situations where annotations are collected experimentally and precision can be controlled in experimental measurements or calculations. Moreover, we consider applying active learning in regression, which, to the best of our knowledge, is relatively under-explored in the field.

In this paper, we propose an extended active learning algorithm, applicable to the setting just described, where, in addition to instance selection, it is possible to select the precision of an annotation. We build the proposed acquisition strategy on the Bayesian Active Learning by Disagreement (BALD, Houlsby et al. (2011)) objective, that relies on the mutual information between the target variable and the model. The mutual information is a measure of how much information one variable carries about the other (Cover and Thomas, 2006). Hence, we can interpret the BALD acquisition function as conveying which target values will give the most information about the model. We use an iterative active learning approach, where we propose to optimise mutual information per annotation cost in each iteration of the active learning algorithm.

To conclude, the contribution of this paper is an extension of the traditional iterative active learning algorithm, in which we include the possibility to select the precision of annotations. By allowing to adjust the annotation precision, the proposed approach can boost model performance under budget constraints. We develop our active learning algorithm using Gaussian Processes, but it can, in many cases, be easily extended to other types of models.

2 Background

In the following section, we describe the main building blocks of the proposed active learning strategy. First, we introduce the general active learning setting considered in the paper and explain the BALD objective, on which the proposed strategy rely, in more detail. Second, we give a brief introduction to Gaussian Processes.
2.1 Active learning

Suppose that we want to learn a probabilistic predictive model, $f$, predicting the distribution of a target variable $Y \in \mathcal{Y}$ given the input variable $X \in \mathcal{X}$. In a pool-based active learning setting, it is assumed that we have a set of observations of the input variable, $X$, available. In the context of this paper, however, we could in principle know the full input space, $\mathcal{X}$. Regardless, we will represent this by a set of $N$ observations, denoted by $X_{1:N}^{(pool)}$.

For the purpose of using supervised learning, we need to collect corresponding target values for the observations in the unannotated set. However, our annotation budget, $B$, is limited and we can therefore only afford to annotate a subset of these instances. In the basic setting, we assume that the budget is defined as the number of annotations that we can afford to acquire, and that all annotations have equal cost, independent of $X$.

To select which instances to acquire annotations for, we consider iterative active learning, where the following steps are repeated until the budget is exhausted

1. Fit the model, $f$, to the current set of annotated data.
2. Select the next instance to annotate, acquire the annotation and add the new data pair to the training data.

Typically, it is assumed that this iterative active learning procedure starts with an initial set of $n$ (randomly sampled) annotated observations, or training instances, $X_{1:n}^{(train)}$. We denote the corresponding target values by $Y_{1:n}^{(train)}$.

For the second step in the iterative active learning algorithm, we define an acquisition function, $\phi$, that conveys our intention of selecting and annotating those instances in $X_{1:N}^{(pool)}$ that we believe will lead to the best model performance. Hence, to select which instance, $X^{(a)}$, to annotate, we solve the following optimisation problem

$$X^{(a)} = \arg\max_{X \in X_{1:N}^{(pool)}} \phi(X). \tag{1}$$

Note that the acquisition function commonly depends on the current model, $f$, and/or the training instances selected so far. We show the full active learning procedure in algorithm 1.

2.1.1 Bayesian active learning by disagreement

Bayesian Active Learning by Disagreement (BALD, Houlsby et al. (2011)), is an acquisition function based on mutual information. More specifically, BALD is defined as the mutual information between the target variable, $Y$, and the model, $f$, conditioned on the input as well as the training data

$$\text{MI}(Y; f \mid X, X_{1:n}^{(train)}, Y_{1:n}^{(train)}) = \mathcal{H}[Y \mid X, X_{1:n}^{(train)}, Y_{1:n}^{(train)}]$$

$$- \mathbb{E}_{f} \left[ \mathcal{H} \left[ Y \mid X_{1:n}^{(train)}, X_{1:n}^{(train)} \right] \mid \mathcal{H}[Y \mid X, f] \right]. \tag{2}$$
Algorithm 1 Active Learning

Require: \( B, X^{(\text{train})}_1 :_n, Y^{(\text{train})}_1 :_n, X^{(\text{pool})}_1 :_N \)

\( b := 0 \)

Initialise model

\[
\text{while } b < B \text{ do}
\]

\[
\text{model.fit}(X^{(\text{train})}_1 :_n, Y^{(\text{train})}_1 :_n) \quad \% \text{Fit model to training data}
\]

\[
X^{(a)} := \arg\max_{X \in X^{(\text{pool})}_1 :_N} \phi(X) \quad \% \text{Select a data point for annotation}
\]

Acquire annotation \( Y^{(a)} \) for \( X^{(a)} \)

\[
X^{(\text{train})}_1 :_n := X^{(\text{train})}_1 :_n \cup X^{(a)} \quad Y^{(\text{train})}_1 :_n := Y^{(\text{train})}_1 :_n \cup Y^{(a)} \quad \% \text{Add } X^{(a)}, Y^{(a)} \text{ to training data}
\]

\[
X^{(\text{pool})}_1 :_N := X^{(\text{pool})}_1 :_N \setminus X^{(a)} \quad \% \text{Remove } X^{(a)} \text{ from the data pool}
\]

\[
b = b + 1 \quad \% \text{Account for annotation costs}
\]

end while

Here, \( \mathcal{H}[\cdot] \) refers to the entropy of a random variable. For a continuous random variable, \( Y \in \mathcal{Y} \), we use differential entropy defined as

\[
\mathcal{H}[Y] = \int_{\mathcal{Y}} g(y) \log g(y) dy,
\]

(3)

with \( g(y) \) the probability density function of \( Y \). For example, the differential entropy of a Gaussian random variable, \( Y \sim \mathcal{N}(\mu, \Sigma) \), is

\[
\mathcal{H}[Y] = - \int_{\mathcal{Y}} \mathcal{N}(y; \mu, \Sigma) \log \mathcal{N}(y; \mu, \Sigma) dy = \frac{1}{2} |d(1 + \log(2\pi)) + \log|\Sigma||
\]

\[
\propto \log|\Sigma| + \text{const.}
\]

(4)

where \(| \cdot |\) denotes the determinant and \( d \) is the variable dimension. In the context of the general active learning algorithm, to use the BALD objective in the iterative active learning loop, we set \( \phi(X) = \text{MI}(Y; f \mid X, X^{(\text{train})}_1 :_n, Y^{(\text{train})}_1 :_n) \).

2.2 Gaussian Processes

A Gaussian Process (GP) is a probabilistic, non-parametric model, postulating a Gaussian distribution over functions, \( f : X \rightarrow \mathbb{R} \) (see e.g. [Bishop (2006)]. Although GPs can be extended to the multivariate setting, we will introduce them for the case in which \( \mathcal{Y} \subseteq \mathbb{R} \). Then, the Gaussian Process prior is defined by

\[
f \sim \text{GP}(m(\cdot), K(\cdot, \cdot)),
\]

\[
Y \mid f, X \sim \mathcal{N}(f(X), \sigma^2),
\]

(5)
Then, for a set of observations $X$ where $m(\cdot)$ is 0. Then, for a set of observations $X^{(\text{train})}$, the prior over the function values $f^{(\text{train})} = f(X_{1:n}^{(\text{train})})$ is written as

$$f^{(\text{train})} \mid X_{1:n}^{(\text{train})} \sim \mathcal{N}(0_n, K(X_{1:n}^{(\text{train})}, X_{1:n}^{(\text{train})})),$$

where $0_n$ is the vector of zeroes of length $n$ and $K(X_{1:n}^{(\text{train})}, X_{1:n}^{(\text{train})})$ is a $n \times n$ matrix with elements $K_{i,j} = K(X_i^{(\text{train})}, X_j^{(\text{train})})$. Following this, we have

$$Y_{1:n}^{(\text{train})} \mid f_{1:n}^{(\text{train})} \sim \mathcal{N}(f_{1:n}^{(\text{train})}, \sigma^2 I_n),$$

with $I_n$ the identity matrix of size $n$.

Based on the introduced probabilistic model, we can derive the predictive distribution of a new observation $X$ to find

$$Y \mid X, X_{1:n}^{(\text{train})}, Y_{1:n}^{(\text{train})} \sim \mathcal{N}(\mu, \sigma^2 + \sigma^2),$$

with parameters

$$\mu = K(X_{1:n}^{(\text{train})}, X) \mathbb{e} \left(K(X_{1:n}^{(\text{train})}, X_{1:n}^{(\text{train})}) + \sigma^2 I_n \right)^{-1} Y_{1:n}^{(\text{train})},$$

$$\sigma^2 = K(X, X) - K(X_{1:n}^{(\text{train})}, X) \mathbb{e} \left(K(X_{1:n}^{(\text{train})}, X_{1:n}^{(\text{train})}) + \sigma^2 I_n \right)^{-1} K(X_{1:n}^{(\text{train})}, X).$$

Here, $K(X_{1:n}^{(\text{train})}, X)$ is a vector with elements $K_i = K(X_i^{(\text{train})}, X)$.

### 2.2.1 BALD for Gaussian Processes

We can derive the BALD objective for Gaussian Processes using the differential entropy of a Gaussian random variable, given in eq. (4). The first term in the acquisition function in eq. (2), follows from the predictive distribution (eq. (8))

$$\mathcal{H}(Y \mid X, X_{1:n}^{(\text{train})}, Y_{1:n}^{(\text{train})}) \propto \log(\sigma_\ast^2 + \sigma^2) + \text{const.},$$

and the second term is

$$\mathbb{E}_{f \mid X_{1:n}^{(\text{train})}, Y_{1:n}^{(\text{train})}}[\mathcal{H}(Y \mid X, f)] \propto \log(\sigma^2) + \text{const.}$$

Hence, the full BALD objective for GPs is given by

$$\text{MI}(Y; f \mid X, X_{1:n}^{(\text{train})}, Y_{1:n}^{(\text{train})}) \propto \log(\sigma_\ast^2 + \sigma^2) - \log \sigma^2,$$

as the constants cancel each other out.
3 BALD with weak supervision

In what follows, we will describe the approach proposed in this paper. In broad terms, the aim is to, at each iteration of the active learning process, be able to select both the instance to annotate and the precision of the annotation, referred to as $\alpha$. The motivation is that if we can acquire noisy, but cheaper, annotations, then we can allow the model to explore a larger part of the input space given the budget constraints. Thus, we replace the optimisation problem given in eq. (1) with

$$X^{(a)}, \alpha^{(a)} = \underset{X \in X_{\text{pool}}, \alpha \in \mathcal{A}}{\text{argmax}} \phi(X, \alpha), \tag{13}$$

where the acquisition function depends on both the input and the annotation precision. We will assume that the possible values of the precision, $\alpha$, is given by the set $\mathcal{A}$.

Subsequently, we assume that we observe the weak, or noisy, target variable $\tilde{Y}$ in place of the clean target, $Y$. The distribution of $\tilde{Y}$ will depend on the selected annotation precision $\alpha$ as well as the input, $X$, and/or the true target, $Y$, depending on how the noise is modelled. Note that, as $\alpha$ is selected and therefore known, we can account for the noise when learning the model.

The acquisition function that we propose is an extension of the BALD objective that includes the precision, $\alpha$. To optimise mutual information and at the same time account for the annotation costs, in the greedy fashion of iterative active learning, we optimise information per cost. Introducing the cost function $C(\alpha)$, describing the cost of acquiring an annotation with precision $\alpha$, the proposed acquisition function is

$$\phi(X, \alpha) = \frac{\text{MI}(\tilde{Y} \mid f, X, \alpha, X_{\text{train}}, \tilde{Y}_{\text{train}}, \alpha_{\text{train}})}{C(\alpha)}, \tag{14}$$

Hence, we want to select the instance, and corresponding annotation precision, for which the mutual information between the noisy target, $\tilde{Y}$, and the model, $f$, is high, while at the same time keeping the annotation cost low. A description of the proposed approach, referred to as BALD with weak supervision (BALD WS), is found in algorithm 2.

3.1 BALD WS for Gaussian Processes

For a Gaussian Process with noisy target, $\tilde{Y}$, we replace $Y$ in eq. (5) with

$$\tilde{Y} \mid f, X \sim \mathcal{N}(f(X), \frac{\sigma^2}{\alpha})$$

The annotation precision, $\alpha \in \mathcal{A} = (0.0, 1.0]$, using this noise model, controls the variance of the noisy target variable. The highest precision is $\alpha = 1.0$, for which the variance of $\tilde{Y}$ is the same as that of the true target. The distribution of $\tilde{Y}_{\text{train}}$ with corresponding precisions given by $\alpha_{\text{train}}$ is

$$\tilde{Y}_{\text{train}} \mid f_{\text{train}}, \alpha_{\text{train}} \sim \mathcal{N}(f_{\text{train}}, \sigma^2(\text{diag}(\alpha_{\text{train}}))^{-1}), \tag{15}$$
Algorithm 2 Active Learning with Weak Supervision

Require: $B$, $X_{1:n}^{(train)}$, $Y_{1:n}^{(train)}$, $\alpha_{1:n}^{(train)}$ $X_{1:N}^{(pool)}$

$b := 0$
Initialise model

while $b < B$ do
  model.train($X_{1:n}^{(train)}$, $Y_{1:n}^{(train)}$, $\alpha_{1:n}^{(train)}$) % Fit model to training data
  $X^{(a)}$, $\alpha^{(a)} = \arg\max_{X \in X_{1:N}^{(pool)}, \alpha \in \mathcal{A}} \phi(X, \alpha)$ % Select data point and precision
  Acquire annotation $\tilde{Y}^{(a)}$ for $X^{(a)}$ with precision $\alpha^{(a)}$

  $X_{1:n} \leftarrow X_{1:n} \cup X^{(a)}$
  $\tilde{Y}_{1:n} \leftarrow \tilde{Y}_{1:n} \cup \tilde{Y}^{(a)}$
  $\alpha_{1:n} \leftarrow \alpha_{1:n} \cup \alpha^{(a)}$ % Add $X^{(a)}$, $\tilde{Y}^{(a)}$, $\alpha^{(a)}$ to training data

  $b = b + C(\alpha^{(a)})$
end while

where we use $\text{diag}(\alpha_{1:n}^{(train)})$ to denote the $n \times n$ diagonal matrix with elements $\alpha_{1:n}^{(train)}$.

The predictive distribution of a new observation $X$ with precision $\alpha$, following eq. (3), is given by

$$\tilde{Y} \mid X, X_{1:n}^{(train)}, \tilde{Y}_{1:n}^{(train)}, \alpha_{1:n}^{(train)} \sim N(\tilde{\mu}_*, \tilde{\sigma}_*^2 + \sigma^2 \alpha),$$

(16)

with parameters

$$\tilde{\mu}_* = K(X_{1:n}^{(train)}, X)^\top \left(K(X_{1:n}^{(train)}, X_{1:n}^{(train)}) + \sigma^2 \text{diag}(\alpha_{1:n}^{(train)})^{-1}\right)^{-1} \tilde{Y}_{1:n},$$

(17)

$$\tilde{\sigma}_*^2 = K(X, X) - K(X_{1:n}^{(train)}, X)^\top \left(K(X_{1:n}^{(train)}, X_{1:n}^{(train)}) + \sigma^2 \text{diag}(\alpha_{1:n}^{(train)})^{-1}\right)^{-1} K(X_{1:n}^{(train)}, X).$$

Following this, the mutual information between the model and the noisy target variable follows as

$$\text{MI}(\tilde{Y}; f \mid X, X_{1:n}^{(train)}, \tilde{Y}_{1:n}^{(train)}) \propto \log(\tilde{\sigma}_*^2 + \frac{\sigma^2}{\alpha}) - \log \frac{\sigma^2}{\alpha},$$

(18)

which we then plug into eq. (14).
4 Experiments

We provide experiments to empirically validate the proposed approach. The data used in the experiments is generated as follows

\[ X \sim \text{Uniform}(0, x_{\text{max}}), \]

\[ \tilde{Y} \mid X, \alpha \sim \mathcal{N}(AX \cdot \sin(\omega X), \sigma^2_{\alpha}). \quad (19) \]

As default, we let \( x_{\text{max}} = 5, A = 0.2, \omega = 3 \) and \( \sigma^2 = 0.01 \). An example of 1,000 data points sampled using this generative model with \( \alpha = 1.0 \) is shown in fig. [1].

The data sets used in the experiments consists of 8,000 data points, whereof 75% is allocated to the data pool and the remaining 25% to a test set. For the initial training set, \( n = 10 \) data points are randomly sampled from the (unannotated) data pool. Corresponding target values are sampled with a precision of \( \alpha = 1.0 \).

In the experiments, we use GP regressors with RBF kernels, where the hyperparameters are set to \( a = 1.0 \) and \( l = 1.0 \). The proposed approach (BALD WS) is compared to using the standard BALD objective function, as well as to using random sampling. Random sampling in this case, means that instances are sampled at random from the data pool and that annotations are acquired with a precision of \( \alpha = 1.0 \).

For the proposed approach, we optimise the objective with respect to \( \alpha \) by making a discretisation of the interval \((\alpha_{\text{min}}, 1.0)\), where \( \alpha_{\text{min}} \) is a lower threshold on the precision. Note that although we choose to discretise the interval in these experiments, it should also be possible to directly optimise the acquisition function over the full interval, using an optimisation method of choice. The lower threshold, \( \alpha_{\text{min}} \), sets a limit to the amount of noise with which an annotation can be acquired. It also gives rise to a basic cost for initialising annotation. In the experiments, we use \( \alpha_{\text{min}} = 0.1 \).

Effect of cost function The BALD WS acquisition function is defined for a general cost function, \( C(\alpha) \). In practice, this cost function can be adapted to fit the task at hand. Thus, in the following experiments, we investigate the effect of the cost function on the precision selection and the performance of the proposed approach. For this purpose, we perform active learning experiments using cost functions of the form \( C(\alpha) = \alpha^q \), with
\( q = 0.2, 0.8, 1.0 \) and \( 2.0 \). We let the budget be fixed at \( B = 40 \), excluding the annotation costs of the initial training data points (which is constant for all cost functions). In fig. 2, we show the mean test Mean Squared Error (MSE) over five experiments, as a function of the total annotation costs. Note that for the curve corresponding to the BALD WS objective with cost function \( C(\alpha) = \alpha^{0.8} \), we use interpolation for the purpose of averaging the results. For each repeated experiment, we sample a new data set of 8,000 observations according to eq. (19).

We observe that the cost functions has a substantial effect on the results, where the most gain with being able to select the annotation precision is obtained for \( C(\alpha) = \alpha^2 \). Using this cost function, it is much cheaper to obtain noisy annotations than annotations of high precision, and the proposed algorithm consistently selects the lowest precision possible. In contrast, using the cost function \( C(\alpha) = \alpha^{0.2} \) the BALD WS approach always selects annotations of highest precision, making it equivalent to standard BALD.

For the cost function \( C(\alpha) = \alpha^{0.8} \), the optimal annotation precision differs over iterations. We show the selected precisions over the course of the first two experiments using the BALD WS objective in fig. 3. We can see that the algorithm starts with selecting annotations of low precision. Thereafter, the annotation precision is gradually increased, until the highest precision is reached. We might interpret the general patterns as portraying an initial phase of exploration. Note that the exception seen in experiment 2 (fig. 3b)), where an annotation of low precision is selected just at the end, is due to the budget being exhausted.

**Effect of data distribution** We next test the proposed approach in a setting where we assume that part of the input space is under-explored. In this setting, observations in the data pool are sampled according to

\[
\begin{align*}
Z &\sim \text{Bernoulli}(p) \\
X \mid Z = 0 &\sim \text{Uniform}(0, \frac{x_{\text{max}}}{2}) \\
X \mid Z = 1 &\sim \text{Uniform}(\frac{x_{\text{max}}}{2}, x_{\text{max}})
\end{align*}
\]  

(20)

where \( Z \) is used as an auxiliary variable. In the experiments, we set \( p = 0.1 \) and \( x_{\text{max}} = 5 \). Although part of the input space is under-explored, reflected in the data pool, we want the model to perform well on the full input space. Therefore, the test set used in the experiments is still sampled according to eq. (19). We argue that circumstances like these are not uncommon in practice. For example, we could have a situation where it is easier to collect data in some parts of the input space than others, but where the unbalance is not present in the real scenario. Another relevant situation is the one in which the input space is in fact non-uniform, but where it is important for the model to perform well also for rare observations.

We perform three set of experiments, generating targets using \( \omega = 3, 5 \) and 7, respectively, in eq. (19). In these experiments we use a fixed budget of \( B = 90 \), excluding the initial training data points, and cost function \( C(\alpha) = \alpha \). The mean test MSE over five experiments is reported in fig. 4. We observe that both the standard BALD and the proposed BALD WS acquisition function gives significantly better model performance.
Figure 2: Active learning experiments using different cost functions for the annotation precision.

(a) $\mathcal{C}(\alpha) = \alpha^{0.2}$

(b) $\mathcal{C}(\alpha) = \alpha^{0.8}$

(c) $\mathcal{C}(\alpha) = \alpha$

(d) $\mathcal{C}(\alpha) = \alpha^2$

Figure 3: Annotation precision selected over iterations using the BALD WS acquisition function and with $\mathcal{C}(\alpha) = \alpha^{0.8}$.
than random sampling. Moreover, as the frequency, $\omega$, of the generative distribution of $\tilde{Y}$ is increased, the advantage of BALD WS over BALD gets more apparent. It seems like it is extra beneficial to be able to observe more data points, and explore a larger part of the input space, when the curve has a higher frequency.

We can see a slight decrease in performance towards the end of the experiments for the standard BALD acquisition function when $\omega = 7$. We hypothesise that this might be a result of the model not being flexible enough to fit the true mean of the data, and therefore that the predictive variance of the model is not reliable. This could potentially be alleviated by letting the model learn the hyperparameters of the RBF kernel from data.

5 Conclusion

In this paper, we proposed an extension of active learning for Gaussian Processes, using an acquisition function that allows not only to select which observation to annotate, but also the precision of the acquired annotation. We demonstrated empirically how the proposed approach can give a performance advantage in situations where it is cheaper
to obtain several noisy annotations, in place of one or a few precise. In addition, it can improve performance in situations where it is beneficial to explore the input space to a larger degree than what is allowed by the budget if only annotations of high precision are acquired.

In the experiments performed in this paper, the optimisation over the annotation precision, $\alpha$, was performed over a grid of precision values. In the future, it might be possible to optimise over $\alpha$ using a gradient search or similar, depending on the task. Moreover, we considered acquiring only one annotation per iteration in the active learning algorithm. An extension of the proposed approach would be to explore the possibility of performing several queries per iteration, using for example batchBALD (Kirsch et al., 2019).

**Acknowledgements**

This research is financially supported the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.
References

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, volume 4. Springer, New York.

Chakraborty, S. (2020). Asking the Right Questions to the Right Users: Active Learning with Imperfect Oracles. *AAAI Conference on Artificial Intelligence*.

Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory*, volume 2. Wiley, New Jersey.

Dawid, A. P. and Skene, A. M. (1979). Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28.

Gao, R. and Saar-Tsechansky, M. (2020). Cost-Accuracy Aware Adaptive Labeling for Active Learning. *AAAI Conference on Artificial Intelligence*.

Guan, M. Y., Gulshan, V., Dai, A. M., and Hinton, G. E. (2018). Who Said What: Modeling Individual Labelers Improves Classification. *arXiv preprint arXiv 1703.08774*.

Herde, M., Kottke, D., Huseljic, D., and Sick, B. (2020). Multi-annotator Probabilistic Active Learning. *IEEE International Conference on Pattern Recognition*.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian Active Learning for Classification and Preference Learning. *arXiv preprint arXiv: 1112.5745*.

Huang, S. J., Chen, J. L., Mu, X., and Zhou, Z. H. (2017). Cost-Effective Active Learning from Diverse Labelers. *International Joint Conference on Artificial Intelligence*.

Kirsch, A., van Amersfoort, J., and Gal, Y. (2019). BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning. *Advances in Neural Information Processing Systems*.

Rasmussen, C. and Williams, C. K. (2006). *Gaussian Processes for Machine Learning*. MIT press, Massachusetts.

Settles, B. (2010). Active Learning Literature Survey. Technical Report Computer Sciences Technical Report 1648, University of Wisconsin, Madison.

Yan, Y., Rosales, R., Fung, G., and Dy, J. G. (2011). Active Learning from Crowds. *International Conference on Machine Learning*.
Supplementary material for Active Learning with Weak Labels for Gaussian Processes

In this supplementary material, we provide additional experiments for which we investigate the effects of learning the hyperparameters of the GP kernel.

A Additional experiments: learning hyperparameters

As an extension to the experiments given in the main paper, we evaluate the effect of learning the hyperparameters of the RBF kernel during the active learning loop. For this experiment, we generate data according to eq. (19) with $\omega = 3$ and $5$. We use the cost function $C(\alpha) = \alpha$ and a budget of $B = 90$. We learn the hyperparameters of the RBF kernel using data $(X_{1:n}^{(train)}, Y_{1:n}^{(train)})$, by minimising the marginalised negative log-likelihood

$$
\text{NLL} \propto \log |K(X_{1:n}^{(train)}, X_{1:n}^{(train)}) + \sigma^2 (\text{diag}(\alpha_{1:n}^{(train)}))^{-1}| \\
+ \tilde{Y}_{1:n}^{(train)} \top \left(K(X_{1:n}^{(train)}, X_{1:n}^{(train)}) + \sigma^2 (\text{diag}(\alpha_{1:n}^{(train)}))^{-1}\right)^{-1} \tilde{Y}_{1:n}^{(train)}.
$$

The mean test MSE over five experiments, with fixed and learned hyperparameters, is shown in fig. 5. While learning the hyperparameters when data is generated with $\omega = 3$ does not seem to have a large effect on the conclusions regarding the relative performance of the active learning algorithms, the test MSE is improved in general. This general improvement in performance is also seen when learning the hyperparameters on data generated with $\omega = 5$. However, in the latter case, we observe that random sampling overtakes the other two methods after approximately half of the experiments when hyperparameters are fixed, see fig. 5(c). In contrast, this effect is not seen in fig. 5(d) when hyperparameters are learned. It could be that using inappropriate values of the hyperparameters of the GP kernel causes the predictive variance of the model to be less reliable, and that the BALD-based objective functions therefore do not select useful instances for annotation.
Figure 5: Effect of learning the hyperparameters of the GP kernel in active learning experiments.