JMLR: Workshop and Conference Proceedings 39:315–329, 2014 ACML 2014

A UCB-Like Strategy of Collaborative Filtering

Atsuyoshi Nakamura ATSU@MAIN.IST.HOKUDAI.AC.JP
Hokkaido University, Kita 14, Nishi 9, Kita-ku, Sapporo 060-0814, Japan

Editor: Dinh Phung and Hang Li

Abstract

We consider a direct mail problem in which a system repeats the following process everyday during some period: select a set of user-item pairs (u, i), send a recommendation mail of item i to user u for each selected pair (u, i), and receive a response from each user. We assume that each response can be obtained before the next process and through the response, the system can know the user's evaluation of the recommended item directly or indirectly. Each pair (u, i) can be selected at most once during the period. If the total number of selections is very small compared to the number of entries in the whole user-item matrix, what selection strategy should be used to maximize the total sum of users' evaluations during the period? We consider a UCB-like strategy for this problem, and show two methods using the strategy. The effectiveness of our methods are demonstrated by experiments using synthetic and real datasets.

Keywords: bandit problem, online learning, collaborative filtering, recommender systems

1. Introduction

Assume that you have a web site for members and are planning a promotion campaign of new digital content service, in which you send an email of content's recommendation to a part of members every day during the period. Each email contains a link to the selling site of the recommended content, and the information of user's site access and content purchase through the link can be obtained. The goal is to maximize the profit of selling through the recommendation link during the campaign period under the condition that the number of user-content pairs recommended each day is restricted to some number ℓ. To achieve the goal, what set of users should be chosen and what content should be recommended to each chosen user on the ith day of the n-day period for $i = 1, 2, ..., n$? We call this problem a *direct mail problem*.

In this paper, we consider collaborative filtering approach to this problem, in which information used for learning user's preference is user's response alone and no feature of users and contents is used, where each user's response can be converted to a profit value represented by a real number.

The direct mail problem is a *bandit problem* (Auer et al., 2002), in which profit value r of content j for user i can be obtained only through i's response to the recommendation of j. This means that each recommendation must be done taking into account not only maximizing its profit but also obtaining the best training data for larger future profit. In batch learning *collaborative filtering* (Goldberg et al., 1992), training data is given and maximizing the next recommendation profit only is considered. In *active learning* (Jin and Si, 2004), obtaining the best training data for larger future profit alone is counted. In general,

the best data for learning does not coincide with the data with the highest rating, so exploration (necessary for learning) and exploitation (necessary for maximization of the next profit) must be balanced in a bandit problem.

In stochastic setting, the most successful bandit algorithm is the UCB (Upper Confidence Bound) algorithm proposed by Auer et al. (2002). The UCB algorithm uses the upper limit of a confidence interval of an estimated profit as a selection index instead of the estimated profit itself to balance exploration with exploitation. In this paper, as a UCB-like index of a random variable X whose Posterior distribution is D, we use $E_D(X) + \alpha\sqrt{V_D(X)}$, where $E_D(X)$ and $V_D(X)$ are the mean and variance of X with respect to D and α is a parameter that balance exploration with exploitation. Concretely speaking, we consider a stochastic matrix factorization model in which rating R_{ij} of item (content) j by user i is assumed to be generated according to a normal distribution with mean $U_i^\intercal V_j$, which is the inner product of two latent vectors U_i and V_j. From 0-mean priors of U_i and V_j, and observations of R_{ij} for some (i, j), posteriors of U_i and V_j are approximated by normal distributions. By assuming independence of U_i and V_j, the mean and variance of $U_i^\intercal V_j$ with respect to the approximated posteriors can be calculated. As methods for posterior approximation, we propose two methods: approximation by variational bayes (VB) (Lim and Teh, 2007) and approximation by probabilistic matrix factorization (PMF) (Mnih and Salakhutdinov, 2008).

According to our simulation results of the direct mail problem using one synthetic and two real datasets, our bandit method by VB approximation with an appropriate value of α outperformed VB. Our bandit method by PMF approximation also performed better than PMF for all but one of the datasets. For one very biased real dataset, simple selection of items with the highest average of observed ratings performed better than our bandit methods, but the VB-approximate bandit method performed best for the synthetic and the other real datasets. These results demonstrates effectiveness of our bandit methods for the direct mail problem.

This paper is organized as follows. In the rest of this section, we describe work related to our study. Basic stochastic matrix factorization is introduced in Sec. 2, and the direct mail problem is defined in Sec. 3. We propose a UCB-like strategy for the problem in Sec. 4 and two approximation methods necessary to use the strategy are explained in Sec. 5 and Sec. 6. The relation between the two approximations are discussed in Sec. 7. The effectiveness of our UCB-like strategy is empirically demonstrated through a simulation of the direct mail problem using synthetic and real datasets in Sec. 8. Our conclusion and future work are described in Sec. 9.

Summary of Contributions

- Our proposed method is the first bandit collaborative filtering method that deterministically selects user-item pairs using an index which depends on both the covariance matrices of the posterior distributions of latent user and item vectors. The method using Thompson sampling proposed by Zhao et al. (2013) is not deterministic method that was used to select items for a user as a solution of the new user problem.

- As methods to obtain something close to those covariance matrices, we proposed approximation methods using VB and PMF, which enabled the implementation of the above bandit collaborative method.

1.1. Related Work

Researches on both *bandit problem* (Auer et al., 2002) and *collaborative filtering* (Goldberg et al., 1992) are very popular and a lot of work has been done so far. Recently, Zhao et al. (2013) proposed *iterative collaborative filtering* which models the bandit-problem aspect of collaborative filtering: iterations of recommendation and rating feedback while balancing exploration for learning user's preference and exploitation for maximization of the next feedbacked rating. Though our direct mail problem is also a kind of iterative collaborative filtering, they mainly considered user-centric scenario and assumed that the latent feature vector of each item was already well-learnt while both the latent feature vectors of users and items are treated equally and must be learnt in our setting. They dealt with user-centric scenario because the main target of their study is *cold-start problem* in which recommender system must recommend a new user some items that already have ratings enough. Under the assumption that item feature vectors are already well-learnt, PMF (Mnih and Salakhutdinov, 2008) becomes ridge regression and its UCB-like version *LinUCB* (Li et al., 2010) is applicable. In direct mail problem, however, LinUCB cannot be used because such assumption does not seem appropriate. Among the bandit methods applied to iterative collaborative filtering (Zhao et al., 2013), only *Thompson sampling* (Chapelle and Li, 2011) is applicable to direct mail problem instead of our UCB-like strategy though the performance variance may become large.

2. Basic Model of Stochastic Matrix Factorization

Assume that there are m users and n items, and users rate some items by real value. Let $\mathcal{U} = \{1, 2, ..., m\}$ be the set of user ids and let $\mathcal{V} = \{1, 2, ..., n\}$ be the set of item ids. The users' ratings are represented by a partially observable $m \times n$ matrix R, whose (i, j)-entry value is the rating of item $j \in \mathcal{V}$ by user $i \in \mathcal{U}$, and the basic collaborative filtering task is to predict the unknown entry values from the known entry values.

One of the most popular prediction method is *matrix factorization* (Koren et al., 2009) in which R is approximated by a product of $m \times k$ matrix $U^\intercal$ and $k \times n$ matrix V, where k is a small natural number and $U^\intercal$ denotes the transposed matrix of U. In this approach, the task is to find matrices U and V such that the observable entry values of R are fit to the corresponding entry values of $U^\intercal V$. Since the (i, j)-entry value R_{ij} of R is predicted by the inner product of two vectors U_i and V_j, which are the ith and jth columns of U and V, respectively, this task can be seen as the task to find a vector U_i for each user i and a vector V_j for each item j from the observable entry values of R.

Through this paper, we adopt a basic model of stochastic matrix factorization, which is represented by the graphical model shown in Figure 1. In this model, the vector U_i for each user i and the vector V_j for each item j are assumed to be independently generated according to distributions $D_U(\Theta_U)$ and $D_V(\Theta_V)$, respectively, where Θ_U and Θ_V are lists of the parameters of the distributions.

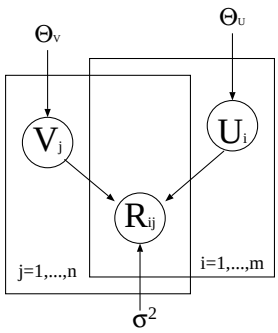


Figure 1: Basic model of stochastic matrix factorization

Given a vector U_i for user i and a vector V_j for item j, we assume that the rating R_{ij} is generated according to the normal distribution with mean $U_i^\intercal V_j$ and variance σ^2, that is,

$$R_{ij}|U_i, V_j \sim \mathcal{N}(U_i^\intercal V_j, \sigma^2).$$

3. Direct Mail Problem

We consider a kind of a recommendation problem as follows. There are m users and n items. Every day during h-day period, we select ℓ user-item pairs (i, j) from mn-sized set $\mathcal{U} \times \mathcal{V}$ and send user i an email with recommendation of item j for each selected pair (i, j). The same user must not be selected more than once on the same day but the same item can[1]. User's feedback to the recommendation can be obtained as a form of its rating for each email within the recommended day. (In real situation, a user's behavior such as clicking the link and buying the item may be converted to a rating.) The objective is maximization of the sum of ℓh ratings that are feedbacked to the recommendations. The total number ℓh of recommendations is assumed to be vary small compared to the number mn of all the user-item pairs. In this paper, we call this problem a *direct mail problem*.

4. UCB-like Strategy for Direct Mail Problem

We consider the following UCB-like strategy for a direct mail problem. Assume the stochastic matrix factorization model in Sec. 2. Let r_{ij} denote the observed value of R_{ij} and let O denote the set of observations (i, j, r_{ij}) obtained so far. Then, the posterior distributions $D_U^i(\Theta_U, O)$ of U_i can be different depending on user i and the posterior distributions $D_V^j(\Theta_V, O)$ of V_j can be also different depending on item j.

Let $\mathbf{u}_i, \mathbf{v}_j$ denote the means of U_i, V_j and let $\Sigma_{U,i}, \Sigma_{V,j}$ denote the covariance matrices of U_i, V_j for the posterior distributions $D_U^i(\Theta_U, O)$ and $D_V^j(\Theta_V, O)$, respectively. Then, under the assumption that U_i and V_j are independent from each other on their posterior

1. We add this restriction because too many recommendations to the same person at the same time is trivially undesirable. Some relaxation of this restriction, however, may improve recommendation performance.

distributions,

$$\begin{aligned} E(U_i^\intercal V_j) &= \mathbf{u}_i^\intercal \mathbf{v}_j \text{ and} \\ V(U_i^\intercal V_j) &= \mathrm{Tr}(\Sigma_{U,i}^\intercal \Sigma_{V,j} + \Sigma_{U,i}^\intercal \mathbf{v}_j \mathbf{v}_j^\intercal + \mathbf{u}_i \mathbf{u}_i^\intercal \Sigma_{V,j}) \end{aligned}$$

hold, where $\mathrm{Tr}(\cdot)$ denotes the trace of a square matrix '$\cdot$'. Unfortunately, the possibility that the above independence assumption holds is little because an observation (i, j, r_{ij}) relates U_i with V_j. So, we calculate $E(U_i^\intercal V_j)$ and $V(U_i^\intercal V_j)$ using the above expression not for the exact joint posterior distribution of U_i and V_j, but for its approximation in which the distributions of U_i and V_j are independent from each other.

The UCB (Upper Confidence Bound) strategy (Auer et al., 2002) uses the upper limit of a confidence interval of an estimated rating as its selection index so as to increase the chance of selecting an item with a small number of its past selections that causes a wide confidence interval of its estimated rating. Applying this idea to the posterior distributions of U_i and V_j, we propose the following index using $E(U_i^\intercal V_j)$ and $V(U_i^\intercal V_j)$ as a selection index of direct mail problem:

$$\mathbf{u}_i^\intercal \mathbf{v}_j + \alpha \sqrt{\mathrm{Tr}(\Sigma_{U,i}^\intercal \Sigma_{V,j} + \Sigma_{U,i}^\intercal \mathbf{v}_j \mathbf{v}_j^\intercal + \mathbf{u}_i \mathbf{u}_i^\intercal \Sigma_{V,j})}, \tag{1}$$

where α is the parameter that balance exploration and exploitation.

Note that $\Omega(k^2)$ time is necessary for calculation of Index (1) while only $O(k)$ time is enough for the simple prediction by the inner product $\mathbf{u}_i^\intercal \mathbf{v}_j$. This difference, however, is not significant when k is small.

5. Approximation by Variational Bayes

In order to use a UCB-like strategy, we have to find an approximation of the joint posterior distribution of U_i and V_j in which the distributions of U_i and V_j are mutually independent. This can be done by applying variational bayesian approach to stochastic matrix factorization (Lim and Teh, 2007). For self-containedness, we explain the detail of the method in the following.

In Figure 1, let $\Theta_U = (\sigma_U^2)$ and $\Theta_V = (\sigma_V^2)$, and let $D_U(\Theta_U) = \mathcal{N}(\mathbf{0}, \sigma_U^2 I)$ and $D_V(\Theta_V) = \mathcal{N}(\mathbf{0}, \sigma_V^2 I)$, that is, assume that each U_i and V_j are generated according to normal distributions with mean $\mathbf{0}$ and covariances $\sigma_U^2 I$ and $\sigma_V^2 I$, respectively, where I denotes the $k \times k$ identity matrix.

Let $f(U, V|O, \sigma^2, \sigma_U^2, \sigma_V^2)$ denote the probability density function of a pair of matrices U and V. Consider the problem of finding probability density functions $g(U)$ and $g(V)$ whose product $g(U)g(V)$ approximates $f(U, V|O, \sigma^2, \sigma_U^2, \sigma_V^2)$. In variational bayesian approach, such $g(U)$ and $g(V)$ can be obtained by minimizing free energy

$$\mathcal{F}(g(U)g(V)) = E_{g(U)g(V)}\left[\ln \frac{g(U)g(V)}{f(U, V, O|\sigma^2, \sigma_U^2, \sigma_V^2)}\right],$$

which means minimizing Kullback-Leibler divergence

$$\mathrm{KL}(g(U)g(V)||f(U, V|O, \sigma^2, \sigma_U^2, \sigma_V^2)) = E_{g(U)g(V)}\left[\ln \frac{g(U)g(V)}{f(U, V|O, \sigma^2, \sigma_U^2, \sigma_V^2)}\right]$$

between the two distributions. The free energy can be written as

$$\begin{aligned}\mathcal{F}(g(U)g(V)) &= E_{g(U)}[\ln g(U)] + E_{g(V)}[\ln g(V)] - \frac{1}{2\sigma^2}\sum_{(i,j)\in O} E_{g(U)g(V)}[(r_{ij} - U_i^\intercal V_j)^2] \\ &\quad -\frac{1}{2\sigma_U^2}\sum_{i=1}^{m} E_{g(U)}[||U_i||^2] - \frac{1}{2\sigma_V^2}\sum_{j=1}^{n} E_{g(V)}[||V_j||^2] \\ &\quad -\frac{|O|}{2}\ln 2\pi\sigma^2 - \frac{km}{2}\ln 2\pi\sigma_U^2 - \frac{kn}{2}\ln 2\pi\sigma_V^2.\end{aligned}$$

Unfortunately, it is not known an efficient way of calculating distributions $g(U)$ and $g(V)$ that attain the global minimum of $\mathcal{F}(g(U)g(V))$. But we can efficiently calculate distributions $g(U)$ and $g(V)$ that attain one of its local minima by alternating the following two steps until convergence.

g(U)-Optimization Step

By optimizing $g(U)$ with fixed $g(V)$ subject to $\int g(U)dU = 1$, we obtain

$$g(U) \propto \prod_{i=1}^{m} \exp\left(-\frac{1}{2}(U_i - \mathbf{u}_i)^\intercal \Sigma_{U,i}^{-1}(U_i - \mathbf{u}_i)\right),$$

where

$$\Sigma_{U,i} = \left(\frac{1}{\sigma^2}\sum_{(i,j,r_{ij})\in O}(\Sigma_{V,j} + \mathbf{v}_j\mathbf{v}_j^\intercal) + \frac{1}{\sigma_U^2}I\right)^{-1} \text{ and} \tag{2}$$

$$\mathbf{u}_i = \Sigma_{U,i}\sum_{(i,j,r_{ij})\in O}\frac{r_{ij}\mathbf{v}_j}{\sigma^2}. \tag{3}$$

Here, $\mathbf{v}_j$ and $\Sigma_{V,j}$ are the mean and the covariance matrix of V_j for the fixed distribution $g(V)$.

g(V)-Optimization Step

By optimizing $g(V)$ with fixed $g(U)$ subject to $\int g(V)dV = 1$, we obtain

$$g(V) \propto \prod_{j=1}^{n} \exp\left(-\frac{1}{2}(V_j - \mathbf{v}_j)^\intercal \Sigma_{V,j}^{-1}(V_j - \mathbf{v}_j)\right),$$

where

$$\Sigma_{V,j} = \left(\frac{1}{\sigma^2}\sum_{(i,j,r_{ij})\in O}(\Sigma_{U,i} + \mathbf{u}_i\mathbf{u}_i^\intercal) + \frac{1}{\sigma_V^2}I\right)^{-1} \text{ and} \tag{4}$$

$$\mathbf{v}_j = \Sigma_{V,j}\sum_{(i,j,r_{ij})\in O}\frac{r_{ij}\mathbf{u}_i}{\sigma^2}. \tag{5}$$

Here, $\Sigma_{U,i} = V_{g(U)}[U_i]$ and $\mathbf{u}_i = E_{g(U)}[U_i]$ for the fixed $g(U)$.

Let $(\mathbf{u}_i^*, \Sigma_{U,i}^*)$ $(i \in \mathcal{U})$ and $(\mathbf{v}_j^*, \Sigma_{V,j}^*)$ $(j \in \mathcal{V})$ be the converged parameters of $g(U)$ and $g(V)$. In the obtained distributions $g(U)$ and $g(V)$, each U_i and V_j are independent from each other and their distributions are normal distributions $\mathcal{N}(\mathbf{u}_i^*, \Sigma_{U,i}^*)$ and $\mathcal{N}(\mathbf{v}_j^*, \Sigma_{V,j}^*)$, respectively. Thus, we can calculate UCB-like indeces (1) using those means and covariance matrices.

In variational bayesian approach, we can also optimize parameters σ_U^2, σ_V^2 and σ^2 for fixed $g(U) = \prod_{i=1}^m \mathcal{N}(\mathbf{u}_i, \Sigma_{U,i})$ and $g(V) = \prod_{j=1}^n \mathcal{N}(\mathbf{v}_j, \Sigma_{V,j})$:

$$\sigma_U^2 = \frac{1}{km} \sum_{i=1}^m \left(\mathrm{Tr}(\Sigma_{U,i}) + \mathbf{u}_i^\intercal \mathbf{u}_i \right), \tag{6}$$

$$\sigma_V^2 = \frac{1}{kn} \sum_{j=1}^n \left(\mathrm{Tr}(\Sigma_{V,j}) + \mathbf{v}_j^\intercal \mathbf{v}_j \right) \text{ and} \tag{7}$$

$$\sigma^2 = \frac{1}{|O|} \sum_{(i,j,r_{ij}) \in O} \left(r_{ij}^2 - 2 r_{ij} \mathbf{u}_i^\intercal \mathbf{v}_j + \mathrm{Tr}[(\Sigma_{U,i} + \mathbf{u}_i \mathbf{u}_i^\intercal)(\Sigma_{V,j} + \mathbf{v}_j \mathbf{v}_j^\intercal)] \right). \tag{8}$$

The above estimations seems reasonable because σ_U^2, σ_V^2 and σ^2 are estimated by

$$\frac{\sum_{i=1}^m E_{g(U)}[||U_i||^2]}{km}, \frac{\sum_{j=1}^n E_{g(V)}[||V_j||^2]}{kn} \text{ and } \frac{\sum_{(i,j,r_{ij}) \in O} E_{g(U)g(V)}[(r_{ij} - U_i^\intercal V_j)^2]}{|O|},$$

respectively.

6. Approximation by PMF

In probabilistic matrix factorization (PMF) (Mnih and Salakhutdinov, 2008), U and V are estimated using MAP (Maximum A Posteriori) estimation. We also explain the details of the method in the following for the sake of self-containedness.

Assume that the prior distributions of U_i and V_j are $\mathcal{N}(\mathbf{0}, \sigma_U^2 I)$ and $\mathcal{N}(\mathbf{0}, \sigma_V^2 I)$, respectively, for all $i \in \mathcal{U}$ and $j \in \mathcal{V}$. Given a set of observations O, consider the posterior probability density function

$$f(U, V | O, \sigma^2, \sigma_U^2, \sigma_V^2) \propto \prod_{(i,j,r_{ij}) \in O} \frac{e^{-\frac{(r_{ij} - U_i^\intercal V_j)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} \prod_{i=1}^m \frac{e^{-\frac{||U_i||^2}{2\sigma_U^2}}}{(2\pi\sigma_U^2)^{k/2}} \prod_{j=1}^n \frac{e^{-\frac{||V_j||^2}{2\sigma_V^2}}}{(2\pi\sigma_V^2)^{k/2}}.$$

Let (U^*, V^*) be the estimated values of (U, V) by MAP estimation, then

$$(U^*, V^*) = \underset{(U,V)}{\mathrm{argmin}} \left(\sum_{(i,j) \in O} \frac{(r_{ij} - U_i^\intercal V_j)^2}{2\sigma^2} + \sum_{i=1}^m \frac{||U_i||^2}{2\sigma_U^2} + \sum_{j=1}^n \frac{||V_j||^2}{2\sigma_V^2} \right)$$

holds. We can easily see that both the $f(U | V^*, O, \sigma^2, \sigma_U^2, \sigma_V^2)$ and $f(V | U^*, O, \sigma^2, \sigma_U^2, \sigma_V^2)$ are normal distributions and their covariance matrices can be calculated. Thus, we can use the UCB-like selection index (1) with the assumption that

$$f(U, V | O, \sigma^2, \sigma_U^2, \sigma_V^2) = f(U | V^*, O, \sigma^2, \sigma_U^2, \sigma_V^2) f(V | U^*, O, \sigma^2, \sigma_U^2, \sigma_V^2)$$

holds approximately.

Though no efficient way of calculating the MAP estimation (U^*, V^*) is known, we can efficiently obtain matrices U and V that attain one of the local maxima of $f(U, V|O, \sigma^2, \sigma_U^2, \sigma_V^2)$ by alternating least square method, which alternates the following two steps until convergence.

U-Optimization Step

By calculating the optimal value $(\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_m)$ of U with fixed $V = (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n)$, we obtain

$$f(U|(\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n), O, \sigma^2, \sigma_U^2, \sigma_V^2) \propto \prod_{i=1}^{m} \exp\left(-\frac{1}{2}(U_i - \mathbf{u}_i)^\intercal \Sigma_{U,i}^{-1}(U_i - \mathbf{u}_i)\right),$$

where

$$\Sigma_{U,i} = \left(\frac{1}{\sigma^2}\sum_{(i,j,r_{ij})\in O} \mathbf{v}_j\mathbf{v}_j^\intercal + \frac{1}{\sigma_U^2}I\right)^{-1} \text{ and} \tag{9}$$

$$\mathbf{u}_i = \Sigma_{U,i}\sum_{(i,j,r_{ij})\in O} \frac{r_{ij}\mathbf{v}_j}{\sigma^2}. \tag{10}$$

V-Optimization Step

By calculating the optimal value $(\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n)$ of V with fixed $U = (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_m)$, we obtain

$$f(V|(\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_m), O, \sigma^2, \sigma_U^2, \sigma_V^2) \propto \prod_{j=1}^{n} \exp\left(-\frac{1}{2}(V_j - \mathbf{v}_j)^\intercal \Sigma_{V,j}^{-1}(V_j - \mathbf{v}_j)\right),$$

where

$$\Sigma_{V,j} = \left(\frac{1}{\sigma^2}\sum_{(i,j,r_{ij})\in O} \mathbf{u}_i\mathbf{u}_i^\intercal + \frac{1}{\sigma_V^2}I\right)^{-1} \text{ and} \tag{11}$$

$$\mathbf{v}_j = \Sigma_{V,j}\sum_{(i,j,r_{ij})\in O} \frac{r_{ij}\mathbf{u}_i}{\sigma^2}. \tag{12}$$

7. Relation between the Two Approximations

Though the derivations of the approximations by variational bayes and PMF are different, the methods to calculate a locally optimal solution are quite similar. In fact, the both methods alternate the calculation of the mean $\mathbf{u}_i$ and covariance matrix $\Sigma_{U,i}$ of a normal distribution over a vector U_i for fixed V's distribution or V, with the calculation of the mean $\mathbf{v}_j$ and covariance matrix $\Sigma_{V,j}$ of a normal distribution over a vector V_j for fixed U's distribution or U until convergence. Furthermore, the ways of calculating those values are almost the same; Eq. (3) and (5) are exactly the same as Eq. (10) and (12), and Eq. (2) can be obtained from Eq. (9) only by replacing $\mathbf{v}_j\mathbf{v}_j^\intercal$ with $\Sigma_{V,j} + \mathbf{v}_j\mathbf{v}_j^\intercal$, and Eq. (4) can be also obtained from Eq. (11) similarly.

In the case with known V, that is, in the case that $V_j \sim \mathcal{N}(\mathbf{v}_j, \mathbf{0})$, where $\mathbf{0}$ is the 0-matrix of $k \times k$, both the approximation methods become the simple least square method. In such case, UCB-like selection index (1) becomes

$$\mathbf{u}_i^\intercal \mathbf{v_j} + \alpha \sqrt{\mathbf{v}_j^\intercal \Sigma_{U,i} \mathbf{v}_j}.$$

The bandit algorithm using this selection index is known as LinUCB (Li et al., 2010) in which $\sigma^2 = \sigma_U^2 = 1$ is used.

Instead of UCB-like method using Index (1), we can use Thompson sampling (Chapelle and Li, 2011). In fact, Zhao et al. (2013) proposed such a method using the approximation by PMF though, in their experiments, it was outperformed by LinUCB with MAP-estimated fixed latent item vectors in the new user setting. We cannot deny the possibility that Thompson sampling performs well in our setting, but its performance is expected to vary more.

8. Experiments

8.1. Experimental Setting

We conducted experiments to check the effectiveness of UCB-like strategy for direct mail problem using synthetic and real datasets.

We used the following three datasets.

SYN: A synthetic dataset generated as follows. First, we independently generated $U_i \in \mathbb{R}^5$ and $V_j \in \mathbb{R}^5$ for all $i = 1, 2, ..., 1000$ and $j = 1, 2, ..., 1000$ according to $\mathcal{N}(\mathbf{0}, I)$, where I is a 5×5 identity matrix. Then, R_{ij} is generated according to $\mathcal{N}(U_i^\intercal V_j, 1)$ for all $i = 1, 2, ..., 1000$ and $j = 1, 2, ..., 1000$. Thirty matrices R are generated by giving different seeds to a random number generator for sampling from normal distributions.

Jester: Joke rating dataset collected between April 1999 - May 2003 by University of California, Berkeley[2]. Rating scales are real values between -10 and 10. The number of jokes are 100, and the dataset contains 14,116 users with no missing rating. We used the $14{,}116 \times 100$ matrix R for such perfectly-rating users in our experiments.

LibimSeTi: Dataset of dating service called LibimSeTi[3] dumped on April 4, 2006. Though the original rating scales are $\{1, 2, ..., 10\}$, we shifted them by -5.5, that is, shifted to $\{-4.5, -3.5, ..., 4.5\}$. We made a no-missing-entry matrix R from the original sparse rating matrix by repeatedly deleting a row or column with the largest number of missing entries. The matrix R used in the experiment is a 120×93 matrix which is composed of ratings of 93 items rated by 120 users.

In the experiment, an initial set O_0 of observations is given to a recommendation algorithm first. We select O_0 so as to make it contain at least one entry of every row and at least one entry of every column. Such a selection with the smallest number of entries is done by the procedure in Figure 2.

2. http://eigentaste.berkeley.edu/dataset/
3. http://www.occamslab.com/petricek/data/

$O_0 \leftarrow \emptyset$
if $m > n$ **then**
  for $i = 1, 2, ..., m$ **do**
    Randomly select item j among the items with the least number of observations in O_0, that is,

$$j = \underset{j' \in \{1,2,...,n\}}{\operatorname{argmin}} |\{i' : (i', j', r_{i'j'}) \in O_0\}|,$$

    where $|\cdot|$ is the number of elements in '$\cdot$'.
    $O_0 \leftarrow (i, j, r_{ij})$
else
  for $j = 1, 2, ..., n$ **do**
    Randomly select user i among the users with the least number of observations in O_0, that is,

$$i = \underset{i' \in \{1,2,...,m\}}{\operatorname{argmin}} |\{j' : (i', j', r_{i'j'}) \in O_0\}|.$$

    $O_0 \leftarrow (i, j, r_{ij})$

Figure 2: Selection procedure of initial set O_0 of observations

1. Set $O = O_0$, where O_0 is a set of observations that is selected by the procedure in Figure 2.
2. Repeat the following round 100 times.
   (a) Update the selection indeces using the current set O of observations.
   (b) For each user i, find item j_i with the maximum selection index among the elements of $\{j : (i, j, r_{ij}) \notin O\}$.
   (c) Select the top 5% user-item pairs (i, j_i) with the largest selection index from $\{(i, j_i) : i \in \mathcal{U}\}$.
   (d) Recommend item j_i to user i for all the selected pairs (i, j_i), and receive rating r_{ij_i} as its feedback.
   (e) Add the triplets (i, j_i, r_{ij_i}) to O for all the selected pairs (i, j_i).

Figure 3: Recommendation process simulation for performance evaluation

Recommendation process using each algorithm was simulated by the procedure shown in Figure 3. The process is composed of 100 rounds and single item recommendation is done to the selected 5% users in each round. The selection of user-item pairs are done by a recommendation algorithm based on the set O of observations so far. For each user, selection indeces for all the items whose ratings have not been observed so far are calculated and the item with the highest index is selected as a recommendation candidate. Among all the candidates, l of them are selected and the recommendation is done for the selected user-item pairs. We set l to 5% of the number of users, that is, $l = 0.05m$. For all the selected

Table 1: Statistics and the average ratings of the five methods for each dataset.

Dataset		SYN	Jester	LibimSeTi
#user×#item		1000×1000	14116×100	120×93
(Shifted) range		$[-21.1, 19.5]$	$[-10, 10]$	$[-4.5, 4.5]$
Average		0.00	1.03	0.12
Observed-rate	Initial O	0.1%	1%	1.1%
	Final O	0.6%	6%	6.5%
Average rating (95% confidence interval)	UCBVB	**3.44**($\pm$0.08) $\langle\alpha = 1.25\rangle$	**5.45**($\pm$0.01) $\langle\alpha = 0.5625\rangle$	3.24($\pm$0.07) $\langle\alpha = 0.5\rangle$
	VB	2.88($\pm$0.06)	5.26($\pm$0.02)	3.07($\pm$0.08)
	UCBPMF	**3.41**($\pm$0.09) $\langle\alpha = 0.1875\rangle$	4.31($\pm$0.03) $\langle\alpha = 0.00195312\rangle$	3.57($\pm$0.23) $\langle\alpha = 0.375\rangle$
	PMF	3.19($\pm$0.07)	4.34($\pm$0.03)	2.96($\pm$0.10)
	POP	0.02($\pm$0.01)	3.46($\pm$0.02)	**4.44**($\pm$0.01)

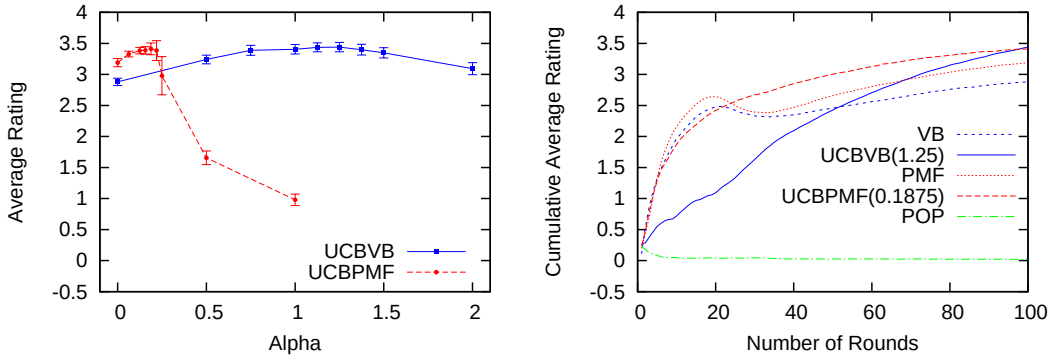
pairs (i, j), the triplets (i, j, r_{ij}) with ratings r_{ij} are added to O. Note that observed-rates of the whole matrices by the initial and final O for each dataset are less than 1.1% and 6.5%, respectively. (See Table 1.)

To check the effectiveness of the UCB-like strategy, we ran five algorithms. Two matrix factorization algorithms VB (Variational Bayes) and PMF, and their UCB-like versions UCBVB and UCBPMF, respectively, were executed. Note that algorithms UCBVB and UCBPMF are equivalent to VB and PMF, respectively, when $\alpha = 0$ in Index (1). The dimension k of vectors U_i and V_j was fixed to 5 for all the datasets and for all the matrix factorization algorithms. The rest one is a very simple algorithm POP whose selection index of an item is the average of its ratings observed so far and the same for all users.

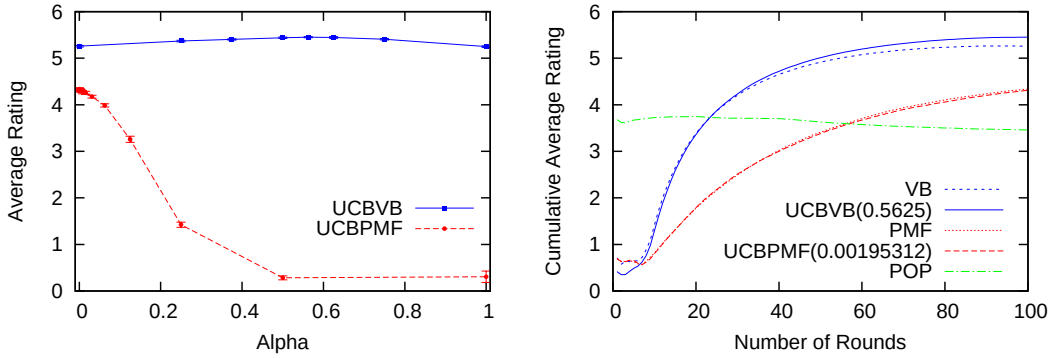
The recommendation performance was evaluated by average rating over all the recommended user-item pairs, and the learning curve for rounds was drawn using *cumulative* average rating, which is the average rating over all the recommended items so far.

The convergence of matrix factorization algorithms was judged by whether average Euclidean distance between current and previous 5-dimensional column vectors U_i and V_j is smaller than 0.001. The optimization of the parameters σ_U^2, σ_V^2 and σ^2 in variational bayes, which are calculated by Eqs. (6), (7) and (8), was also applied to PMF. These parameter optimization and the optimization of U and V were done alternately until the sum of absolute differences between current and previous parameters is less than 0.1. The cumulative average rating for each algorithm was averaged over thirty rating matrices for SYN dataset and also averaged over thirty randomly generated initial O for other datasets.

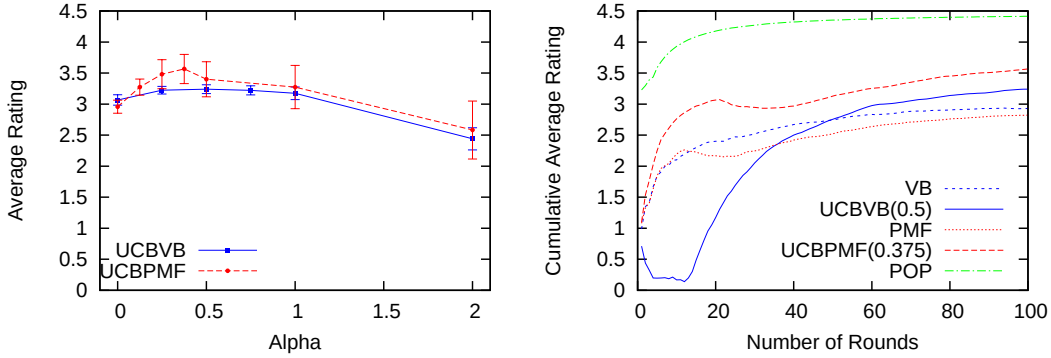
In performance comparison, we used empirically nearly optimal values of the exploration-exploitation balancing parameter α for the UCB-like strategies, which were found by Algorithm SearchOptAlpha described in Appendix A using $K = 30$ and $\gamma = 0.05$.



(*a*) SYN dataset



(*b*) Jester dataset



(*c*) LibimSeTi dataset

Figure 4: [Left] Curves of average rating averaged over thirty runs for the exploration-exploitation balancing parameter α. The plotted points were the searched points in Algorithm SearchOptAlpha described in Appendix A. Their 95% confidence intervals are also shown. [Right] Curves of cumulative average rating for rounds. The values are also averaged over thirty runs.

8.2. Results

The left figures in Figure 4 are the curves of average rating over all the recommendations for the values of α searched in Algorithm SearchOptAlpha. UCBPMF looks more sensitive to the value of the parameter α. Algorithm SearchOptAlpha found better α-value than 0 except for UCBPMF using Jester dataset, which means that UCBVB performed better than VB for all the three datasets and so does UCBPMF except for one dataset if appropriate values were set to α. The average ratings for the empirically-found nearly optimal α and their 95% confidence intervals are shown in in Table 1. You can see that the performance difference between VB and optimized UCBVB and that between PMF and optimized UCBPMF are statistically significant in all the case that the UCB-like strategy performed better. The right figures show the learning curves of five algorithms: VB, optimized UCBVB, PMF, optimized PMF and POP, where the learning curve is the curve of the cumulative average ratings for rounds. In early rounds, UCBVB performed worst among the four but its improvement was largest in the later rounds. The exploring tendency of UCBPMF in early stage was not so high compared with UCBVB.

In total, the performances of UCBVB and UCBPMF are comparable, but the usability of UCBVB seems better from the viewpoint of sensitivity to the parameter α.

As for comparison with POP, the four matrix factorization methods outperformed POP for SYN and Jester datasets, but POP performed best for LibimSeTi dataset. LibimSeTi is a very biased dataset; Among the 93 items, two items have the highest rating alone and the rating standard deviations of 12 items are less than 0.01. It is very natural that POP performs extremely well for such a biased dataset.

9. Conclusions

We proposed UCB-like methods of collaborative filtering using VB or PMF approximation for direct mail problem. According to our experimental results, the UCB-like methods are effective compared with original VB and PMF if we choose an appropriate exploration-exploitation balancing parameter. Especially, the UCB-like method using VB approximation stably performed well. Experimental performance comparison with active learning methods and Thompson sampling, and theoretical analyses of the proposed methods are our future work.

Acknowledgments

I would like to thank my former master-course student Tran Duc Toan, and my former internship students Sumit Raj, Tu Shitao and Jonathan Young for their investigation and experiments on this theme, which helped the progress of this study. I would also like to thank Professor Mineichi Kudo and anonymous reviewers for their useful comments. This work was partially supported by JSPS KAKENHI Grant Number 25280079.

References

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

O. Chapelle and L. Li. An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems 24*, pages 2249–2257, 2011.

D. Goldberg, Nichols, B. D. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

R. Jin and L. Si. A bayesian approach toward active learning for collaborative filtering. In *Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence*, pages 278–285, 2004.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

L. Li, W. Chu, J. Langford, and R. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web (WWW2010)*, pages 661–670, 2010.

Y. Lim and Y. Teh. Variational bayesian approach to movie rating prediction. In *Proceedings of KDD Cup and Workshop 7*, pages 15–21, 2007.

A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems 20*, pages 1257–1264, 2008.

X. Zhao, W. Zhang, and J. Wang. Interactive collaborative filtering. In *Proceedings of the 2nd ACM International Conference on Information and Knowledge Management (CIKM2013)*, pages 1411–1420, 2013.

Appendix A. Algorithm for Tuning α

Consider a Gaussian distribution family $\{\mathcal{N}(\mu(\alpha), \sigma^2(\alpha))|\alpha \in [0, \infty)\}$ parameterized by α, where μ, σ^2 are positive mean and variance functions. Assume that the mean function μ has a unique maximal point, and K samples of a random variable $X \sim \mathcal{N}(\mu(\alpha), \sigma^2(\alpha))$ can be obtained by sampling oracle $\textsc{Sampling}(\alpha, K)$ for any natural numbers K and any non-negative real number α. Under these assumptions, we use Algorithm SearchOptAlpha in Figure 5 to estimate the maximal point $\alpha_{\max}$ of the mean function μ in our experiment for tuning the exploration-exploitation balancing parameter α.

In Algorithm SearchOptAlpha, $\mu(\alpha)$ is estimated by the sample mean $\bar{x}(\alpha)$ over samples $x_1, x_2, ..., x_K$, which are obtained by oracle $\textsc{Sampling}(\alpha, K)$. Algorithm SearchOptAlpha is a kind of a binary search algorithm. First, the algorithm find the range $[\alpha_{\mathrm{L}}, \alpha_{\mathrm{U}}]$ that contains the maximal point under the assumption of a unique maximal point. This task is done by Function FindRange. Function FindRange tries to find a range $[\alpha_{\mathrm{L}}, \alpha_{\mathrm{U}}]$ in which a sample mean $\bar{x}(\alpha)$ at $\alpha \in (\alpha_{\mathrm{L}}, \alpha_{\mathrm{U}})$ is larger than $\bar{x}(\alpha_{\mathrm{L}})$ and $\bar{x}(\alpha_{\mathrm{U}})$ by doubling or halving α_{U}. It gives up to find such a range when the smaller one of $\bar{x}(\alpha_{\mathrm{L}})$ and $\bar{x}(\alpha_{\mathrm{U}})$ is larger than the lower limit of $100(1-\gamma)\%$ confidence interval of the larger one $\bar{x}(\alpha_{\max})$ of them, and in such a case FindRange outputs $\alpha_{\max}$ as a nearly optimal α, where $\gamma \in [0, 1]$ is a given confidence level. In the case that FindRange successes to find the range with an inner maximal point, the range is narrowed to the first half if the estimated mean function value

at the first quarter point is greater than that at the middle point, narrowed to the last half if the estimated mean function value at the third quarter point is greater than that at the middle point, and narrowed to the middle half otherwise, where the middle half is the range from the first quarter point to the third quarter point. Algorithm SearchOptAlpha stops when $100(1-\gamma)\%$ confidence interval by the sample mean estimator of $\mu(\alpha_{\max})$ at the middle point of the current range $[\alpha_{\mathrm{L}}, \alpha_{\mathrm{U}}]$ includes both of the values $\bar{x}(\alpha_{\mathrm{L}})$ and $\bar{x}(\alpha_{\mathrm{U}})$, and output $\alpha_{\max}$ as a nearly optimal α. Note that $100(1-\gamma)\%$ confidence interval by the sample mean estimator of $\mu(\alpha)$ over K samples is

$$\left[\bar{x}(\alpha) - t_{K-1}(\gamma/2)\sqrt{s^2(\alpha)/(K-1)}, \bar{x}(\alpha) + t_{K-1}(\gamma/2)\sqrt{s^2(\alpha)/(K-1)}\right],$$

where $t_{K-1}(\gamma/2)$ is the upper $100(\gamma/2)$ percentage point of Student's t-distribution with $K-1$ degree of freedom and $s^2(\alpha)$ is the sample variance over the K samples.

Algorithm SEARCHOPTALPHA
input: K: Number of samples, $\gamma \in [0,1]$: Confidence level
output: Estimated maximal point $\alpha_{\max}$

$(\alpha_{\mathrm{L}}, \alpha_{\mathrm{U}}, \alpha_{\max}, \bar{x}(\alpha_{\mathrm{L}}), \bar{x}(\alpha_{\mathrm{U}}), \bar{x}(\alpha_{\max}), s^2(\alpha_{\max})) \leftarrow$ FINDRANGE()
while $\bar{x}(\alpha_{\max}) - t_{K-1}(\gamma/2)\sqrt{s^2(\alpha_{\max})/(K-1)} > \min\{\bar{x}(\alpha_{\mathrm{L}}), \bar{x}(\alpha_{\mathrm{U}})\}$ **do**
  $\alpha_{\mathrm{LM}} \leftarrow (\alpha_{\mathrm{L}} + \alpha_{\max})/2$, $\alpha_{\mathrm{UM}} \leftarrow (\alpha_{\max} + \alpha_{\mathrm{U}})/2$
  $(\bar{x}(\alpha_{\mathrm{LM}}), s^2(\alpha_{\mathrm{LM}})) \leftarrow$ ESTIMATEFROMSAMPLES$(\alpha_{\mathrm{LM}}, K)$
  $(\bar{x}(\alpha_{\mathrm{UM}}), s^2(\alpha_{\mathrm{UM}})) \leftarrow$ ESTIMATEFROMSAMPLES$(\alpha_{\mathrm{UM}}, K)$
  if $\bar{x}(\alpha_{\mathrm{LM}}) > \bar{x}(\alpha_{\max})$ **then**
    $\alpha_{\mathrm{U}} \leftarrow \alpha_{\max}$, $\alpha_{\max} \leftarrow \alpha_{\mathrm{LM}}$
  else if $\bar{x}(\alpha_{\mathrm{UM}}) > \bar{x}(\alpha_{\max})$ **then**
    $\alpha_{\mathrm{L}} \leftarrow \alpha_{\max}$, $\alpha_{\max} \leftarrow \alpha_{\mathrm{UM}}$
  else
    $\alpha_{\mathrm{L}} \leftarrow \alpha_{\mathrm{LM}}$, $\alpha_{\mathrm{U}} \leftarrow \alpha_{\mathrm{UM}}$
return $\alpha_{\max}$

Function FINDRANGE()
$\alpha_{\mathrm{L}} \leftarrow 0.0$, $\alpha_{\mathrm{U}} \leftarrow 1.0$
$(\bar{x}(\alpha_{\mathrm{L}}), s^2(\alpha_{\mathrm{L}})) \leftarrow$ ESTIMATEFROMSAMPLES(α_{L}, K)
$(\bar{x}(\alpha_{\mathrm{U}}), s^2(\alpha_{\mathrm{U}})) \leftarrow$ ESTIMATEFROMSAMPLES(α_{U}, K)
if $\bar{x}(\alpha_{\mathrm{L}}) < \bar{x}(\alpha_{\mathrm{U}})$ **then**
  $\alpha_{\max} \leftarrow \alpha_{\mathrm{U}}$, $\alpha_{\mathrm{U}} \leftarrow 2\alpha_{\mathrm{U}}$
  $(\bar{x}(\alpha_{\mathrm{U}}), s^2(\alpha_{\mathrm{U}})) \leftarrow$ ESTIMATEFROMSAMPLES(α_{U}, K)
  while $\bar{x}(\alpha_{\max}) < \bar{x}(\alpha_{\mathrm{U}})$ **do**
    $\alpha_{\mathrm{L}} \leftarrow \alpha_{\max}$, $\alpha_{\max} \leftarrow \alpha_{\mathrm{U}}$
    if $\bar{x}(\alpha_{\max}) - t_{K-1}(\gamma/2)\sqrt{s^2(\alpha_{\max})/(K-1)} < \bar{x}(\alpha_{\mathrm{L}})$ **then break**
    $\alpha_{\mathrm{U}} \leftarrow 2\alpha_{\mathrm{U}}$
    $(\bar{x}(\alpha_{\mathrm{U}}), s^2(\alpha_{\mathrm{U}})) \leftarrow$ ESTIMATEFROMSAMPLES(α_{U}, K)
else
  $\alpha_{\max} \leftarrow \alpha_{\mathrm{U}}/2$
  $(\bar{x}(\alpha_{\max}), s^2(\alpha_{\max})) \leftarrow$ ESTIMATEFROMSAMPLES$(\alpha_{\max}, K)$
  while $\bar{x}(\alpha_{\max}) < \bar{x}(\alpha_{\mathrm{L}})$ **do**
    $\alpha_{\mathrm{U}} \leftarrow \alpha_{\max}$
    if $\bar{x}(\alpha_{\mathrm{L}}) - t_{K-1}(\gamma/2)\sqrt{s^2(\alpha_{\mathrm{L}})/(K-1)} < \bar{x}(\alpha_{\mathrm{U}})$ **then**
      $\alpha_{\max} \leftarrow \alpha_{\mathrm{L}}$, **break**
    $\alpha_{\max} \leftarrow \alpha_{\mathrm{U}}/2$
    $(\bar{x}(\alpha_{\max}), s^2(\alpha_{\max})) \leftarrow$ ESTIMATEFROMSAMPLES$(\alpha_{\max}, K)$
return $\alpha_L, \alpha_U, \alpha_{\max}, \bar{x}(\alpha_L), \bar{x}(\alpha_U), \bar{x}(\alpha_{\max}), s^2(\alpha_{\max})$

Function ESTIMATEFROMSAMPLES(α, K)
$x_1, x_2, ..., x_K \leftarrow$ SAMPLING(α, K)
$\bar{x} = \frac{1}{K}\sum_{i=1}^{K} x_i$
$s^2 = \frac{1}{K}\sum_{i=1}^{K}(x_i - \bar{x})^2$
return $\bar{x}, s^2$

Figure 5: Pseudocode of Algorithm SearchOptAlpha