

Asymptotic Meta learning for Cross Validation of models for financial data

Haitao Xiang¹, Jianwu Lin², Chun-hung Chen³, Ying Kong⁴

Abstract – Meta learning is an advanced field of Artificial Intelligence (AI) where automatic learning algorithms are applied to acquire learning experience for a set of learning algorithms to improve learning performance. One of popular Meta learning methodologies is based on cross validation, especially for selection processes among different machine learning models. However, the challenge is that it is very time-consuming to do cross validation among models in large data sets, especially in financial big data with high noise. This paper proposes two Asymptotic Meta learning algorithms (AML-lin and AML-xiang), which are ordinal optimization algorithms for Meta learning based on cross validation. The numerical experiments and real-world cases are conducted to illustrate its efficiency in cross validation of models in different scenarios, especially for financial data. The method proposed in this paper has significant improvement by comparing with those ones in existing algorithms OCBA and IAML (e.g., see reference [8] [9]), and it is new in dealing with financial data.

Keywords – Financial data, Quantitative Investment, Artificial intelligence, Meta learning, Cross validation

1 INTRODUCTION

Meta learning is an advanced field of AI where automatic learning algorithms are applied to acquire learning experience for a set of learning algorithms to improve learning performance.¹ A Meta learning system must include a learning subsystem, so called Meta learner, which adapts with Meta data of underlying base learners to improve overall learning performance over data [1]. The goal of Meta learner can be to select a best base learner amongst many candidate learners, to select a best combination of multiple base learners or to dynamically modify the current base learner to improve its learning performance by leveraging Meta data. The Meta data, also called Meta-knowledge can be characteristics of data, configuration spaces and prior learning experience [2]. For the learning

performance over data, Meta learning may focus on maxing out learning speed subject to certain learning accuracy or maxing out learning accuracy subject to certain learning speed.

In order to make the problem more general, we concentrate on selecting a best model by prior learning experience to improve learning accuracy restricted by certain learning speed by following reasons. 1) Combining multiple models effectively can be viewed as selecting the best weighting set for a combination of multiple models. Modifying the current model dynamically can be viewed as seeking the best updating logic for the current model. 2) In most of time, leveraging characteristics of data and configuration spaces is specific to certain learning problems case by case. So we like to focus on leveraging prior learning experience, which can

¹ Haitao Xiang is with Mathematical Institute, University of Oxford, Andrew Wiles Building, Radcliffe, Observatory Quarter, Woodstock Road, Oxford, OX2 6GG, United Kingdom. Haitao Xiang is also with Tsinghua-Berkeley Shenzhen Institute, Shenzhen, 518000, China

²Jianwu Lin is with Graduate school at Shenzhen, Tsinghua University, Shenzhen, 518000, China

lin.jianwu@sz.tsinghua.edu.cn

³ Chun-hung Chen is with George Mason University, Virginia, 22030, USA

⁴ Ying Kong is Graduate school at Shenzhen, Tsinghua University, Shenzhen, 518000, China

be generally applied to most of Meta learning systems. 3) In most of time-limited real-world cases, max out the computing time to improve learning accuracy is more important.

Even if some researchers try to treat Meta learning as a deterministic optimization problem, we consider it as a stochastic optimization problem and measure base-learner's expected performance based on cross validation, which is a typical way to compare AI models. The main idea of cross validation is to divide the data into two parts, one is for model training, the other is for estimating the prediction error of the trained model. Finally, the model with the least prediction error is selected as the optimal model. In addition, due to the different cross-validation methods and times of data segmentation, many different methods have been generated. How to select appropriate cross-validation methods for the data in hand has become the focus of research [3]. Typically, the cross validation has 3 types: Holdout validation, k-fold cross validation and leave-one-out cross validation. K-fold cross validation and leave-one-out cross validation are similar. They equally divide the data set to several parts and pick one part for validation while using the rest of parts for model training. They are popular because they are simple and speedy. However, it misses many different combinations of data set we need to test. Holdout validation is a non-typical cross validation. It randomly divides the data set for training and validation. It can fully leverage the data set to get a relatively complete distribution for training performance. However, the challenge is that it is very time-consuming.

When Holdout validation tries to find the best model subject to a time constraint, it becomes a stochastic optimization problem by Discrete Event Simulation (DES). And the cross validation care more about the rank of candidate learners than their accurate performance. Ordinal Optimization has emerged as an efficient technique for validation and optimization. The

underlying philosophy is to obtain good estimates through ordinal comparison while the value of an estimate is still very poor [4]. Since our goal is to find the best models rather than to find an accurate estimate of the best performance value, it is advantageous to use ordinal comparison for selecting the best model. Further Dai (1996) [5] shows that the convergence rate for ordinal optimization can be exponential. This idea has been successfully applied to several problems [6] [7].

While ordinal optimization could significantly reduce the computational cost for DES, there is potential to further improve its performance by intelligently controlling the validation experiments, or by determining the best number of validation samples among different designs as validation proceeds [8]. The design Chen mentioned in his paper is equivalent to a base learner in this paper. One of general information we can leverage in the Meta learning problem is prior learning experience. It is applied to not only rank the candidate learners, but also improve the learning accuracy subject to a time constraint. However, OCBA does not deal with certain issues which have different computation time for each design (Or base learner in Meta learning) Our previous work for asymptotic Meta learning in 2017 intuitively modified the OCBA theorem by adding weights proportional to square root of computation time of each design [9].

Among many data analysis fields, quantitative investment is one of most challenging ones because its data is very noisy and its patterns are fragile and non-stationary in some sense [9]. So this paper tests our algorithm not only on pre-setup learning problems, but also simple quantitative investment strategies based on financial indicators and AI machine learning problems in quantitative investment areas to verify its effectiveness and efficiency.

The structure of the rest of the paper is: Section 2 defines Meta learning problem based on cross validation as an ordinal optimization problem by

DES. In section 3, we provide asymptotic closed-form solution of this optimization problem, then introduce the AML-lin that asymptotically solve the optimization problem. The AML-xiang provides the sequential quadratic programming solution of the ordinal optimization problem. Session 4 tests the algorithms in different numerical experiments to verify their effectiveness and efficiency. Section 5 applies it to two typical processes of quantitative investment strategy modeling to verify its efficiency in more challenging real-world cases. Section 6 concludes this research and proposes future improvements we can make for AML.

2 PROBLEM STATEMENT

As discussed in the last section, this paper focuses on a problem to max the chance of selecting the best base learner by cross validation subject to a time limit in a stochastic ordinal optimization. So we can define the Meta learning as a probability max problem. In this paper, we specifically define a cross validation as the following process: Divide data set into two parts, training data set and validation data set, train a base learner in-sample by the training data set and validate the trained learner out-of-sample by the validation data set. We do a cross validation by randomly dividing data set for each base learner and we select the base learner among a finite number of candidates by the rank of their mean performance. The total computing time for the cross validation is subject to a predefined time limit. We define Probability of Correct Selection (PCS) as the probability that base learner b is the true best base learner. To be more precisely, our goal function—a lower bound for PCS—the approximate probability of correct selection (APCS) is defined as

$$\begin{aligned} \max_{N_1, \dots, N_k} & 1 - \sum_{i=1, i \neq b}^k P\{-\bar{J}_b > -\bar{J}_i\} \\ \text{s.t.} & \sum_{i=1}^k C_i N_i = T \end{aligned}$$

(1)

where by normal assumption of \bar{J}_b and \bar{J}_i we have

$$\begin{aligned} P\{-\bar{J}_b > -\bar{J}_i\} &= \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_{b,i}} e^{-\frac{(x-\delta_{b,i})^2}{2\sigma_{b,i}^2}} dx \\ &= \int_{\frac{\delta_{b,i}}{\sigma_{b,i}}}^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt \end{aligned} \quad (2)$$

$$\text{Where } \sigma_{b,i} = \sqrt{\frac{\sigma_b^2}{N_b} + \frac{\sigma_i^2}{N_i}}, \delta_{b,i} = -\bar{J}_b + \bar{J}_i.$$

So the goal function then becomes

$$\begin{aligned} (P1) \min_{N_1, \dots, N_k} & F(N_1, \dots, N_k) \\ &= \sum_{i=1, i \neq b}^k \int_{\frac{\delta_{b,i}}{\sigma_{b,i}}}^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt \\ &\text{s.t. } \sum_{i=1}^k C_i N_i = T \end{aligned} \quad (3)$$

where $\sigma_{b,i} = \sqrt{\frac{\sigma_b^2}{N_b} + \frac{\sigma_i^2}{N_i}}, \delta_{b,i} = -\bar{J}_b + \bar{J}_i$, and b is the index for current best.

Here N is the set of non-negative integers and *summation denotes* the constraint on the time constraint, where C_i stands for time consumption in one cross validation of base learner i , and N_i stands for the number of cross validations allocated to base learner i . For simplicity, we assume the time consumption for each base learner is known and fixed.

3 ALGORITHM

We propose two Meta-algorithms to crackle the budget allocation problem.

a) AML-Lin

We use asymptotic scheme to solve the constrained optimization problem in (1), and nest it into the budget allocation scheme, which we name as AML-Lin.

Specifically, the problem in section 2 can be solved by proving Theorem 1.

Theorem 1. *Given a total number of computing time T to be allocated to k base learners whose performance is depicted by cross validation performance $L(\theta_1, \xi), L(\theta_2, \xi), \dots, L(\theta_k, \xi)$ with means $J(\theta_1), J(\theta_2), \dots, J(\theta_k)$, and finite variances $\sigma_1^2, \sigma_2^2, \dots, \sigma_k^2$ and average learning time per learner C_1, C_2, \dots, C_k , respectively, as $T \rightarrow \infty$, the Approximate Probability of Correct Selection (APCS) of Meta learning can be asymptotically maximized when*

$$1) \quad \frac{N_i}{N_j} = \left(\frac{\left(\frac{\sigma_i}{\delta_{b,i}} \right)}{\left(\frac{\sigma_j}{\delta_{b,j}} \right)} \right)^2, i, j \in \{1, 2, \dots, k\}, i \neq j \neq b$$

$$2) \quad N_b = \sigma_b \sqrt{\sum_{i=1, i \neq b}^K \frac{C_i N_i^2}{C_b \sigma_i^2}}.$$

Where N_i is the number of samples allocated to base learner i , $\delta_{b,i} = \bar{J}_b - \bar{J}_i$, and $\bar{J}_b \geq \max_i \bar{J}_i$.

And we assume $N_b \gg N_i$.

The detail of proof can be found in the Appendix. In our previous work in 2017, optimal N_b is approved. However, the ratio between N_i and N_j is just assumed intuitively. This paper provides a better solution for this ratio based on mathematical approval. Instead of using weights proportional to square root of computation time of each design, this paper finds in the best mathematical solution, the ratio between N_i and N_j is not related to computation time of each design. It looks like a little counter-intuitive. However, it is supported by mathematical approval and numerical experiments in later section.

Since we need to perform an initial run without previous cross validation, it is important to discuss an optimal allocation for the initial run. In the initial run, our best estimation for all base learners is that they have equal performance mean

and variance. So from theorem 1, we can prove that

$$\frac{N_i}{N_j} = \left(\frac{\left(\frac{\sigma_i}{\delta_{b,i}} \right)}{\left(\frac{\sigma_j}{\delta_{b,j}} \right)} \right)^2 = 1$$

With Theorem 1, we now present a cost-effective sequential Meta learning algorithm called Asymptotic Meta learning (AML) to select the best base learner from k candidates with a computing time limit. Ideally, each new replication should bring us closer to the optimal solution. This procedure is continued until the total computing time T is exhausted. The algorithm is summarized as follows.

Meta-Algorithm 1: An Asymptotic Meta learning by Lin (AML-Lin)

Step 0. Perform cross validation in n_0 times for each base learners; $l \leftarrow 0$; $N_1^l = N_2^l = \dots = N_k^l = n_0$ and $\sum_{i=1}^k C_i N_i^l = t_0$.

Step 1. If $\sum_{i=1}^k C_i N_i^l \geq T$ stop.

Step 2. Increase the computing time limit (i.e., number of additional cross validation time) by ΔT and compute the new validation time allocation, $N_1^{l+1}, N_2^{l+1}, \dots, N_k^{l+1}$, using Theorem 1.

Step 3. Perform additional $\max(0, N_i^{l+1} - N_i^l)$ cross validations for base learner i , $i = 1, \dots, k$.

$l \leftarrow l + 1$. Go to Step 1.

In the above algorithm, l is the iteration number. As the learning process evolves, base learner b , which is the base learner with the largest sample mean performance, may change from iteration to iteration, although it will converge to the optimal base learner as the l goes to infinity. When b changes, Theorem 1 is directly applied in step 2. However, the older base learner b may not be cross validated at all in this iteration in step 3 due to extra allocation to this base learner in earlier iterations.

We can see that if we assume $t_0 = T$ and all base learners have same validation time, this algorithm goes back to typical Holdout cross

validation with equal number of validation allocated to each base learners.

b) AML-Xiang

We use Sequential Quadratic Programming (SQP) to solve the constrained optimization problem in (1), and nest it into the budget allocation scheme, which we name as AML-Xiang, to differentiate from the asymptotic scheme AML-Lin.

We define $\mathbf{N} = [N_1, \dots, N_k]^T$ the budget allocation, and $\mathbf{N}^l = [N_1^l, \dots, N_k^l]^T$ denote the l th iteration budget allocation. $\mathbf{C} = [C_1, \dots, C_k]^T$ as the average time consumption for each design, and the goal function becomes $F(\mathbf{N})$, and we propose Algorithm SQP as follows.

Algorithm SQP: SQP ($\mathbf{N}^1, \mathbf{C}, \Delta S, \Delta T, \epsilon, \delta_{b,i}, \sigma_{b,i}$)

Input: $\mathbf{N}^0 = \mathbf{0}$, current allocation of budget $\mathbf{N}^1 > \mathbf{0}$, time consumption \mathbf{C} , additional budget ΔS , additional time budget ΔT , error tolerance ϵ ,

$$\delta_{b,i} = -\bar{f}_b + \bar{f}_i, \sigma_{b,i} = \sqrt{\frac{\sigma_b^2}{N_b} + \frac{\sigma_i^2}{N_i}}$$

Output: computational budget allocation \mathbf{N}

While $\|\mathbf{N}^{l+1} - \mathbf{N}^l\|^2 > \epsilon$:

$$H^l = \nabla^2 F(\mathbf{N}^l) \text{ (using L-BFGS [11])}$$

$$G^l = \nabla F(\mathbf{N}^l)$$

solve (using Lagrangian Newton [12]) \mathbf{s} for

$$\min_{\mathbf{s}} \frac{1}{2} \mathbf{s}^T H^l \mathbf{s} + G^l \mathbf{s}$$

$$\text{s.t. } \mathbf{C}^T \mathbf{s} \leq \Delta S, \mathbf{s} > \mathbf{0}, \mathbf{1}^T \mathbf{s} \leq \Delta T$$

$$\mathbf{N}^{l+1} = \mathbf{N}^l + \mathbf{s}$$

For the simulation budget allocation problem, the application of Algorithm SQP is nested in the AML Algorithm. Specifically, Given K alternatives to select the best design from, and given a total computational budget T . Initially, we assign n_0 simulation to each of the alternatives. With n_0 assigned, we obtain the performances (mean and variance) of each alternative, which are then used to compute the $\delta_{b,i}, \sigma_{b,i}$ specified

in Algorithm SQP. An incremental simulation budget ΔS is then allocated to all the designs based on Algorithm SQP. This complete one loop. The loop will keep on until the total budget is used up, which is as follows

Meta-Algorithm 2: An Asymptotic Meta learning by Xiang (AML-Xiang)

Step 0. Perform simulation in n_0 times for each designs; $l \leftarrow 0$; $N_1^l = N_2^l = \dots N_k^l = n_0$ and $\sum_{i=1}^k N_i^l = t_0$.

Step 1. If $\sum_{i=1}^k N_i^l \geq T$ stop.

Step 2. Increase the computing time limit (i.e., number of additional simulation budget) by Δ and compute the new simulation budget allocation, $\mathbf{N}^{l+1} = [N_1^{l+1}, N_2^{l+1}, \dots, N_k^{l+1}]^T$, using

Algorithm SQP:

$$\mathbf{N}^{l+1} = \text{SQP}(\mathbf{N}^l, \mathbf{C}, \Delta S, \Delta T, \epsilon, \delta_{b,i}, \sigma_{b,i}).$$

Step 3. Perform additional $\max(0, N_i^{l+1} - N_i^l)$ simulations for design $i, i = 1, \dots, K$.

$l \leftarrow l + 1$. Go to Step 1.

4 NUMERICAL EXPERIMENTS

In this section, the AML algorithm is tested and compared with several different allocation procedures by performing a series of numerical experiments including OCBA (Assume valuation time for all learners are equal) [9]:

$$1) \quad \frac{N_i}{N_j} = \left(\frac{\left(\frac{\sigma_i}{\delta_{b,i}} \right)^2}{\left(\frac{\sigma_j}{\delta_{b,j}} \right)^2} \right), i, j \in \{1, 2, \dots, k\}, i \neq j \neq b$$

$$2) \quad N_b = \sigma_b \sqrt{\sum_{i=1, i \neq b}^K \frac{N_i^2}{\sigma_i^2}}.$$

Greedy (always allocate valuation time to the current best) [10], IAML (It is our previous work for asymptotic Meta learning based on our intuitive assumption for the ratio between N_i and N_j in 2017. In order to differentiate with AML-lin in this paper, it is called IAML in this paper):

TABLE 1

Validation time with 95% PCS Cutoff (in seconds) for six Meta learning algorithms under different case. The smallest time is highlighted in bold.

	AML-Lin	IAML	OCBA	Equal	Greedy	AML-Xiang
Case 1	415.00	527.77	418.11	962.50	616.22	429.75
Case 2	758.34	773.16	654.40	3675.00	951.13	602.89
Case 3	1311.60	1531.65	1486.47	2035.00	1321.33	1004.63
Case 4	1183.51	2143.04	1368.57	3272.50	2864.19	1014.47
Case 5	1822.53	2347.85	1932.46	3272.50	2921.06	1368.89
Case 6	577.10	838.43	588.08	1402.50	807.50	463.14

$$1) \quad \frac{N_i}{N_j} = \left(\frac{C_j}{C_i}\right)^{\frac{1}{2}} \left(\frac{\left(\frac{\sigma_i}{\delta_{b,i}}\right)^2}{\left(\frac{\sigma_j}{\delta_{b,j}}\right)^2}\right), i, j \in \{1, 2, \dots, k\}, i \neq j \neq b$$

$$2) \quad N_b = \sigma_b \sqrt{\sum_{i=1, i \neq b}^K \frac{C_i N_i^2}{C_b \sigma_i^2}}.$$

and typical Holdout validation (Allocate the number of validations equally to each learner, called as Equal in experiments).

In all the numerical illustrations, we estimate *PCS* by counting the number of runs we successfully find the true best learner out of 10,000 independent runs of each selection procedure. *PCS* is then obtained by dividing this number by 10,000, representing the probability of correct selection.

$$P\{CS, t\} = \Pr(\mu_b > \mu_i, i \neq b | X_i, i = 1, \dots, k)$$

To illustrate the efficiency of AML in different scenarios, we test six cases as following:

- 1) Base case: Learning performance of base learners is normally distributed
In this case, there are K base learners. Suppose $L(\theta, \xi) \sim N(i, 6)$, $i = 1, \dots, K$. Each base learner will take a random validation time $C_i \in \{1 * 0.5, \dots, K * 0.5\}$ seconds. We want to find a base learner with the max mean learning performance. It is obvious that learner K is the actual best learner. In the

numerical experiment, we compare the convergence of *PCS* for different allocation procedures. We have $t_0 = 10$, $\Delta S = 10$ and $\Delta T = 100$. We set $K=10$ in this case. The cutoff of *PCS* is 95% in this case.

- 2) Effect of the numbers of base learners: The number of learners is larger than ones in base case

This case studies the effect of numbers of base learners. Therefore, we double the numbers of learners to 20 ($K=20$) in this case. The remaining settings are identical to that of the base case. So we can compare $K = 10$ and $K=20$ to see how AML performs over the number of learners.

- 3) Effect of validation time of base learners: The difference of validation time among base learners is larger than ones in base case

This case studies the effect of validation time of learners. Therefore, Each base learner will take a doubling random validation time $C_i \in \{2, 4, \dots, 2 * K\}$ units. The remaining settings are identical to that of the base case.

- 4) Effect of variance of learning performance: The variance of learning performance of base learners is larger than ones in base case

This case studies the effect of the variance of learning performance. Therefore, we set $L(\theta, \xi) \sim N(i, 10)$, $i = 1, \dots, K$. The remaining settings are identical to that of the base case.

5) Effect of cardinality of learning performance:

The gap of mean learning performance of base learners is smaller than ones in base case. This case studies the effect of gap of mean learning performance. Therefore, we divide the mean learning performance of each learner by 2, so that the gap of mean learning performance of base learners is smaller. Set $L(\theta, \xi) \sim N(i/2, 6)$, $i = 1, \dots, K$. The remaining settings are identical to that of the base case.

6) Effect of distribution type of learning performance: Learning performance of base learners is uniformly distributed with same mean and variance as base case

This case studies the effect of a different distribution than previous normal distribution. Therefore, we choose uniform distribution with a larger variance. set $L(\theta, \xi) \sim \text{uniform}(i, i+24)$ $i = 1, \dots, K$. so the variance is $\text{var} = \frac{(\beta-\alpha)^2}{12} = 24 * \frac{24}{12} = 48$.

The remaining settings are identical to that of the base case.

Intuitively, cases 2,3,4,5,6 are tougher cases than the base case in a Meta-learning case. The more candidate learners, a smaller gap of mean performance, larger variance and uniform distribution with fatter tail than normal distribution usually make the sample distribution of learning performance have more overlap across base learners, and therefore increase the difficulty to select best base learner. And the larger difference among base learners' learning time makes it more difficult to balance the trade-off between allocating numbers of validations and allocating learning time. In other words, there may be the case that we need to spend more learning time in improving the estimation of learning performance for some base learners. Table 1 shows the results of numerical experiments for the six cases.

Table 2. Time improvement ratio for the best Meta-algorithm in all six cases

time improvement ratio	Min	Max	Average
Case 1	99%	45%	73%
Case 2	92%	16%	44%
Case 3	77%	49%	65%
Case 4	86%	31%	47%
Case 5	75%	42%	56%
Case 6	80%	33%	55%

In table 1, we can see that both AML-lin and AML-xiang try to reduce time. AMLs spends the least time among all algorithms (AML-lin the least in case 1, and AML-xiang the least in case 2,3,4,5,6). These results reflect the very essence of AML, to reduce validation time, not to reduce number of validations. And since AML has better mathematical solutions than IAML, both AML-lin and AML-xiang beats IAML in all cases.

Since cases 2~6 are tougher cases than the base case, AMLs spends more validation time in these five cases than the base case. Moreover, AML-xiang performs the best in case 2~6, meaning AML-xiang is more useful in harder cases. However, the min time improvement ratio (validation time of AML / validation time of the second best algorithm), max time improvement ratio (validation time of AML / validation time of the worst algorithm) and average time improvement ratio (validation time of AML / average of validation time of the rest of 3 algorithms) shown in Table 2 are stable. The results illustrate that AMLs performs well even for tougher cases.

5 REAL-WORLD APPLICATIONS

Constructing quantitative investment strategies is one of most challenging tasks in financial data modeling. If we treat candidate quantitative investment strategies as base learners, the process to select the best strategy among them is a typical Meta learning problem for cross validation and can be optimized by AML. However, this kind of

real-world financial data modeling problem combines many challenging cases tested in last experiment together: Larger difference of validation time among learners, larger variance of learning performance of learners, smaller gap of mean learning performance of learners and non-normally distributed Learning performance of learners. In the following part, AML is applied to two typical applications, quantitative investment strategies based on simple financial indicators and complex AI machine learning algorithms, in order to verify its efficiency.

Application 1. Quantitative investment strategies based on simple financial indicators

In this application, we adopt 5 investment strategies: Bollinger Band, MACD, RSI, momentum, AR model [13] [14] [15] [16].

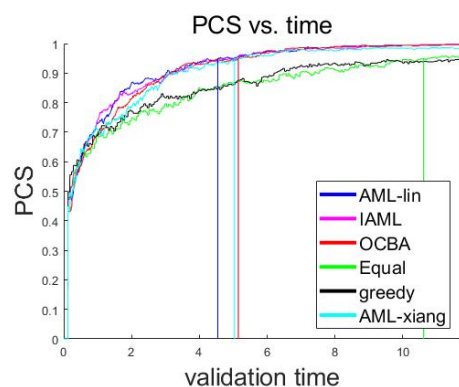
Each strategy is a base learner with parameters to be learned from stock price time series data. We train each strategy using in-sample data. The training process is grid search of best parameters. And the trained parameters are used to conduct out-of-sample validation. We pick 24 months out of 18 months of data as in-sample training data set, and the rest 6 months as out-of-sample validation data set from January 2017 to December 2018 from all stocks traded in China A-share market. We define a strategy with the highest average out-of-sample Sharpe Ratio to be the best learner.

In Figure 1a, we can find that AML-lin performs best in terms of validation time. This might be due to the non-normality of the financial investment Sharpe ratio. Therefore, it might be more desirable to use AML-lin in the strategy selection problem, when AML-xiang is sub-optimal. However, both AML algorithms outperforms state of the art OCBA. In contrast to traditional Meta learning methods that minimize number of validation, what really matters is validation time in the financial industry, therefore our AMLs are more favorable. And both AML-lin and AML-xiang beats IAML again in this real-world case.

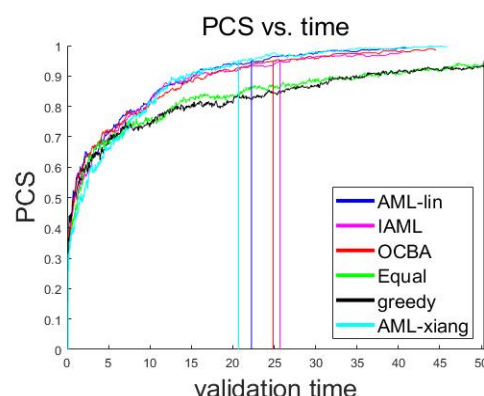
Application 2. Quantitative investment strategies based on complex AI machine learning algorithms

In this application, we choose 5 machine learning algorithms as base learners on a task of stock selection. The 5 algorithms are generalized linear models, naïve Bayes, ensemble model, support vector machine and artificial neural networks. We conduct the six Meta learning algorithms based on cross validation for each of them. Specifically, we have a daily panel data of features from January 2017 to December 2018 from all stocks traded in China A-share market. The goal is to train the model using in-sample data to predict performance of out-of-sample data, and pick the best 50 stocks in the out-of-sample data for a stock portfolio. Cross validation of the data requires traversal of all the combinations of the 18 months of data. For illustrative purposes, we set in-sample training set to be 18 months and out-of-sample test set to be 6 months complementary. The combination then will be $C_{24}^{18} = 134596$ for all the cross validation sets. We define an AI machine learning algorithm with the highest average out-of-sample excess return to be the best learner.

In Figure 1b, we can find that AML-xiang outperforms all other algorithms in terms of validation time, reaching about 12% reduction of validation time against the second best validation algorithm, OCBA and AML-lin. Therefore, in more difficult settings, it is more desirable to use AML-xiang. In this relatively complex case which is not included in our pervious paper in 2017, IAML loses to AML-lin, AML-xiang and OCBA.



a. Selection of financial indicators



b. Selection of AI machine learning algorithms

Figure 1. PCS of six allocation algorithms against validation time for selections of quantitative investment strategies

6 CONCLUSION AND FUTURE RESEARCH

Based on numerical experiments above, we can see the both AML-lin and AML-xiang performs better than existing cross-validation algorithms and the advantage of AML-xiang becomes more significant in some harder Meta learning cases. Since both AML-lin and AML-xiang comes out of full mathematical approval, they performs much better than our previous work, IAML. However, there is still some future research left for us to conduct. Currently we separate our data set into training data set and validation data set and initial time increasing time for each run (n_0 and $\Delta S, \Delta T$) based on a simple rule of thumb. In the future research, optimizing these parameters

will be an interesting research topic to further improve the performance of this algorithm. Also the stochastic unknown validation time we ignore in the problem statement can be an interesting future research direction.

In this paper, we verify AMLs' efficiency in two quantitative investment modeling selection problems. Further investigation on other Fin Tech modeling areas such as credit risk modeling and block-chain algorithms may be needed to expand its applications.

Once again, the numerical study shows that the method proposed in this paper has significant improvement by comparing with those existing algorithms OCBA and IAML (e.g., see reference [8] [9]), and it is first time used to deal with financial data.

REFERENCES

- [1] Lemke, Christiane; Budka, Marcin; Gabrys, Bogdan, "Metalearning: a survey of trends and technologies". *Artificial Intelligence Review*. **44** (1): 117–130, 2013
- [2] Vanschoren, Joaquin, *Meta-Learning: A Survey*, arXiv:1810.03548 [cs.LG], 2018
- [3] Fan, Yongdong. *Cross validation in model selection: A Survey*, Dissertation of Shanxi University, 2013.
- [4] Ho, Y. C., R. S. Sreenivas, and P. Vakili, "Ordinal Optimization of DEDS," *Journal of Discrete Event Dynamic Systems*, 2, #2, 61-88, 1992.
- [5] Dai, L., "Convergence Properties of Ordinal Comparison in the Validation of Discrete Event Dynamic Systems," *Journal of Optimization Theory and Applications*, Vol. 91, No.2, pp. 363-388, 1996.
- [6] Ramirez-Marquez J E. Identification of top contributors to system vulnerability via an ordinal optimization based method [J]. *Reliability Engineering & System Safety*, 2013, 114: 92-98.
- [7] Tsai C F, Hsu Y F. A Meta-learning Framework for Bankruptcy Prediction [J]. *Journal of Forecasting*, 2013, 32(2): 167-

179.

- [8] Chun-Hung Chen, Jianwu Lin, Enver Yücesan, Stephen E. Chick, Validation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization, *Journal of Discrete Event Dynamic Systems: Theory and Applications*, Vol. 10, pp. 251-270, July 2000
- [9] Lin J, Xiang H, Li J, et al. Best investment strategy selection using asymptotic meta learning[C]//2017 IEEE/SICE International Symposium on System Integration (SII). IEEE, 2017: 72-76.
- [10] Chen, H. C., Chen, C. H., Dai, L., and Yücesan, E. 1997. New development of optimal computing budget allocation for discrete event simulation. Proceedings of the 1997 Winter Simulation Conference, pp. 334-341.
- [11] Zhu C, Byrd R H, Lu P, et al. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization[J]. ACM Transactions on Mathematical Software (TOMS), 1997, 23(4): 550-560.
- [12] Alt W, Malanowski K. The Lagrange-Newton method for state constrained optimal control problems[J]. Computational Optimization and Applications, 1995, 4(3): 217-239.
- [13] Bollinger J. Using bollinger bands[J]. Stocks & Commodities, 1992, 10(2): 47-51.
- [14] Rosillo R, De la Fuente D, Brugos J A L. Technical analysis and the Spanish stock exchange: testing the RSI, MACD, momentum and stochastic rules using Spanish market companies[J]. Applied Economics, 2013, 45(12): 1541-1550.
- [15] Wilder J W. New concepts in technical trading systems[M]. Trend Research, 1978.
- [16] Neftci S N. Naive trading rules in financial markets and wiener-kolmogorov prediction theory: a study of" technical analysis"[J]. Journal of Business, 1991: 549-571.

Appendix: Proof of AML-Lin

We quote literature [9] for relevant assumptions, and our goal function is

$$\max_{N_1, \dots, N_k} 1 - \sum_{i=1, i \neq b}^k P\{\bar{J}_b > \bar{J}_i\}$$

$$\text{s.t. } \sum_{i=1}^k C_i N_i = T$$

where by normal assumption of \bar{J}_b and \bar{J}_i we have

$$P\{\bar{J}_b > \bar{J}_i\} = \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_{b,i}} e^{-\frac{(x-\delta_{b,i})^2}{2\sigma_{b,i}^2}} dx$$

$$= \int_{\frac{\delta_{b,i}}{\sigma_{b,i}}}^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$$

$$\text{Where } \sigma_{b,i} = \sqrt{\frac{\sigma_b^2}{N_b} + \frac{\sigma_i^2}{N_i}}$$

the lagrangian of objective maximization is

$$F = 1 - \sum_{i=1, i \neq b}^k \int_{\frac{\delta_{b,i}}{\sigma_{b,i}}}^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt - \lambda \left(\sum_{i=1}^k C_i N_i - T \right)$$

by F.O.N.C, we have

$$\begin{aligned} \frac{\partial F}{\partial N_i} &= \frac{\partial F}{\partial \left(-\frac{\delta_{b,i}}{\sigma_{b,i}} \right)} \frac{\partial \left(-\frac{\delta_{b,i}}{\sigma_{b,i}} \right)}{\partial \sigma_{b,i}} \frac{\partial \sigma_{b,i}}{\partial N_i} - \lambda C_i \\ &= \frac{-1}{\sqrt{2\pi}} \exp \left[\frac{\delta_{b,i}^2}{\sigma_{b,i}^2} \right] \left(\frac{\delta_{b,i} \sigma_i^2}{N_i^2 \sigma_{b,i}^3} \right) - \lambda C_i = 0 \quad (1) \end{aligned}$$

$$\frac{\partial F}{\partial N_b} = \frac{-1}{\sqrt{2\pi}} \sum_{i=1, i \neq b}^k \exp \left[\frac{\delta_{b,i}^2}{\sigma_{b,i}^2} \right] \left(\frac{\delta_{b,i} \sigma_b^2}{N_b^2 \sigma_{b,i}^3} \right) -$$

$$\lambda C_b = 0 \quad (2)$$

From equation (1), we have

$$\frac{-1}{\sqrt{2\pi}} \exp \left[\frac{\delta_{b,i}^2}{\sigma_{b,i}^2} \right] \left(\frac{\delta_{b,i}}{\sigma_{b,i}^3} \right) = \lambda C_i \frac{N_i^2}{\sigma_i^2}$$

$$(3)$$

Putting together, we have

$$\sum_{i=1, i \neq b}^k -\frac{\lambda C_i N_i^2 \sigma_b^2}{C_b N_b^2 \sigma_i^2} - \lambda = 0$$

Then

$$N_b = \sigma_b \sqrt{\sum_{i=1, i \neq b}^k \frac{C_i N_i^2}{C_b \sigma_i^2}}$$

From equation (1), we have

$$\begin{aligned} & \exp\left(-\frac{\delta_{b,i}^2}{2\left(\frac{\sigma_b^2}{N_b} + \frac{\sigma_i^2}{N_i}\right)}\right) \cdot \frac{\frac{\delta_{b,i}\sigma_i^2}{C_i N_i^2}}{\left(\frac{\sigma_b^2}{N_b} + \frac{\sigma_i^2}{N_i}\right)^{\frac{3}{2}}} \\ &= \exp\left(-\frac{\delta_{b,j}^2}{2\left(\frac{\sigma_b^2}{N_b} + \frac{\sigma_j^2}{N_j}\right)}\right) \cdot \frac{\frac{\delta_{b,j}\sigma_j^2}{C_j N_j^2}}{\left(\frac{\sigma_b^2}{N_b} + \frac{\sigma_j^2}{N_j}\right)^{3/2}} \end{aligned} \quad (4)$$

we conclude from equation (3) that

$$N_b = \sigma_b \sqrt{\sum_{i=1, i \neq b}^K \frac{C_i N_i^2}{C_b \sigma_i^2}}$$

and from equation (4) by assuming

$N_b \gg N_i$ we have that

$$\begin{aligned} & \exp\left(-\frac{\delta_{b,i}^2}{2\left(\frac{\sigma_i^2}{N_i}\right)}\right) \cdot \frac{\frac{\delta_{b,i}\sigma_i^2}{C_i N_i^2}}{\left(\frac{\sigma_i^2}{N_i}\right)^{\frac{3}{2}}} \\ &= \exp\left(-\frac{\delta_{b,j}^2}{2\left(\frac{\sigma_j^2}{N_j}\right)}\right) \cdot \frac{\frac{\delta_{b,j}\sigma_j^2}{C_j N_j^2}}{\left(\frac{\sigma_j^2}{N_j}\right)^{3/2}} \end{aligned}$$

Which gives us that

$$\exp\left(\frac{1}{2}\left(\frac{\delta_{b,j}^2}{\left(\frac{\sigma_j^2}{N_j}\right)} - \frac{\delta_{b,i}^2}{\left(\frac{\sigma_i^2}{N_i}\right)}\right)\right) \frac{C_j N_j^{\frac{1}{2}}}{C_i N_i^{\frac{1}{2}}} = \frac{\delta_{b,j}\sigma_i}{\delta_{b,i}\sigma_j}$$

Taking the log on both sides, we have

$$\begin{aligned} & \frac{\delta_{b,j}^2}{\sigma_j^2} N_j + \log(C_j^2 N_j) \\ &= \frac{\delta_{b,i}^2}{\sigma_i^2} N_i + \log(C_i^2 N_i) + 2\log\left(\frac{\delta_{b,j}\sigma_i}{\delta_{b,i}\sigma_j}\right) \end{aligned}$$

Multiply $C_i * C_j$ on both sides, we have

$$C_i \frac{\delta_{b,j}^2}{\sigma_j^2} C_j N_j + C_i C_j \log(C_j^2 N_j)$$

$$\begin{aligned} &= C_j \frac{\delta_{b,i}^2}{\sigma_i^2} C_i N_i + C_i C_j \log(C_i^2 N_i) \\ &\quad + 2C_i C_j \log\left(\frac{\delta_{b,j}\sigma_i}{\delta_{b,i}\sigma_j}\right) \end{aligned}$$

Define $\alpha_i = \frac{C_i N_i}{T}$ as the proportion of

time consumed for strategy i. then we have

$$\begin{aligned} & C_i \frac{\delta_{b,j}^2}{\sigma_j^2} \alpha_j + C_i C_j \log(C_j^2 N_j) \\ &= C_j \frac{\delta_{b,i}^2}{\sigma_i^2} \alpha_i + C_i C_j \log(C_i^2 N_i) \\ &\quad + 2C_i C_j \log\left(\frac{\delta_{b,j}\sigma_i}{\delta_{b,i}\sigma_j}\right) \end{aligned}$$

We consider the case when $T \rightarrow \infty$. in this case all the log term become much smaller than the linear term and are negligible. Therefore, we have

$$C_i \frac{\delta_{b,j}^2}{\sigma_j^2} \alpha_j = C_j \frac{\delta_{b,i}^2}{\sigma_i^2} \alpha_i$$

Which is

$$\frac{\alpha_i}{\alpha_j} = \frac{C_i}{C_j} \left(\frac{\left(\frac{\sigma_i}{\delta_{b,i}}\right)^2}{\left(\frac{\sigma_j}{\delta_{b,j}}\right)} \right), i, j \in \{1, 2, \dots, k\}, i \neq j \neq b$$

And

$$\frac{N_i}{N_j} = \left(\frac{\left(\frac{\sigma_i}{\delta_{b,i}}\right)^2}{\left(\frac{\sigma_j}{\delta_{b,j}}\right)} \right), i, j \in \{1, 2, \dots, k\}, i \neq j \neq b$$

Q.E.D.

Haitao Xiang received bachelor degree in theoretical physics in Sun Yat-sen University. He Obtained Ph.D degree from Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, and is now a postdoctoral research assistant in Oxford University. His research interests include Financial Economics, Energy Economics, Machine learning, Optimal Control and Reinforcement Learning in energy and financial systems.

Jianwu Lin received dual Bachelor degree in Engineering and Economics and Master of Engineering in Tsinghua University and Ph.D. and dual master degree in University of Pennsylvania.

Dr. Lin is a finance professor in Graduate School at Shenzhen, Tsinghua University. His research interests include Financial Technology, Financial Engineering, Meta learning, Machine Learning, Behavioral Finance, Quantitative Investment and Quantitative, Financial Risk Management, Internet Finance and Supply Chain Finance.

Chun-hung Chen received bachelor degree in Control Engineering in National Chiao-Tung University and Master degree in Decision and Control in National Taiwan University and Ph.D. from Harvard University.

Dr. Chen is a Professor of Systems Engineering & Operations Research in George Mason

University. He has worked on the development of highly efficient methodology for stochastic simulation optimization and its applications to air transportation system, semiconductor manufacturing, healthcare, security network, power grids, and missile defense system.

Ying Kong received bachelor degree in Theoretical Physics in Peking University and Ph.D. degree from Carleton University.

Dr. Kong is a Professor in Graduate School at Shenzhen, Tsinghua University and a Principal Investigator of the Lab of Low Carbon Economy and Financial Risk Analysis, Tsinghua-Berkeley Shenzhen Institute, Tsinghua University. His research interests includes Low Carbon Economics, Environmental Economics, Health Economics and Industrial Economics.