QUERY BY SELF

Anonymous authors
Paper under double-blind review

ABSTRACT

Training with hard-to-obtain and therefore valuable data, improving generalization performance, and accelerating training speed are all challenging problems in machine learning community. Active learning , whose performance depends on its query strategy, is a powerful tool for these challenges. Unlike the famous Query by Committee strategy aimed at classification problem and dependent on a committee of student, our proposed query by self is suitable to both classification and regression problem and requires only a student, which benefits from estimated output variance, the intermediate product of Kalman filtering optimization. This means larger scope of application and less requirement for computation and the number of data. Besides, this strategy reduces training time and improves accuracy via filtration of similar data and better generalization. We theoretically explain query by self strategy from the perspective of entropy. To verify effectiveness of query by self empirically, we conduct several experiments on two classical models in machine learning.

1 INTRODUCTION

Massive amount of computation, money, and time are necessary to acquire enough data for training a neural network model with certain accuracy in some research areas, for example, molecular dynamics modeling (Zhang et al., 2020) with ab initio accuracy based on density functional theory (Kohn & Sham, 1965)). Therefore, it is imperative to find such a method that selects out possibly informative and representative data leading the model to a given accuracy with a thus relative small training dataset. Active learning (Settles, 2009; Olsson, 2009) is an eligible technique, which is applied to help computational chemistry (Podryabinkin & Shapeev, 2017), chemical physics (Smith et al., 2018), materials design and discovery (Jablonka et al., 2021), mental health treatment (Ahmed et al., 2021), drug discovery (Graff et al., 2021) recently. Besides, active learning can solve imbalanced data problem (Dong, 2021).

The dataset of an active learning task is divided into labeled dataset and unlabeled dataset. The training occurs on the labeled one, thereafter generating unlabeled dataset usually via three methods (Membership query synthesis (Angluin, 1988), Stream-Based Selective Sampling (Atlas et al., 1989), Pool-Based Sampling (Lewis, 1995)) and implementing query strategies on the unlabeled dataset to select part of data for teacher to label. These labeled input data of choice are merged into labeled dataset for training the model further and a whole round of active learning ends.

Efficient query strategies design is at the center of active learning. The performance of whole algorithm heavily depends on those query strategies as they decide whether or not unlabeled samples are sent to teacher. Hence, various query strategies are proposed for different tasks (Query by Committee (Seung et al., 1992), Uncertainty Sampling (Lewis, 1995), Query by Bagging and Boosting (Abe & Mamitsuka, 1998), Expected Model Change (Settles et al., 2007), Expected Error Reduction (Roy & McCallum, 2001), Variance Reduction (MacKay, 1992; Cohn, 1993; Cohn et al., 1996), Density-Weighted Methods (Settles & Craven, 2008)) in past research. Some of them are discussed in the next section. However, these strategies have their own drawbacks. Query by Committee has to training several models together with same dataset. Uncertainty Sampling, Expected Model Change and Expected Error Reduction can not solve regression problem. Query by Bagging and Boosting, Variance Reduction and Density-Weighted Methods need large computation.

Main Contributions. To tackle the aforementioned difficulties, we propose a novel query strategy, named query by self strategy, inspired by Kalman filter (Kalman et al., 1960; Welch et al., 1995)

giving a direct estimate of output variance, and thus no longer necessary to train extra models for variance like query by committee.

- Our proposed query by self strategy, requires only one model or student, is suitable for both classification and regression tasks, has a new criterion to describe output variance which reduces the computational complexity of variance reduction $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$ and is handy to combine with other common query strategies.
- Theoretical explanation and analysis of the efficacy of query by self based on information theory. Here, we delineate the connection among Kalman filtering, Fisher information matrix, recursive least square algorithm and variance reduction.
- Accelerate the training, improving generalization performance and protecting algorithm from over-fitting.

2 Related work

Queries strategies of direct relation to our query by self are query by committee (Seung et al., 1992) and variance reduction (MacKay, 1992; Cohn, 1993; Cohn et al., 1996). Query by self estimates variance with a different methods from them. Since the label of an unlabeled sample is unknown, the algorithm approximates the variance of output via linearization and Kalman filter (Haykin & Haykin, 2001; Kalman et al., 1960; Welch et al., 1995).

Query by Committee. Both query by self and query by committee (Seung et al., 1992) estimate variance of output but in different way. The former approximates the variance through quadratic form (covariance matrix $\mathbf{P}$ in the next section) at current derivatives of outputs w.r.t. trainable parameters theoretically, while the latter estimates it empirically $\sum_i(\hat{y}_i - \langle\hat{y}\rangle)^2/(n-1)$, where $\hat{y}_i$ is the output of the i-th student or model and $\langle\hat{y}\rangle$ is the mean value of these n models. This unbiased estimator of variance consumes n times computation.

Variance Reduction. Same as query be self, variance reduction (MacKay, 1992; Cohn, 1993; Cohn et al., 1996) does not need re-training and committee, but its flaw is also obvious, $\mathcal{O}(N^3)$ computational complexity in solving inverse of Fisher information matrix (Schervish, 2012). We employ extended Kalman filter (Smith et al., 1962) to overcome this difficulty and reduce the time complexity to $\mathcal{O}(N^2)$, where N is the number of trainable parameters of neural networks. This is achieved by

$$\frac{\partial^2}{2\partial\theta_i\partial\theta_j}(\hat{y}_\theta(x)-y)^2 \approx \frac{\partial}{\partial\theta_i}\hat{y}_\theta(x)\frac{\partial}{\partial\theta_j}\hat{y}_\theta(x)$$

omitting the second derivative term and the iterative algorithm in kalman filter to solve the inverse of $\mathbf{P}_t$.Therefore, query by self is an approximate and fast version of variance reduction (MacKay, 1992; Cohn, 1993; Cohn et al., 1996).

Fisher Information Matrix. This matrix (Schervish, 2012)

$$\mathcal{I}(\theta)_{i,j} := -\mathbb{E}_{(x,y)}\left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}(\log\mathbb{P}_\theta(y|x))\middle|\,\theta\right]$$

measures how much information we can get from a given query x at current trainable parameters θ. The determinate $\det(\mathcal{I}^{-1})$ (Chaloner & Verdinelli, 1995) and reference trace (MacKay, 1992) $\mathrm{Tr}(vv^\mathsf{T}\mathcal{I}^{-1}) = v^\mathsf{T}\mathcal{I}^{-1}v$ of $\mathcal{I}^{-1}$ are usually adopted as query criteria, where $v := v(\nabla_\theta\hat{y})$ is a reference vector dependent on $\nabla_\theta\hat{y}$. The former can be regarded as a measure of volume in version space (Seung et al., 1992) and we combine the latter with Kalman filter as query by self for reduction in computational complexity.

Kalman Filtering for Neural Network. Extending Kalman filter to nonlinear case is named extended Kalman filter (Smith et al., 1962). For neural network, it is an optimizer on which there exists much mature research (Haykin & Haykin, 2001). At the very beginning, the optimizer based on Kalman filter without any covariance omitted is global extended Kalman filter (Chui et al., 2017). Luckily, we find using the global extended Kalman filter as an optimizer during active learning, it is unnecessary to solve the inverse of Fisher information matrix on purpose which takes to much time during training, since the weights error covariance matrix $\mathbf{P}$, an intermediate during training based

Kalman filter, is nothing but an approximation of the inverse of Fisher information matrix. To reduce the computational complexity further, variants of extended Kalman filter should be considered such as NDEKF (Murtuza & Chorian, 1994), ONDEKF, LDEKF, and FDEKF. The performance of these variants (Heimes, 1998) are compared.

3 ALGORITHM

As stated above, query by self can be applied to both classification 1 and regression 2 problem. Although there is still much space to optimize the efficiency of query by self strategy, we show its pseudo-code as following for clarity and simplicity. The active learning part unfolds in two subsection, one for classification another for regression. In addition to the active learning part, the rest of the algorithm is updating weights via Kalman filter. To reduce complexity, we compressed the output dimension into one. This keep us from solving the inverse of output covariance matrix. See Appendix A.1 for more related information and background for Kalman filter. Finally, we indicate several extensions of query by self algorithm. In these cases, it collaborates with common optimizer (SGD, Adam) and committee-based query strategies (query by committee, query by bagging and query by boosting) to become a even better query strategies or just accelerate training process, avoid over-fitting problem, and improve generalization performance.

3.1 QUERY BY SELF FOR CLASSIFICATION

h is the neural-network-based classification model before softmax with trainable weights $\boldsymbol{w}_{t-1}$ vector at a given timestep $t \in \{1, 2, \ldots, T\}$, x_t is input data, and l_t is corresponding label. Therefore, $\{(x_t, l_t)\}_{0 \leq t \leq t_0}$ is the dataset of t_0 labeled samples and $\{x_t\}_{t_0+1 \leq t \leq T}$ are unlabeled input samples randomly chosen from sample space dominated by certain distribution. Our target is to find the trainable weights $\boldsymbol{w}_T$ vector at the last timestep T for neural network model in order to achieve a given accuracy by labeling samples as few as possible.

Algorithm 1 Query by self for classification, a novel and Kalman-filter-based query strategy for choosing representative data (see subsection 3.1). $\langle v \rangle$ means calculating the mean of v's components. $\mathbf{e}_i$ is a unit vector, whose i-th component is 1. $\mathrm{sign}(x) := \mathbf{1}_{[0,+\infty)}(x)$. Hyper-parameters for out experiments are $\nu = 0.9937$, $\lambda_1 = 0.988$, $\mathbf{P}_1 = \mathbf{I}$.

Input: $\{\boldsymbol{w}_0\}$ (default initial weight vector), $\{(x_t, l_t)\}_{0 \leq t \leq t_0}$(dataset of labeled samples), $\{x_t\}_{t_0+1 \leq t \leq T}$ (randomly chosen unlabeled input samples).

for $t = 1, 2, \ldots, T$ **do**
  $\hat{y}_t \leftarrow h(\boldsymbol{w}_{t-1}, x_t)$ (Objective function with weights $\boldsymbol{w}_{t-1}$)
  $\hat{s}_t \leftarrow \mathrm{softmax}(\hat{y}_t)$ (Get probability vector)
  if $t > t_0$ **then**
    $\hat{l}_t \leftarrow \arg\max(\hat{s}_t)$ (Get predicted label)
    $S_t \leftarrow \nabla_w \hat{s}_{t,\hat{l}_t} \mid_{w=\boldsymbol{w}_{t-1}}$ (Get gradient of probability of predicted label w.r.t.$\boldsymbol{w}_{t-1}$)
    $c_t \leftarrow \lambda_t + S_t^{\mathsf{T}} \mathbf{P}_t S_t$ (Variance criterion, which can be replaced with another criterion)
    if $c_-(t) < c_t < c_+(t)$ **then**
      Let the teacher label x_t with l_t. (Get new labeled sample)
    else
      Continue
    end if
  end if
  $H_t \leftarrow \nabla_w \langle \mathrm{sign}(\mathbf{e}_{l_t} - \hat{s}_t)\hat{s}_t \rangle \mid_{w=\boldsymbol{w}_{t-1}}$ (Compress output dimension and get its gradient)
  $K_t \leftarrow \mathbf{P}_t H_t$ (Calculate Kalman Gain)
  $a_t \leftarrow 1/(\lambda_t + H_t^{\mathsf{T}} K_t)$ (Calculate the inverse of output variance)
  $\mathbf{P}_{t+1} \leftarrow (\mathbf{P}_t - a_t K_t K_t^{\mathsf{T}})/\lambda_t$ (Update the weights covariance matrix)
  $\boldsymbol{w}_t \leftarrow \boldsymbol{w}_{t-1} + a_t K_t \langle |\mathbf{e}_{l_t} - \hat{s}_t| \rangle$ (Update weights)
  $\lambda_{t+1} \leftarrow \lambda_t \nu + 1 - \nu$ (Update memory factor λ_t via moving average with parameter ν)
end for

Output: $\{\boldsymbol{w}_T\}$ (Wanted weights or trainable parameters)

Before timestep t_0, these weights $\boldsymbol{w}_t$ are updated according to Kalman fiter optimizer and we obtain a half-trained model. After that, we refer to previous criteria in literature on active learning and propose some criteria c_t, only dependent on the output or predict $\hat{y}_t$ of a student, judging whether a sample is representative and critical enough to be an ideal sample in terms of training efficiency. Those samples with a positive answer will be utilized to update weights for another half of training.

Variance and Standard Deviation. They are approximated by

$$\mathcal{V}(\hat{s}_{t,\hat{l}_t}) := \lambda_t + S_t^{\mathsf{T}} \mathbf{P}_t S_t \quad \text{and} \quad \sqrt{\mathcal{V}}$$

up to a factor respectively, where $\hat{s}_t$ is the probability vector with $\Sigma_i \hat{s}_{t,i} = 1$, $\hat{l}_t$ is the predicted label, λ_t is the memory factor, $\mathbf{P}_t$ is the error weights covariance matrix and $S_t = \nabla_w \hat{s}_{t,\hat{l}_t} \mid_{w=\boldsymbol{w}_{t-1}}$. This criterion describes how uncertain our model is to estimate the probability of the predicted label. The larger it is, the more necessary the current sample is needed to be labeled (more different from labeled samples). However, if it is extremely large, we should consider whether or not it is a sample out of input domain of our interest.

Entropy.

$$\mathcal{S}(\hat{s}_t) := -\Sigma_i \hat{s}_{t,i} \ln(\hat{s}_{t,i}).$$

It represents the information gained if the current sample is labeled. Thus, we should label samples with entropy as higher as we can unless it is very noisy. Sample with high entropy usually near a point where many categories assemble around.

Margin.

$$\mathcal{M}(\hat{s}_t) := \mathbf{M}_1(\hat{s}_t) - \mathbf{M}_2(\hat{s}_t) \in [0, 1],$$

where $\mathbf{M}_n$ is the n-th largest number in a vector. Since the change of the output label is only related to $\mathbf{M}_1(\hat{s}_t)$ and $\mathbf{M}_2(\hat{s}_t)$, the margin describes the stability and uncertain of predict. Input samples with low margin delineate the boundary between two categories. The lower this quantity is, the more critical the current sample is.

Clarity.

$$\mathcal{C}(\hat{s}_t) := \frac{\#(\hat{s}_t)\hat{s}_{t,\hat{l}_t} - 1}{\#(\hat{s}_t) - 1} \in [0, 1]$$

where $\hat{s}_{t,\hat{l}_t}$ is defined as above and $\#$ is the length of a vector. It means how clear current classifier is on the prediction. In most cases, we should label those inputs with lower clarity.

With these simple criteria defined, we can construct composite criteria such as $(\mathbf{M}_1(\hat{s}_t) - \mathbf{M}_2(\hat{s}_t))\sqrt{\lambda_t + S_t^{\mathsf{T}} \mathbf{P}_t S_t}$, $(\mathbf{M}_1(\hat{s}_t) - \mathbf{M}_2(\hat{s}_t))/\sqrt{\lambda_t + S_t^{\mathsf{T}} \mathbf{P}_t S_t}$ (similar to signal-to-noise ratio), and $(\mathbf{M}_1(\hat{s}_t) - \mathbf{M}_2(\hat{s}_t))^2 + (\mathcal{S}(\hat{s}_t))^{-2}$. $c_-(t)$ and $c_+(t)$ are two bound limiting samples to not only enough difference from labeled sample and criticality (meaning samples near sharp boundary between categories) but reasonable and tolerable noise as well. Generally, choosing proper constants for $c_-(t)$ and $c_+(t)$ works well enough.

3.2 Query by Self for Regression

Using similar notation as in case of classification, we show the algorithm of query by self for regression as bellow. There are two difference. The first is only the variance and standard deviation criterion is meaningful. The second is the output is a general vector, no longer be a probability vector.

Variance and Standard Deviation. The variance of output $\hat{y}$ is given by

$$\sigma_{\hat{y}}^2 \approx (\nabla_w \hat{y})^{\mathsf{T}} \mathcal{I}^{-1} \nabla_w \hat{y}$$

proposed by MacKay (1992). According to theorem 3, we have $\mathbf{P}_t \sim \mathcal{I}^{-1}$ when $t \to \infty$. Hence, adding a regularization term λ_t, we define

$$\mathcal{V}(\hat{y}_t) := \lambda_t + S_t^{\mathsf{T}} \mathbf{P}_t S_t \quad \text{and} \quad \sqrt{\mathcal{V}}$$

as an approximation to variance and standard deviation respectively, where $\hat{y}_t$ is the output vector and $S_t = v(\nabla_w \hat{y}_t \mid_{w=\boldsymbol{w}_{t-1}})$ is reference vector, usually a linear combination of column vectors in

$\nabla_w \hat{y}_t \mid_{w=\boldsymbol{w}_{t-1}}$. It is feasible to fetch several reference vectors and sum their variances or standard deviations up as a new one (as in algorithm 2). This criterion describes how uncertain our model is about the prediction. The larger it is, the more necessary the current sample is needed to be labeled (more different from labeled samples). However, if it is extremely large, we should consider whether or not it is a sample out of input domain of our interest.

Algorithm 2 Query by self for regression, a novel and Kalman-filter-based query strategy for choosing representative data (see subsection 3.2). $\langle v \rangle$ means calculating the mean of v's components. $\mathbf{e}_i$ is a unit vector, whose i-th component is 1. $\text{sign}(x) := \mathbf{1}_{[0,+\infty)}(x)$. Hyper-parameters for out experiments are ν = 0.98, λ_1= 0.9987, $\mathbf{P}_1 = \mathbf{I}$.

Input: $\{\boldsymbol{w}_0\}$ (default initial weight vector), $\{(x_t, l_t)\}_{0 \le t \le t_0}$(dataset of labeled samples), $\{x_t\}_{t_0+1 \le t \le T}$ (randomly chosen unlabeled input samples).

for $t = 1, 2, \ldots, T$ **do**
  $\hat{y}_t \leftarrow h(\boldsymbol{w}_{t-1}, x_t)$ (Objective function with weights $\boldsymbol{w}_{t-1}$)
  if $t > t_0$ **then**
    for $i = 1, 2, \ldots, n$ **do**
      $S_{t,i} \leftarrow v_i(\nabla_w \hat{y}_t \mid_{w=\boldsymbol{w}_{t-1}})$ (Get reference vector of gradient w.r.t.$\boldsymbol{w}_{t-1}$)
      $c_{t,i} \leftarrow \lambda_t + S_{t,i}^{\mathsf{T}} \mathbf{P}_t S_{t,i}$(Variance criterion, which can be replaced with another)
    end for
    if $c_-(t) < \Sigma_i c_{t,i} < c_+(t)$ **then**
      Let the teacher label x_t with y_t. (Get new labeled sample)
    else
      Continue
    end if
  end if
  $H_t = \nabla_w \langle \text{sign}(y_t - \hat{y}_t) \hat{y}_t \rangle \mid_{w=\boldsymbol{w}_{t-1}}$(Compress output dimension and get its gradient)
  $K_t \leftarrow \mathbf{P}_t H_t$ (Calculate Kalman Gain)
  $a_t \leftarrow 1/(\lambda_t + H_t^{\mathsf{T}} K_t)$ (Calculate the inverse of output variance)
  $\mathbf{P}_{t+1} \leftarrow (\mathbf{P}_t - a_t K_t K_t^{\mathsf{T}})/\lambda_t$ (Update the weights covariance matrix)
  $\boldsymbol{w}_t \leftarrow \boldsymbol{w}_{t-1} + a_t K_t \langle |y_t - \hat{y}_t| \rangle$ (Update weights)
  $\lambda_{t+1} \leftarrow \lambda_t \nu + 1 - \nu$ (Update memory factor λ_t via moving average with parameter ν)
end for

Output: $\{\boldsymbol{w}_T\}$ (Wanted weights or trainable parameters)

Algorithm 3 Extended query by self for classification.

Input: $\{\boldsymbol{w}_0\}$ (default initial weight vector), $\{(x_t, l_t)\}_{0 \le t \le t_0}$(dataset of labeled samples), $\{x_t\}_{t_0+1 \le t \le T}$ (randomly chosen unlabeled input samples).

for $t = 1, 2, \ldots, T$ **do**
  $\hat{y}_t \leftarrow h(\boldsymbol{w}_{t-1}, x_t)$ (Objective function with weights $\boldsymbol{w}_{t-1}$)
  $\hat{s}_t \leftarrow \text{softmax}(\hat{y}_t)$ (Get probability vector)
  if $t \le t_0$ **or** $c_-(t) < c(\hat{s}_t) < c_+(t)$ **then**
    Get data (x_t, l_t) via dataset ($t \le t_0$) or teacher's labeling ($t \ge t_0$).
  else
    Continue
  end if
  Based on (x_t, l_t), update $\boldsymbol{w}_{t-1}$ by common optimizers. (e.g. SGD, Adam)
end for

Output: $\{\boldsymbol{w}_T\}$

3.3 Extensions

Although query by self benefits from Kalman filtering, it can be extended to other neural networks with common optimizer (SGD, Adam). We extend query by self strategy to common optimizers. Using the same notation as in algorithm 1, the pseudo-code is like algorithm 3.

Except for choosing and labeling extra representative sample, query by self strategy can be utilized to accelerate training process, avoid overfitting, improve generalization performance (see algotirhm 4).

Algorithm 4 Extended query by self for acceleration , avoiding overfitting, and improving generalization performance.

Input: $\{\boldsymbol{w}_0\}$ (default initial weight vector), $\{(x_t, l_t)\}_{0\leq t\leq T}$(dataset).
for $t = 1, 2, \ldots, T$ **do**
  $\hat{y}_t \leftarrow h(\boldsymbol{w}_{t-1}, x_t)$ (Objective function with weights $\boldsymbol{w}_{t-1}$)
  $\hat{s}_t \leftarrow \mathrm{softmax}(\hat{y}_t)$ (Get probability vector)
  if $t \leq t_0$ **or** $c_-(t) < c(\hat{s}_t) < c_+(t)$ **then**
    Based on (x_t, l_t), update $\boldsymbol{w}_{t-1}$ by common optimizers. (e.g. SGD, Adam) .
  else
    Continue
  end if
end for
Output: $\{\boldsymbol{w}_T\}$

This strategy can be easily mixed with other strategies (Query by Committee (Seung et al., 1992), Query by Bagging and Boosting (Abe & Mamitsuka, 1998)) to achieve better performance. Here, we take Query by Committee as an example (see algorithm 5).

Algorithm 5 Combination of query by self and other committee-based algorithm.

Input: $\{\boldsymbol{w}_0\}$ (default initial weight vector), $\{(x_t, l_t)\}_{0\leq t\leq t_0}$(dataset of labeled samples), $\{x_t\}_{t_0+1\leq t\leq T}$ (randomly chosen unlabeled input samples).
for $t = 1, 2, \ldots, T$ **do**
  $\hat{y}_t \leftarrow h(\boldsymbol{w}_{t-1}, x_t)$ (Objective function with weights $\boldsymbol{w}_{t-1}$)
  $\hat{s}_t \leftarrow \mathrm{softmax}(\hat{y}_t)$ (Get probability vector)
  $q_t \leftarrow$ Uncertainty of Committee(x_t)
  if $t \leq t_0$ **or** $c_-(t) < c(\hat{s}_t) + q_t < c_+(t)$ **then**
    Get data (x_t, l_t) via dataset ($t \leq t_0$) or teacher's labeling ($t \geq t_0$).
  else
    Continue
  end if
  Based on (x_t, l_t), update $\boldsymbol{w}_{t-1}$ by common optimizers. (e.g. SGD, Adam)
end for
Output: $\{\boldsymbol{w}_T\}$

4 Theoretical Analysis

In this section, we first give an asymptotic analysis on $\mathbf{P}_t$ and then establish the connection between recursive least square problem and Kalman filter. Using probability model, we are lucky enough to find a direct relation between $\mathcal{I}(\theta)$ and $\mathbf{P}_t$.

Theorem 1 *Assuming components of H_t are independent and subject to identical distribution with mean 0 and variance σ^2, we have*

$$\lim_{t\to\infty} \mathbf{P}_t \overset{a.s.}{=\!=} \lim_{t\to\infty} \frac{\mathbf{I}}{\sigma^2 S(t)}, S(t) \sim \mathcal{O}(t)$$

where $S(t) := \sum_{k=1}^{t} \alpha_k^2 \alpha_t^{-2}$, $\alpha_t := \Pi_{i=1}^{t} \lambda_i^{-1/2}$, and $\alpha_0 := 1$.

Based on basic KF theory, we obtain

$$\begin{aligned}
\mathbf{P}_t &= \lambda_t^{-1}\mathbf{P}_{t-1} - \lambda_t^{-2}\mathbf{P}_{t-1}\mathbf{H}_t^{\mathsf{T}}\mathbf{a}_t^{-1}\mathbf{H}_t\mathbf{P}_{t-1}, \\
\mathbf{a}_t &= \lambda_t^{-1}\mathbf{H}_t^{\mathsf{T}}\mathbf{P}_{t-1}\mathbf{H}_t + 1 = \mathbb{E}[\epsilon_t^2]\alpha_t^2, \\
\lambda_t &= 1 - (1-\lambda_1)\nu^{t-1}, \\
\mathbf{P}_t &= \mathbb{E}[\tilde{\boldsymbol{w}}_t\tilde{\boldsymbol{w}}_t^{\mathsf{T}}]\alpha_t^2,
\end{aligned}$$

and

$$\mathbf{P}_t^{-1} = \lambda_t \mathbf{P}_{t-1}^{-1} + \mathbf{H}_t \mathbf{H}_t^{\mathsf{T}}$$

by using Woodbury matrix identity, where $\tilde{\boldsymbol{w}}_t = \boldsymbol{w} - \boldsymbol{w}_t$, weights error covariance matrix $\mathbb{E}[\tilde{\boldsymbol{w}}_t \tilde{\boldsymbol{w}}_t^{\mathsf{T}}] = (\mathbf{P}_0^{-1} + \sum_{i=1}^{t} \alpha_i^2 \mathbf{H}_i \mathbf{H}_i^{\mathsf{T}})^{-1}$. In the following experiments, we assume components of $\mathbf{H}_i$ are independent and subject to identical distribution with mean 0 and variance σ^2. So, the covariance matrix of $\mathbf{H}_i$ is $\sigma^2 \mathbf{I}$ and $\mathbf{P}_0 = \mathbf{I}$. Hence

$$\mathbb{E}[\mathbf{P}_t^{-1}] = \mathbf{I}\alpha_t^{-2} + \sum_{k=1}^{t} \alpha_k^2 \alpha_t^{-2} \sigma^2 \mathbf{I}.$$

Using q-Pochhammer symbol, we find

$$\lim_{t\to\infty} \alpha_t^2 = \Pi_{i=0}^{\infty}(1 - (1-\lambda_1)\nu^i) = ((1-\lambda_1);\nu)_\infty = \alpha$$

exists. Therefore, $S(t) := \sum_{k=1}^{t} \alpha_k^2 \alpha_t^{-2}$ of order $\mathcal{O}(t)$. According to the law of large numbers, we get

$$\lim_{t\to\infty} \frac{\mathbf{P}_t^{-1}}{S(t)} \xrightarrow{a.s.} \sigma^2 \mathbf{I},$$

i.e.

$$\lim_{t\to\infty} \mathbf{P}_t \overset{a.s.}{=\!=} \lim_{t\to\infty} \frac{\mathbf{I}}{\sigma^2 S(t)}.$$

Theorem 2 *Consider the recursive least square problem*

$$\hat{\theta}_t = \arg\min_{\theta} L(\theta) = \arg\min_{\theta} \frac{1}{2}(\theta^{\mathsf{T}} \mathbf{P}_1^{-1} \theta \alpha_t^{-2} + \Sigma_{k=1}^{t} \|y_k - \hat{y}_\theta(x_k)\|_2^2 \alpha_k^2 \alpha_t^{-2}).$$

If we do Taylor expansion as below

$$\hat{y}_{\theta_t}(x_k) \approx \hat{y}_{\theta_{k-1}}(x_k) + H_k^{\mathsf{T}}(\theta_t - \theta_{k-1}), \quad \forall k \in \{1, 2, \ldots, t\}$$

where $H_k := \nabla_\theta \hat{y}_{\theta_{k-1}}(x_k)$*, we obtain an estimator*

$$\hat{\theta}_t = (\mathbf{P}_1^{-1}\alpha_t^{-2} + \Sigma_{k=1}^{t} H_k H_k^{\mathsf{T}} \alpha_k^2 \alpha_t^{-2})^{-1} (\Sigma_{k=1}^{t} \alpha_k^2 \alpha_t^{-2} H_k (y_k - \hat{y}_{\theta_{k-1}}(x_k) + H_k^{\mathsf{T}} \theta_{k-1})),$$

and we have

$$\mathbf{P}_t = (\mathbf{P}_1^{-1}\alpha_t^{-2} + \Sigma_{k=1}^{t} H_k H_k^{\mathsf{T}} \alpha_k^2 \alpha_t^{-2})^{-1}.$$

Theorem 3 *Set* $\mathbb{S} := \{(x_1, y_1), (x_2, y_2), \ldots, x_t, y_t\}$ *and assume* $\forall k \in \{1, 2, \ldots, t\}, y_k = h(x_k, \theta) + \epsilon_k, \epsilon_k$ *are independent and subject to* $\mathcal{N}(0, \mathbf{I}\alpha_k^2 \alpha_t^{-2})$ *and prior distribution of* θ *is* $\mathcal{N}(0, \mathbf{P}_1^{-1}\alpha_t^{-2})$*. Then*

$$\mathbb{P}_\theta(\mathbb{S}) = Z e^{-L(\theta)}.$$

If

$$\frac{\partial^2}{2\partial\theta_i \partial\theta_j}(\hat{y}_\theta(x) - y)^2 \approx \frac{\partial}{\partial\theta_i}\hat{y}_\theta(x) \frac{\partial}{\partial\theta_j}\hat{y}_\theta(x),$$

i.e. the second derivative term can be omitted, we have the following estimate

$$\mathcal{I}_{\mathbb{S}}(\theta) = Z\mathbb{E}_{\mathbb{S}}[\mathbf{P}_t^{-1} \big|\, \theta].$$

This theorem can be proved directly as following

$$\begin{aligned}
\mathcal{I}_{\mathbb{S}}(\theta)_{i,j} &= -\mathbb{E}\left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}(\log \mathbb{P}_\theta(\mathbb{S}))\middle|\, \theta\right] \\
&= Z\mathbb{E}\left[\frac{\partial^2}{\partial\theta_i\partial\theta_j} L(\theta)\middle|\, \theta\right] \\
&= Z\mathbb{E}\left[(\mathbf{P}_1^{-1})_{i,j}\alpha_t^{-2} + \Sigma_{k=1}^{t} H_{k,i} H_{k,j} \alpha_k^2 \alpha_t^{-2}\middle|\, \theta\right].
\end{aligned}$$

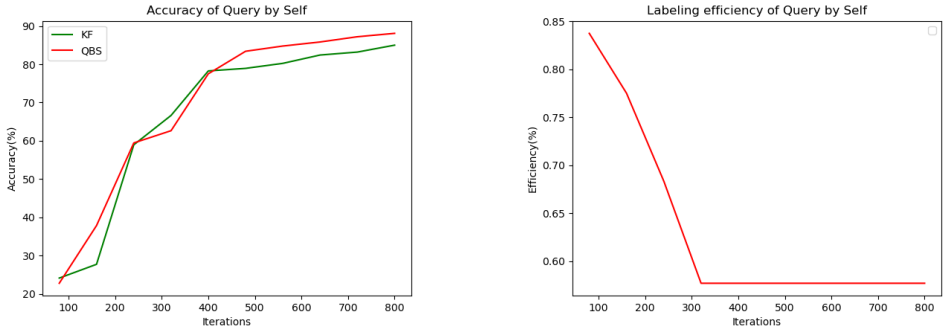

Figure 1: The left figure is the accuracy of query by self after certain number of iterations. Compared with no-strategy Kalman filtering, it achieves a advantage of $5 \sim 10$ percentage points during most of time. This result is stable. The right figure is the labeling efficiency of query by self. It is the ratio of the number of unlabeled samples which are chosen by query by self but the current model can predict right to the number of unlabeled sample chosen by the strategy.

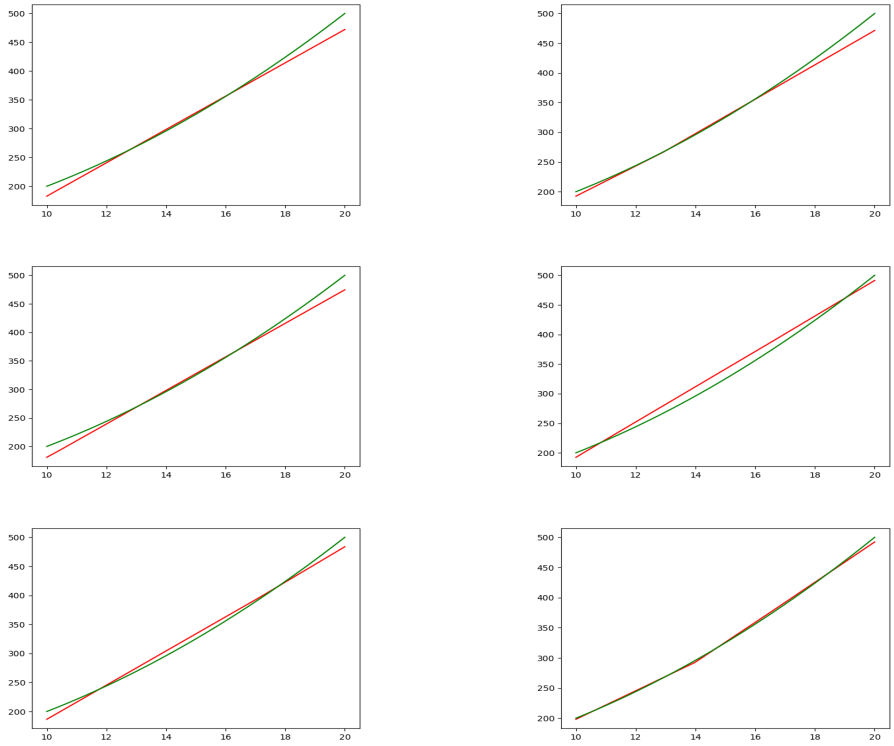

Figure 2: These figures are graphs of $y(x) = x^2 + 100$, $x \in [10, 20]$ (green) and its approximation (red). The first row, the second and the third are the results after 300, 400, and 500 iterations respectively. The left column is the fitting result of no-strategy Kalman filter and the right column is the result of query by self.

5 EXPERIMENTS

Generally, Kalman filtering as an optimizer is usually use on small neural network for its $\mathcal{O}(N^2)$ computational complexity, where N is the number of trainable parameters. Therefore, we evaluate

our strategy on various small neural networks like logistic regression and single output regression. Using computer-generated and MNIST datasets, we demonstrate query by self efficiently selects out suitable unlabeled data for teacher to label and thus enhance the accuracy, reduce the training time, and improve the generalization performance. To verify the practical utility of query by self, the hyper-parameters in our algorithm are not fine-tuned.

5.1 Logistic Regression.

We realize handwritten digit recognition by multi-class logistic regression on a one-layer neural network using the MNIST dataset. The input of this network is $28 \times 28 = 784$ dimension image vectors and the output is 10 dimension which will be activated by softmax function. Then output the index of the largest softmax output as the predict. We compare the accuracy of the model containing query by self and another directly trained with Kalman filter as its optimizer. As shown in Figure 4, we discover query by self is able to enhance accuracy compared to only using Kalman filter as an optimizer. After the same number of iterations, accuracy of query by self is 5$\sim$10 percentage points higher than that of no-strategy kalman filtering. This indeed means higher training efficiency.

As we can see in right sub-figure of figure 4, the labeling efficiency converges around 58%, which is very high compare to the model's error rate 10% or so. This means query by self is $58/10 = 5.8$ times efficiency than no-strategy Kalman filter during training process.

5.2 Fitting 1 Dimensional Continuous Function.

This experiment is suitable to testify to the ability of query by self to improve generalization performance, since we can see the fitting result in form of a visible graph. To this end, we should not choose big network due to its strong ability to approximate such a simple function that we can not differentiate which approximation enjoys a better generalization performance from its graph. Considering this, we use a model with just a hidden layer which has only 64 neurons. Our target function is $y(x) = x^2 + 100, x \in [10, 20]$.

The result on the right sides is better than its counterpart on the left. These figures show our strategy query by self can protect neural network from over-fitting and enjoy a excellent generalization performance.

6 Conclusion

We propose a novel active learning query strategy, query by self. It does not relay on committee to give an estimate of variance of output. Besides, we theoretically explain query by self strategy from the perspective of entropy and information theory. Experimentally, it can accelerate training process, improving generalization performance and helping us to save money and time. We hope this strategy can become a powerful tool to deal with practical problem. In fact, popular neural network in these days are deep neural networks, to which more attempts should be devoted and do related research on it further.

REFERENCES

Naoki Abe and Hiroshi Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pp. 19, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1558605568.

Usman Ahmed, Suresh Kumar Mukhiya, Gautam Srivastava, Yngve Lamo, and Jerry Chun-Wei Lin. Attention-based deep entropy active learning using lexical algorithm for mental health treatment. *Frontiers in Psychology*, 12:642347, 2021.

Dana Angluin. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

Les Atlas, David Cohn, and Richard Ladner. Training connectionist networks with queries and selective sampling. *Advances in neural information processing systems*, 2, 1989.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pp. 273–304, 1995.

Charles K Chui, Guanrong Chen, et al. *Kalman filtering*. Springer, 2017.

David Cohn. Neural network exploration using optimal experiment design. *Advances in neural information processing systems*, 6, 1993.

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

Shi Dong. Multi class svm algorithm with active learning for network traffic classification. *Expert Systems with Applications*, 176:114885, 2021. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2021.114885. URL https://www.sciencedirect.com/science/article/pii/S0957417421003262.

David E Graff, Eugene I Shakhnovich, and Connor W Coley. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chemical science*, 12(22):7866–7881, 2021.

Simon S Haykin and Simon S Haykin. *Kalman filtering and neural networks*, volume 284. Wiley Online Library, 2001.

F. Heimes. Extended kalman filter neural network training: experimental results and algorithm improvements. In *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*, volume 2, pp. 1639–1644 vol.2, 1998. doi: 10.1109/ICSMC.1998.728124.

Kevin Maik Jablonka, Giriprasad Melpatti Jothiappan, Shefang Wang, Berend Smit, and Brian Yoo. Bias free multiobjective active learning for materials design and discovery. *Nature communications*, 12(1):1–10, 2021.

Rudolf Emil Kalman et al. Contributions to the theory of optimal control. *Bol. soc. mat. mexicana*, 5(2):102–119, 1960.

Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Physical review*, 140(4A):A1133, 1965.

David D Lewis. A sequential algorithm for training text classifiers: Corrigendum and additional data. In *Acm Sigir Forum*, volume 29, pp. 13–19. ACM New York, NY, USA, 1995.

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

S. Murtuza and S.F. Chorian. Node decoupled extended kalman filter based learning algorithm for neural networks. In *Proceedings of 1994 9th IEEE International Symposium on Intelligent Control*, pp. 364–369, 1994. doi: 10.1109/ISIC.1994.367790.

Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. 2009.

Evgeny V Podryabinkin and Alexander V Shapeev. Active learning of linearly parametrized interatomic potentials. *Computational Materials Science*, 140:171–180, 2017.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, 2:441–448, 2001.

Mark J Schervish. *Theory of statistics*. Springer Science & Business Media, 2012.

Burr Settles. Active learning literature survey. 2009.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *proceedings of the 2008 conference on empirical methods in natural language processing*, pp. 1070–1079, 2008.

Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. *Advances in neural information processing systems*, 20, 2007.

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294, 1992.

Gerald L Smith, Stanley F Schmidt, and Leonard A McGee. *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle*. National Aeronautics and Space Administration, 1962.

Justin S Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev, and Adrian E Roitberg. Less is more: Sampling chemical space with active learning. *The Journal of chemical physics*, 148(24): 241733, 2018.

Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

Yuzhi Zhang, Haidi Wang, Weijie Chen, Jinzhe Zeng, Linfeng Zhang, Han Wang, and E Weinan. Dp-gen: A concurrent learning platform for the generation of reliable deep learning based potential energy models. *Computer Physics Communications*, 253:107206, 2020.