

Meta-Learning PAC-Bayes Priors in Model Averaging

Yimin Huang, Weiran Huang, Liang Li, Zhenguo Li

Huawei Noah's Ark Lab

{yimin.huang, weiran.huang, liliang103, Li.Zhenguo}@huawei.com

Abstract

Nowadays model uncertainty has become one of the most important problems in both academia and industry. In this paper, we mainly consider the scenario in which we have a common model set used for model averaging instead of selecting a single final model via a model selection procedure to account for this model's uncertainty in order to improve reliability and accuracy of inferences. Here one main challenge is to learn the prior over the model set. To tackle this problem, we propose two data-based algorithms to get proper priors for model averaging. One is for meta-learner, the analysts should use historical similar tasks to extract the information about the prior. The other one is for base-learner, a subsampling method is used to deal with the data step by step. Theoretically, an upper bound of risk for our algorithm is presented to guarantee the performance of the worst situation. In practice, both methods perform well in simulations and real data studies, especially with poor quality data.

1 Introduction

It is very common in practice that the distributions generating the observed data are described more adequately by multiple models. A standard procedure to make the inference is that, according to some criteria, such as model predictive ability, model fitting ability and many different information criteria a best model is chosen and assumed as the true model. After selection, all the inferences and conclusions are made based on the assumption.

However, the drawbacks of this approach exist. The selection of one particular model may lead to riskier decisions since it ignores the model uncertainty. In other words, if we choose a wrong model, the consequence will be disastrous. Moral-Benito (2015) already pointed out the concern, "From a pure empirical viewpoint, model uncertainty represents a concern because estimates may well depend on the particular model considered." Therefore, combining multiple models to reduce the model uncertainty is very desirable.

As an alternative strategy, combining multiple models which is called model averaging enables researchers to draw

conclusions based on the whole universe of candidate models. In particular, researchers estimate all the candidate models and then compute a weighted average of all the estimates. There are two different approaches to model averaging in the literature, including Frequentist Model Averaging (FMA) and Bayesian Model Averaging (BMA). Frequentist approaches focus on improving prediction and use weighted mean of estimates from different models while Bayesian approaches focus on the probability that a model is true and consider priors and posteriors for different models.

The FMA approach does not consider priors, so the corresponding estimators depend solely on data. For its simplicity, the FMA approach has received some attention over last decade. See Yang (2001), Leung and Barron (2006), Hansen (2007) and a detailed review Wang, Zhang, and Zou (2009) for reference.

Leamer (1978) suggested to use Bayesian inference to reduce the model uncertainty as a framework and pointed out the importance of the fragility of regression analysis to arbitrary decisions about the choice of control variables. Bayesian Model Averaging considers model uncertainty through the prior distribution. The model posteriors are obtained by Bayes' theorem, and therefore allow for combined estimation and prediction. Compared with the FMA approaches, there are a huge literature on the use of BMA in statistics.

Influenced by Leamer (1978), most works were concentrated on the linear models only. Raftery (1996) extended in generalized linear models by providing a straightforward approximation. For more details, refer to a landmark review Moral-Benito (2015) on BMA.

The Bayesian approaches have the advantage of using arbitrary domain knowledge through a proper prior. However, as commented by Hjort and Claeskens (2003), how to set prior probabilities and how to deal with the priors when they are in conflict with each other are still problems. The PAC-Bayes framework, first formulated by McAllester (1999), was proposed to take the priors into account. In the beginning, most researches assumed that loss functions were bounded. For detailed information, see Catoni (2007). For unbounded loss, Catoni (2004) provided a result under exponential moment assumptions. In last

decade, it has been widely developed. Different types of PAC-Bayes bounds were presented under various assumptions, for example, Seeger (2002), Seldin and Tishby (2010), Guedj, Alquier, and others (2013), Alquier, Ridgway, and Chopin (2016), Grünwald and Mehta (2016), Catoni (2016), Lugosi, Mendelson, and others (2019), and Alquier and Guedj (2018). And, many distribution-dependent priors are used to derive tighter PAC-Bayes bounds such as Lever, Laviolette, and Shawe-Taylor (2013), Oneto, Anguita, and Ridella (2016), Dziugaite and Roy (2018) and Rivasplata et al. (2018). Here, we must distinguish between obtaining the tighter bounds by distribution-dependent priors and using part of the data to meta-learn a prior and the rest to learn the function.

Note that for getting the posterior distribution of the weights, Ye, Yang, and Yang (2016) gave a method without choosing a proper prior. For meta-learning the prior, some meta-learners (Finn, Abbeel, and Levine 2017; Li et al. 2017; Amit and Meir 2018) are limited to their use of gradient. Amit and Meir (2018) provided an extended PAC-Bayes bound for learning the proper priors. Nevertheless, it involved reusing of the data which increased the probability of overfitting, and they gave an implementation only for the normal distribution of the weights.

In this paper, we propose a specific risk bound under our settings and two data-based methods for adjusting the priors in PAC-Bayes framework. And, two practical algorithms are given accordingly. The main contributions of this work are the following. First, when the historical data existed, we use similar old tasks to extract the mutual knowledge with the current task for adjusting the priors. Second, a sequential batch sampling method is proposed to deal with the base-learner for learning posterior by subsampling with the rules made by researchers. Third, two theoretical risk bounds are provided for these two situations respectively. Fourth, empirical demonstration shows that the proposed meta-methods have excellent performances in the numerical studies.

The remainder of this paper is organized as follows. In Section 2, an upper bound for the averaging model and a practical historical data related algorithm are established for obtaining a better prior. In case that there is no historical data, Section 3 proposes another method called a sequential batch sampling algorithm to adjust the prior step by step. Illustrative simulations including regression and classification tasks given in Section 4 show that our algorithms will lead to more effective prediction. We further apply the proposed methods to two real datasets and confirm the higher prediction accuracy of the minimizing risk bound method. Some proofs of theories are delegated to the full paper (Huang et al. 2019).

2 Learning the Prior in Meta-Learner

In a traditional supervised learning task, the learner needs to find an optimal *model* (or hypothesis) to fit the data, and then uses the learned model to make predictions. In the Bayesian approach, various models are allowed to fit the data. In particular, the learner needs to learn an optimal model *distribution*

over the candidate models, and then uses the learned model distribution to make predictions.

More specifically, in a supervised learning task, we are given a set $S = \{(x_i, y_i)\}_{i=1}^n$ of i.i.d. samples drawn from an unknown distribution D over $\mathcal{X} \times \mathcal{Y}$, i.e., $(x_i, y_i) \sim D$. The goal is to find a model h in the candidate model set \mathcal{H} , a set of functions mapping features (feature vector) to responses, that minimizes the expected loss function $\mathbb{E}_{(x,y) \sim D} L(h, x, y)$, where L is a bounded loss function. Without loss of generality, we assume L is bounded by $[0, 1]$. Note that this assumption is often used in the beginning of the PAC-Bayes framework. This paper uses McAllester’s bound, so this assumption is necessary. Note that our procedure can use other, possibly tighter bounds. In other words, under different regularization conditions, it can be replaced by many other PAC-type bounds as long as the assumption matches the corresponding PAC bound. In the Bayesian framework, a distribution Q over \mathcal{H} is the purpose instead of searching a specific optimal model $h \in \mathcal{H}$. Therefore, the goal turns to finding the optimal model distribution Q , which minimizes $\mathbb{E}_{h \sim Q} \mathbb{E}_{(x,y) \sim D} L(h, x, y)$. Then one could weighted average the models over \mathcal{H} to make predictions, namely, $\hat{y} = \mathbb{E}_{h \sim Q} h(x)$. More generally, we further assume that the candidate model set \mathcal{H} consists of K classes of models $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_K$ with $\mathcal{H} = \bigcup_{k=1}^K \mathcal{M}_k$. Each model class \mathcal{M}_k is associated with a probability w_k , and for each model class \mathcal{M}_k , there is a distribution Q_k over \mathcal{M}_k . For example, a model class \mathcal{M}_k could be a group of models obtained from the Lasso method, and the hyper-parameter λ in Lasso follows a distribution Q_k . Another common example is that \mathcal{M}_k is a group of neural networks with a certain architecture, and the hyperparameters of neural networks follow a joint distribution Q_k . In this way, the total distribution over \mathcal{H} can be written as $\xi = (\mathbf{w}, Q_1, \dots, Q_K)$, where \mathbf{w} consists of w_1, \dots, w_K with $\|\mathbf{w}\|_1 = 1$. The goal of the learning task is to find an optimal distribution ξ , the posterior of h , which minimizes the expected risk $R(\xi, D) := \mathbb{E}_{h \sim \xi} \mathbb{E}_{(x,y) \sim D} L(h, x, y)$, and then the prediction is made by $\hat{y} = \mathbb{E}_{h \sim \xi} h(x) = \sum_{k=1}^K [w_k \cdot \mathbb{E}_{h \sim Q_k} h(x)]$.

Since sample distribution D is unknown, the expected risk $R(\xi, D)$ cannot be computed directly. Therefore, it is usually approximated by the empirical risk $\hat{R}(\xi, S) := \mathbb{E}_{h \sim \xi} \sum_{(x_i, y_i) \in S} L(h, x_i, y_i) / |S|$ in practice, and ξ is learned by minimizing the empirical risk $\hat{R}(\xi, S)$. When the sample size is large enough, it would be a good approximation. However, in many situations, we do not have so much data, which may lead to large difference between them. Thus, using the empirical risk $\hat{R}(\xi, S)$ to approximate the expected risk $R(\xi, D)$ is not appropriate any longer.

We first study the difference between the empirical risk $\hat{R}(\xi, S)$ and the expected risk $R(\xi, D)$. Based on the literature (McAllester 1999), we can obtain an upper bound of their difference which is stated as the following lemma.

Lemma 1 *Let ξ^0 be a prior distribution over \mathcal{H} that must be chosen before observing the samples, and let $\delta \in (0, 1)$. Then with probability at least $1 - \delta$, the following inequality*

holds for all posterior distributions ξ over \mathcal{H} ,

$$R(\xi, D) \leq \hat{R}(\xi, S) + \sqrt{\frac{\text{KL}(\mathbf{w}||\mathbf{w}^0) + \sum_{k=1}^K w_k \text{KL}(Q_k||Q_k^0) + \ln \frac{n}{\delta}}{2(n-1)}}, \quad (1)$$

where n is the cardinality of sample set S , and $\text{KL}(\cdot||\cdot)$ is the Kullback-Leibler (KL) divergence between two distributions¹.

According to the above lemma, it is clear that only when the sample size n is large, the difference $R(\xi, D) - \hat{R}(\xi, S)$ can be guaranteed to be small. Thus, minimizing $\hat{R}(\xi, S)$ may not lead to the minimizer of $R(\xi, D)$, which matches our intuition. To avoid the risk of the approximation, one can minimize the upper bound of the expected risk $R(\xi, D)$ in stead of using the empirical risk $\hat{R}(\xi, S)$ as an approximation. In particular, we denote the right hand side of Eq.(1) by $\bar{R}(\xi, \xi^0, S)$. Then one can learn the model distribution ξ by minimizing $\bar{R}(\xi, \xi^0, S)$. Intuitively, such choice of ξ for the learning task makes the worst case best.

Lemma 1 also indicates that the prior ξ^0 plays an important role. Since the choice of ξ balances the tradeoff between the empirical risk $\hat{R}(\xi, S)$ and the regularization term, if the prior ξ^0 is far away from the true optimal model distribution ξ^* , the posterior ξ will also be bad. The best situation for optimizing the posterior ξ is that the prior ξ^0 exactly equals to the true optimal model distribution ξ^* . Then, the regularization term disappears. In other words, if there is a good prior ξ^0 which is close to ξ^* , the upper bound $\bar{R}(\xi, \xi^0, S)$ will be small. However, without any prior knowledge, one can only use data to help obtain a better prior. The naive method is directly using the non-informative prior as ξ^0 for minimizing $\bar{R}(\xi, \xi^0, S)$ to get the posterior ξ .

When the extra data of historical tasks has been collected, the learner has the chance to learn a good prior for more reliable inferences. In order to get a good prior, it is helpful to extract the mutual knowledge from similar tasks. Figure 1 schematically illustrates this process. In particular, there are m sample tasks T_1, \dots, T_m i.i.d. generated from an unknown task distribution τ . For each sample task T_i , a sample set S_i with n_i samples is generated from an unknown distribution D_i . Without ambiguity, we use notation $\xi(\xi^0, S)$ to denote the posterior under the prior ξ^0 after observing the sample set S . Note that, the proposed meta-learner still works if the base-learner of getting the posterior is replaced by other popular methods. The quality of a prior ξ^0 is measured by $\mathbb{E}_{D_i \sim \tau} \mathbb{E}_{S_i \sim D_i^{n_i}} R(\xi(\xi^0, S_i), D_i)$. Thus, the expected loss we want to minimize is

$$R(\xi^0, \tau) = \mathbb{E}_{D_i \sim \tau} \mathbb{E}_{S_i \sim D_i^{n_i}} R(\xi(\xi^0, S_i), D_i).$$

The above expected risk cannot be computed directly for the unknown distribution D_i , thus the following empirical risk is used to estimate it:

$$\hat{R}(\xi^0, S_1, \dots, S_m) = \frac{1}{m} \sum_{i=1}^m \hat{R}(\xi(\xi^0, S_i^{train}), S_i^{validation}),$$

¹ $\text{KL}(P||P^0)$ is defined as $\mathbb{E}_{x \sim P} \ln \frac{P(x)}{P^0(x)}$.

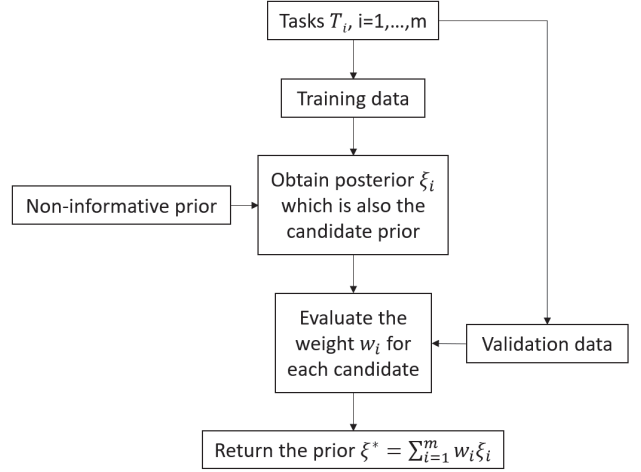


Figure 1: Illustration of the meta-learning process.

where each sample set S_i is divided into a training set S_i^{train} and a validation set $S_i^{validation}$.

Consider the regression setting for task T . Suppose the true model is

$$y_T = f_T(x_T) + \sigma_T(x_T) \cdot \varepsilon_T,$$

where $f_T: \mathbb{R}^d \rightarrow \mathbb{R}$ is the function to be learned, the error term ε_T is assumed to be independent of x_T and has a known probability density $q(t), t \in \mathbb{R}$ with mean 0 and a finite variance. The unknown function $\sigma_T(x_T)$ controls the variance of the error at $X = x_T$. There are n_T i.i.d. samples $\{(x_{T,i}, y_{T,i})\}_{i=1}^{n_T}$ drawn from an unknown joint distribution of (x_T, y_T) . Assume that there is a candidate model set \mathcal{H} , each of which is a function mapping features (feature vector) to response, i.e., $h \in \mathcal{H}: \mathbb{R}^d \rightarrow \mathbb{R}$. To take the information of the old tasks, which can reflect the importance of each $h \in \mathcal{H}$, the following Algorithm 1 (meta-learner) is proposed.

This algorithm is based on the cross-validation framework. First, using T_i to obtain the candidate priors ξ_i by any Bayesian procedure, for example minimizing the PAC bound introduced in Lemma 1 with non-informative prior. Cross-validation determines the importance of the priors. The j -th task is divided into two parts randomly. The first part is used to learn the posterior with the prior ξ_i . The second part is to evaluate the performance of the posterior by its likelihood function. This evaluation is inspired by Raftery (1995). To simplify the determination of the weights, Raftery (1995) proposed a frequentist approach to BMA. The Bayes' theorem was replaced by the Schwarz asymptotic approximation which could be viewed as using maximized likelihood function as the weights of the candidate models. The $\hat{\sigma}$ on the denominator of E_j^i makes the weight larger if the model is accurate. This procedure repeats many times for each pair (i, j) . Their averages reveal the importance of the priors. In the end, the ξ^* is obtained by weighted averaging them all. The property of this algorithm can be guaranteed by Lemma 2.

The following regularization conditions are assumed for

Algorithm 1 Historical Data Related Algorithm

- 1: **for** $i = 1$ to m **do**
 - 2: Using T_i to obtain ξ_i by the same Bayesian procedure in base-learner with non-informative prior.
 - 3: **end for**
 - 4: **for** $i = 1$ to m **do**
 - 5: Randomly split the data S_i into two parts
 $S_{i,n'_i}^{(1)} = (x_{i,\alpha}, y_{i,\alpha})_{\alpha=1}^{n'_i}$ for training and
 $S_{i,n'_i}^{(2)} = (x_{i,\alpha}, y_{i,\alpha})_{\alpha=n'_i+1}^{n_i}$ for validation.
 - 6: **for each** $j \neq i$ **do**
 - 7: Obtain estimates $\hat{f}_{j,n'_i}(x, S_{i,n'_i}^{(1)})$, $\hat{\sigma}_{j,n'_i}(x, S_{i,n'_i}^{(1)})$ with prior ξ_j .
 - 8: Evaluate predictions on $S_{i,n'_i}^{(2)}$ and compute

$$E_j^i = \frac{\prod_{\alpha=n'_i+1}^{n_i} q\left(\frac{y_{i,\alpha} - \hat{f}_{j,n'_i}(x_{i,\alpha})}{\hat{\sigma}_{j,n'_i}(x_{i,\alpha})}\right)}{\prod_{\alpha=n'_i+1}^{n_i} \hat{\sigma}_{j,n'_i}(x_{i,\alpha})}.$$
 - 9: **end for**
 - 10: **end for**
 - 11: Repeat the random data segmentation more times and average the weights E_j^i after normalization to get $w_j^{(i)} (j \neq i)$.
 - 12: Average all the $w_j^{(i)} (j \neq i)$ from $i = 1$ to m to obtain the final weights w_j .
 - 13: The prior learned for a new task is $\xi^* = \sum_{i=1}^m w_i \xi_i$.
-

the results. First, q is assumed to be a known distribution with 0 and variance 1.

(C1) The functions f and σ are uniformly bounded, i.e., $\sup_x |f(x)| \leq A < \infty$ and $0 < m \leq \sigma(x) \leq M < \infty$ for constants A, m and M .

(C2) The error distribution q satisfies that for each $0 < s_0 < 1$ and $t_0 > 0$, there exists a constant B such that

$$\int q(x) \ln \frac{q(x)}{\frac{1}{s} q\left(\frac{x-t}{s}\right)} \mu(dx) \leq B((1-s)^2 + t^2)$$

for all $s_0 \leq s \leq s_0^{-1}$ and $-t_0 \leq t \leq t_0$.

(C3) The risks of the estimators for approximating f and σ^2 decrease as the sample size increases.

For the condition (C1), note that, when we deal with k -way classification tasks, the responses belong to $\{1, 2, \dots, k\}$ which is bounded obviously. Moreover, if the input space is a finite region which often happens in real datasets, most common functions are bounded uniformly. The constants A, m, M are involved in the derivation of the risk bounds, but they can be unknown in practice when we implement the Algorithm 1. The condition (C2) is satisfied by Gaussian, t (with the degree of freedom larger than two), double-exponential, and so on. The condition (C3) usually

holds for a good estimating procedure, like consistent estimators. An estimator is called consistent if the expected risk tends to zero when experimental size tends to infinity.

Lemma 2 Assume three regularization conditions are satisfied. The loss function $L(h, x, y) = |y - h(x)|^2$ and σ_{T_i} is known. Then, the combined prior ξ^* as given above satisfies

$$R(\xi^*, \tau) \leq \inf_j \left(\frac{C_1}{\sum_{i \neq j} (n_i - n'_i)} + \frac{C_2}{\sum_{i \neq j} (n_i - n'_i)} \sum_{i \neq j} (n_i - n'_i) R(\xi_j, D_i) \right),$$

with probability at least $1 - \delta$, where the constant C_1, C_2 depend on the regularization conditions.

Note that we assume a known σ_{T_i} just for simplifying the expression. It has a more general version for unknown σ_{T_i} . The proof is given briefly with unknown σ_{T_i} in Supplementary Materials.

In this general prove, it can be seen that, (i) Variance estimation is also important for the Algorithm 1. Even if a procedure estimates f_T very well, a bad estimator of σ_T can substantially reduce its weight in the final estimator. (ii) Under the condition (C3), the risks of a good procedure for estimating f_T and σ_T usually decrease as the sample size increases. The influence of the number of testing points n'_i is quite clear. Smaller n'_i decreases the first penalty term but increases the main terms that involve the risks of each j . (iii) Lemma 2 reveals the vital property that if one alternative model is consistent, the combined model will also have the consistency.

If Algorithm 1 is used to obtain the prior from multi-tasks, we could get the following theorem theoretically by simply combining Lemmas 1 and 2. See Huang et al. (2019) for the detailed proofs.

Theorem 1 Assume conditions (C1), (C2) and (C3) are satisfied. The loss function $L(h, x, y) = |y - h(x)|^2$ and σ_{T_i} is known. Then, the combined posterior ξ^* as given above satisfies

$$R(\xi^*, \tau) \leq \inf_j \left(\frac{C_1}{\sum_{i \neq j} (n_i - n'_i)} + \frac{C_2}{\sum_{i \neq j} (n_i - n'_i)} \sum_{i \neq j} (n_i - n'_i) \left[\hat{R}(\xi_j^*, S_{i,n'_i}^{(2)}) + \sqrt{\frac{\text{KL}(\omega_j^* || \omega_j) + \sum_{k=1}^K \omega_{j,k} \text{KL}(Q_{j,k}^* || Q_{j,k}) + \ln \frac{n_i}{\delta}}{2(n_i - 1)}} \right] \right)$$

with probability at least $1 - \delta$, where the constant C_1, C_2 depend on the regularization conditions, π is the initial prior which should be non-informative prior and ξ_j^* is the minimizer of Eq.(1) with $\xi^0 = \xi_j$ and $S = S_{i,n'_i}^{(1)}$.

The major contribution of Theorem 1 is that it implies a simple consequence on consistency. The penalty for adaptation (the first term in RHS of Theorem 1) is basically of order $1/n$, which is negligible for nonparametric rates. Thus,

Algorithm 2 Sequential Batch Sampling Algorithm

- 1: Obtain a sample set B_1 from the sample space $\mathcal{X} \times \mathcal{Y}$ by a initial space-filling design.
 - 2: Get the posterior ξ_1 based on the sample set B_1 by minimizing the risk bound with non-informative prior.
 - 3: **for** $i = 2$ to b **do**
 - 4: Search next sample set B_i ($|B_i| = n_b$) with the large volatility under the current posterior ξ_{i-1} , i.e., $\hat{v}(\xi_{i-1}, B_i) > \gamma_i$ where γ is a given constant vector.
 - 5: Get the posterior ξ_i based on the sample set B_i by minimizing the risk bound with the prior ξ_{i-1} .
 - 6: **end for**
 - 7: The final posterior is ξ_b .
-

for any bounded regression function, the combined model performs asymptotically as well as any model in the candidate model set \mathcal{H} . The detailed expression refers to Corollary 1 in Yang (2000). Further, even if the underlying true model is not in the candidate set, the combined model may still be able to approach the true model, e.g, there exists a sequence of models in the set approaching an optimal one.

For classification tasks, one-hot response is used which means the i -th response y_i in the model is a vector describing the probability of each class. The loss function still uses $\sum_{i=1}^n \|y_i - \hat{y}_i\|^2$ to maintain consistency with the regression case. The prediction of a new observation x^* is to choose the class with the largest probability in \hat{y}^* . Thus, we do not need to change the condition in Theorem 1, but just change the response to be one-hot for classification. Consequently, the results for regression still hold for classification.

Besides the l_2 risk that we consider, other performance measures (e.g., cross entropy loss which is used in our last experiment with the MNIST dataset or hinge loss) are useful from both theoretical and practical points of view. It is thus of interest to investigate whether similar adaptation procedures exist for other loss functions and if not, what prices are one needs to pay for adaptation, which we leave for future work.

3 Adjusting the Prior in Base-Learner

In this section, we will discuss how to adjust the prior of models if there is no information from extra similar tasks.

In the following, we consider an iterative procedure of adjusting the prior in the base-learner. In each round, the learner can sample the data according to the prior distribution in the current round. Such iterative procedure updates the prior step by step. Ultimately, compared with dealing the whole data at once, this procedure of adjusting prior leads to a smaller upper bound. Moreover, it also gives an opportunity to choose some good sample sets for reducing the volatility of the estimators which is measured by $v(\xi, D) = \mathbb{E}_x \mathbb{E}_h (h(x) - \mathbb{E}_h h(x))^2$. The function $\hat{v}(\xi, B) = \frac{1}{|B|} \sum_{x \in B} \mathbb{E}_h (h(x) - \mathbb{E}_h h(x))^2$ is defined to measure the volatility of the posterior ξ at the sample set B . The complete algorithm for sequential batch sampling is shown in Algorithm 2.

For Algorithm 2, we do not handle the whole data at

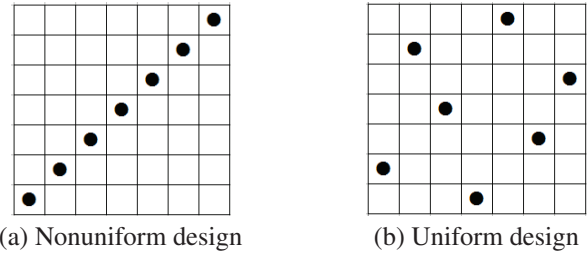


Figure 2: The illustration for uniform space-filling design.

once. Instead, the data is processed in b steps. First, a space-filling design is used as initial experiment points to reduce the probability of overfitting caused by the unbalanced sampling. Traditional space-filling design aims to fill the input space with design points that are as “uniform” as possible in the input space. The uniformity of space-filling design is illustrated in Figure 2. For next steps, uncertain points are needed to be explored. And, the uncertainty is measured by the volatility v . Hence, the batch with large volatility will be chosen. Note that if we set a huge γ , we will just explore a small region of the input space.

The setting of γ refers to (Zhou et al. 2018). However, in practice, it is found that this parameter γ does not matter much, since the results are similar with a wide range of γ . This procedure helps to reduce the variance of the estimator which is proved in (Zhou et al. 2018) by sequential sampling. Furthermore, it also helps to adjust the prior in each step which is called learning the prior. The proposition is stated below.

Proposition 1 For $i = 1, 2, \dots, b$, let $B_i = S$, ξ^* is the minimizer of the RHS of Eq.(1) with non-informative prior ξ^0 and ξ_i obtained by Algorithm 2, then we have $\bar{R}(\xi_b, \xi_{b-1}, S) \leq \bar{R}(\xi^*, \xi^0, S)$.

The above proposition can be understood straightforwardly. First, since we adjust the prior through the data step by step, the final prior ξ_{b-1} is better than the non-informative prior. Consequently, it receives the smaller expected risk. Second, we choose the sample sets sequentially with large volatility to do experiments in order to reduce uncertainty. The performance of SBS (Sequential Batch Sampling) method is also demonstrated in Section 4.

4 Experiments

In this section, some examples are shown to illustrate the procedure of Algorithms 1 and 2. The method of minimizing the upper bound in Lemma 1 with non-informative prior is denoted by RBM (Risk Bound Method). First, we begin with linear regression models which has the same setting in Ye, Yang, and Yang (2016). Hence, their method called SOIL is under the comparison. The optimization for RHS of Eq.(1) in our algorithms is dealt by gradient descend. R package “SOIL” is used to obtain the results of the SOIL method.

4.1 Synthetic Dataset

Example 1 The simulation data $\{(x_i, y_i)\}_{i=1}^n$ is generated for the RBM from the linear model $y_i = 1 + x_i^T \beta + \sigma \epsilon_i$,

where $\varepsilon_i \sim N(0, 1)$, $\sigma \in \{1, 5\}$ and $\mathbf{x}_i \sim N_d(0, \Sigma)$. For each element Σ_{ij} of Σ , $\Sigma_{ij} = \rho^{|i-j|}$ ($i \neq j$) or 1 ($i = j$) with $\rho \in \{0, 0.9\}$. The sequential batch sampling has b steps, and each step uses n/b samples following Algorithm 2.

All the specific settings for parameters are summarized in Table 1, and the confidence level δ in Lemma 1 is set to 0.01. The Mean Squared Prediction Error (MSPE) $\mathbb{E}_x |f(x) - \hat{f}(x)|^2$ and volatility defined in the base-learner are compared. They are obtained by sampling 1000 samples from the same distribution and computing their empirical MSPE $\sum_x |f(x) - \hat{f}(x)|^2 / 10^3$ and volatility. For each model setting with a specific choice of the parameters (ρ, σ) , we repeat 100 times and compute the average empirical value. The comparison among RBM, SOIL and SBS is shown in Tables 2, 3 and 4.

The volatility of SOIL method is the smallest and very close to zero. This phenomenon shows that SOIL is focused on a few models, even just one model when the volatility equals to zero. Consequently, its MSPE is larger than other two methods. SBS as a modification of RBM has similar results with RBM when σ is small. However, when σ is large, SBS performs much better than RBM. In this situation, the information of data is easily covered by big noises. Hence, a good prior which can provide more information is vital for this procedure.

Table 1: Simulation settings of Example 1.

Model	n	d	b	β
1	50	8	5	$(3, 1.5, 0, 0, 2, 0, 0, 0)^T$
2	150	50	5	$(1, 2, 3, 2, 0.75, 0, \dots, 0)^T$
3	50	50	5	$(1, 1/2, 1/3, 1/4, 1/5, 1/6, 0, \dots, 0)^T$

Table 2: Comparison among RBM, SOIL and SBS for Model 1 of Example 1.

Model 1	(ρ, σ)	(0, 1)	(0, 5)	(0.9, 1)	(0.9, 5)
MSPE	RBM	2.03	48.23	3.71	53.83
	SOIL	2.13	53.21	2.17	53.21
	SBS	1.71	14.08	3.25	26.40
Volatility	RBM	1.64	3.47	1.31	0.49
	SOIL	0	0	0.002	0
	SBS	1.61	7.41	1.03	0.42

Next example considers the same comparison but in non-linear models. In last example, the alternative models include the true model, but now the true non-linear model is approximated by many linear models.

Example 2 The simulation data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{50}$ is generated for the RBM from the non-linear models

- $y_i = 1 + \sin(x_{i,1}) + \cos(x_{i,2}) + \varepsilon_i$,
- $y_i = 1 + \sin(x_{i,1} + x_{i,2}) + \varepsilon_i$,

where $\varepsilon_i \sim N(0, 1)$, and $\mathbf{x}_i \sim N_8(0, I)$. The sequential batch sampling has 5 steps, and each step uses 10 samples following Algorithm 2.

Table 3: Comparison among RBM, SOIL and SBS for Model 2 of Example 1.

Model 2	(ρ, σ)	(0, 1)	(0, 5)	(0.9, 1)	(0.9, 5)
MSPE	RBM	1.97	46.26	1.46	35.97
	SOIL	2.01	50.23	1.96	49.78
	SBS	1.93	38.69	1.38	12.92
Volatility	RBM	1.60	2.72	3.38	7.48
	SOIL	0	0	0.001	0.01
	SBS	1.46	8.67	3.35	6.74

Table 4: Comparison among RBM, SOIL and SBS for Model 3 of Example 1.

Model 3	(ρ, σ)	(0, 1)	(0, 5)	(0.9, 1)	(0.9, 5)
MSPE	RBM	1.67	42.06	1.24	38.51
	SOIL	1.99	49.80	1.93	47.99
	SBS	1.65	27.32	1.23	29.44
Volatility	RBM	0.27	1.54	0.74	3.39
	SOIL	0	0	0.02	0.36
	SBS	0.29	0.47	0.77	4.06

The results of Example 2 are listed in Table 5. Mostly, it is similar with the results of Example 1. The difference is that the volatility of SOIL becomes large when the model is completely non-linear. Using linear models to fit non-linear model obviously increases the model uncertainty, since none of the fitting models is correct.

Table 5: Comparison among RBM, SOIL and SBS of Example 2.

		Model 1	Model 2
MSPE	RBM	1.26	1.54
	SOIL	1.42	1.80
	SBS	1.23	1.47
Volatility	RBM	0.1	0.11
	SOIL	0.07	0.02
	SBS	0.11	0.14

The final example is under the situation that the data has been already collected. Hence, we cannot use the SBS method to get the data. However, we have the extra data of many old similar tasks. In particular, we have the data of Example 1. Now, the new task is to fit a new model.

Example 3 The data of Example 1 with $(\rho, \sigma) = (0, 1)$ is given. The new task data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{20}$ is generated from the linear model $y_i = 1 + \mathbf{x}_i^T \beta + \sigma \varepsilon_i$, where $\varepsilon_i \sim N(0, 1)$, $\sigma \in \{1, 2, 3, 4, 5\}$, $\beta = \{1, -1, 0, 0, 0.5, 0, \dots, 0\}$ and $\mathbf{x}_i \sim N_{10}(0, I)$.

The method described in Algorithm 1 is denoted by HDR (Historical Data Related). The results in Figure 3 show the high consistency with the last two examples. When σ is small, the different priors lead to similar result since the current data has key influence. However, when σ is large, the difference between RBM and HDR is huge. The reason is that the current data has been polluted by the strong noise.

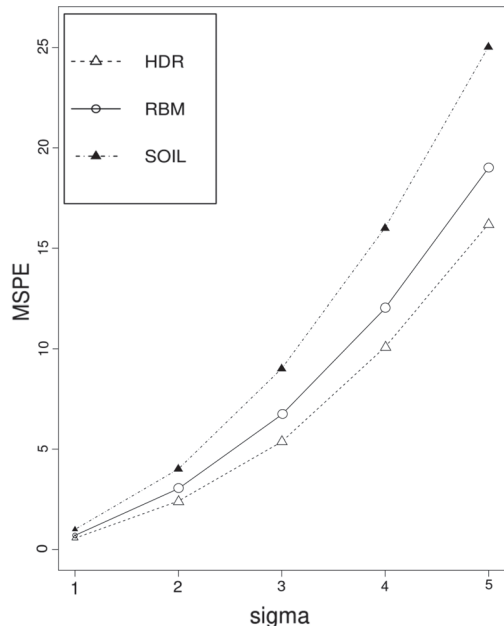


Figure 3: Comparison among RBM, SOIL and HDR of Example 3.

Hence, a good prior can provide the vital information about the model distribution.

4.2 Real-World Dataset

Here, we apply the proposed methods to two real datasets, BGS data and Bardet data, which are also used in Ye, Yang, and Yang (2016).

First, the BGS data is with small d and from the Berkeley Guidance Study (BGS) by Tuddenham and Snyder (1954). The dataset records 66 boys’ physical growth measures from birth to eighteen years. Following Ye, Yang, and Yang (2016), we consider the same regression model. The response is age 18 height and the factors include weights at ages two (WT2) and nine (WT9), heights at ages two (HT2) and nine (HT9), age nine leg circumference (LG9) and age 18 strength (ST18).

Second, for large d , the Bardet data collects tissue samples from the eyes of 120 twelve-week-old male rats. For each tissue, the RNAs of 31,042 selected probes are measured by the normalized intensity valued. The gene intensity values are in log scale. Gene TRIM32, which causes the Bardet-Biedl syndrome, is the response in this study. The genes that are related to it are investigated. A screening method (Huang, Ma, and Zhang 2008) is applied to the original probes. This screened dataset with 200 probes for each of 120 tissues is also used in Ye, Yang, and Yang (2016).

Both cases are data-given cases that we can’t use sequential batch sampling method. For the different setting of d , we assign corresponding similar historical data for two real datasets. The data of model 1 in Example 1 for the BGS data with small d . The data of model 3 in Example 1 for the

Bardet data with large d .

We randomly sample 10 rows from the data as the test set to calculate empirical MSPE and volatility. The results are summarized in Table 6. From Table 6, we can see that both RBM and HDR have smaller MSPE than SOIL. However, HDR does not perform much better than RBM. This can be explained intuitively as follows. In theory, the historical tasks and the current task are assumed that they come from the same task distribution. But in practice, how to measure the similarity between tasks is still a problem. Hence, an unrelated historical dataset may provide less information for the current prediction.

Table 6: Comparison among RBM, SOIL and HDR in real data.

		BGS	Bardet
MSPE	RBM	13.54	0.0054
	SOIL	16.74	0.0065
	HDR	13.06	0.0050
Volatility	RBM	1.99	0.0013
	SOIL	0.43	0.0013
	HDR	1.84	0.0012

4.3 Classification Tasks

In this section, the performance of our HDR method for classification tasks is demonstrated. Here, we use the same image classification example in Amit and Meir (2018). The hypothesis class is the set of neural networks including the architecture given in Amit and Meir (2018) and other CNN architectures provided in Keras. Different architectures are weighted by the parameter ω . And, the distribution Q_i is to characterize the hyperparameter of the i -th architecture. The cross-entropy loss is used.

The task environment is constructed based on augmentations of the MNIST dataset (LeCun 1998). Each task is created by a permutation of the image pixels. We randomly pick 100, 200 and 300 pixel swaps, and find that they have similar results. Thus we just show the results with 200 pixel swaps. For the meta-learner, it is trained by the meta-training tasks each with 50000 training samples and 10000 validation samples. For a new task with fewer training samples and 10000 test samples, we randomly sample 2000 training samples 20 times, and compare the average test error percentage of different learning methods. ‘95% CI’ in Table 7 means the 95% confidence interval. MLAP represents the method in Amit and Meir (2018), and different subscripts mean that different PAC bounds are used. The Model-Agnostic-Meta-Learning (Finn, Abbeel, and Levine 2017) is denoted by MAML. See Amit and Meir (2018) for detailed settings of the example.

Table 7 summarizes the results for the permuted pixels environment with 200 pixel swaps and 10 training tasks. We find that the best results are obtained by our HDR method. Note that for classification tasks, the weighted prediction is for the one-hot response. Hence, the prediction can be viewed as picking the largest probability among all the models not just in one model. Consequently, model averaging

Table 7: Comparisons of different learning methods on 20 test tasks of classification.

METHOD	ERROR	95% CI
MLAP _M	3.4	0.18
MLAP _S	3.54	0.2
MLAP _{PL}	74.9	4.03
MLAP _{VB}	3.52	0.17
MAML	3.77	0.8
HDR	0.72	0.0003

used in classification has much better performance than regression cases.

We also investigate whether the number of training tasks affects the error rate of the predictions on the new test tasks, and find that it is improved a lot if the meta-learner is used. But, the number of training tasks does not have significant effect.

Acknowledgment

We sincerely thank Prof. Zhihua Zhang (Peking University) and all reviewers for their valuable comments which have led to further improvement of this work.

References

Alquier, P., and Guedj, B. 2018. Simpler PAC-Bayesian bounds for hostile data. *Machine Learning* 107(5):887–902.

Alquier, P.; Ridgway, J.; and Chopin, N. 2016. On the properties of variational approximations of Gibbs posteriors. *The Journal of Machine Learning Research* 17(1):8374–8414.

Amit, R., and Meir, R. 2018. Meta-learning by adjusting priors based on extended PAC-Bayes theory. *international conference on machine learning* 205–214.

Catoni, O. 2004. *Statistical learning theory and stochastic optimization: Ecole d’Eté de Probabilités de Saint-Flour XXXI-2001*. Springer.

Catoni, O. 2007. PAC-Bayesian supervised classification: the thermodynamics of statistical learning. IMS.

Catoni, O. 2016. PAC-Bayesian bounds for the gram matrix and least squares regression with a random design. *arXiv preprint arXiv:1603.05229*.

Dziugaite, G. K., and Roy, D. M. 2018. Data-dependent PAC-Bayes priors via differential privacy. In *Advances in Neural Information Processing Systems*, 8430–8441.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1126–1135. JMLR. org.

Grünwald, P. D., and Mehta, N. A. 2016. Fast rates with unbounded losses. *arXiv preprint arXiv:1605.00252* 2:12.

Guedj, B.; Alquier, P.; et al. 2013. PAC-Bayesian estimation and prediction in sparse additive models. *Electronic Journal of Statistics* 7:264–291.

Hansen, B. E. 2007. Least squares model averaging. *Econometrica* 75(4):1175–1189.

Hjort, N. L., and Claeskens, G. 2003. Frequentist model average estimators. *Journal of the American Statistical Association* 98(464):879–899.

Huang, Y.; Huang, W.; Li, L.; and Li, Z. 2019. Meta-learning PAC-Bayes priors in model averaging. *arXiv preprint*.

Huang, J.; Ma, S.; and Zhang, C. H. 2008. Adaptive lasso for sparse high-dimensional regression. *Stat Sin* 18(4):1603–1618.

Leamer, E. E. 1978. *Specification searches*. New York: Wiley.

LeCun, Y. 1998. The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>.

Leung, G., and Barron, A. R. 2006. Information theory and mixing least-squares regressions. *IEEE Transactions on Information Theory* 52(8):3396–3410.

Lever, G.; Laviolette, F.; and Shawe-Taylor, J. 2013. Tighter PAC-Bayes bounds through distribution-dependent priors. *Theoretical Computer Science* 473(2):4–28.

Li, Z.; Zhou, F.; Chen, F.; and Li, H. 2017. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*.

Lugosi, G.; Mendelson, S.; et al. 2019. Regularization, sparse recovery, and median-of-means tournaments. *Bernoulli* 25(3):2075–2106.

Mcallester, D. A. 1999. PAC-Bayesian model averaging. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, 164–170.

Moral-Benito, E. 2015. Model averaging in economics: An overview. *Journal of Economic Surveys* 29(1):46–75.

Oneto, L.; Anguita, D.; and Ridella, S. 2016. PAC-bayesian analysis of distribution dependent priors: Tighter risk bounds and stability analysis. *Pattern Recognition Letters* 80:200–207.

Raftery, A. E. 1995. Bayesian model selection in social research. *Sociological Methodology* 25(25):111–163.

Raftery, A. E. 1996. Approximate bayes factors and accounting for model uncertainty in generalised linear models. *Biometrika* 83(2):251–266.

Rivasplata, O.; Szepesvari, C.; Shawe-Taylor, J. S.; Parrado-Hernandez, E.; and Sun, S. 2018. PAC-Bayes bounds for stable algorithms with instance-dependent priors. In *Advances in Neural Information Processing Systems*, 9214–9224.

Seeger, M. 2002. PAC-Bayesian generalisation error bounds for gaussian process classification. *Journal of Machine Learning Research* 3(2):233–269.

Seldin, Y., and Tishby, N. 2010. PAC-Bayesian analysis of co-clustering and beyond. *Journal of Machine Learning Research* 11(Dec):3595–3646.

Tuddenham, R. D., and Snyder, M. M. 1954. Physical growth of california boys and girls from birth to eighteen years. *Publications in Child Development* 1:183–364.

Wang, H.; Zhang, X.; and Zou, G. 2009. Frequentist model averaging estimation: a review. *Journal of Systems Science and Complexity* 22(4):732.

Yang, Y. 2000. Combining different procedures for adaptive regression. *Journal of multivariate analysis* 74(1):135–161.

Yang, Y. 2001. Adaptive regression by mixing. *Journal of the American Statistical Association* 96(454):574–588.

Ye, C.; Yang, Y.; and Yang, Y. 2016. Sparsity oriented importance learning for high-dimensional linear regression. *Journal of the American Statistical Association* (2):1–16.

Zhou, Q.; Ernst, P. A.; Morgan, K. L.; Rubin, D. B.; and Zhang, A. 2018. Sequential rerandomization. *Biometrika* 105(3):745–752.