LETTER

α-Bayesian Collaboration of Multiple Predictors and its Applications to Hybrid Recommendation and User Modeling

Jun-ichiro Hirayama[1], Masashi Nakatomi[2], Takashi Takenouchi[1] and Shin Ishii[1,3]

[1] Graduate School of Information Science, Nara Institute of Science and Technology,
Takayama 8916-5, Ikoma, Nara
{junich-h, ttakashi}@is.naist.jp

[2] Ricoh Company, Ltd.
nakatomi@rdc.ricoh.co.jp

[3] Graduate School of Informatics, Kyoto University
ishii@i.kyoto-u.ac.jp

(Submitted on December 19, 2007)

Abstract – In recent widespread areas of machine learning or data mining applications, one often should deal with multiple statistical prediction tasks simultaneously, each of which cannot be successful by itself due to limitation of data amount. Among several approaches to utilizing task relationship, a naive but still important approach is to separately train task-specific predictors in advance and then integrate them at the time of prediction. In this study, we propose a general framework of realizing such a "collaborative prediction" mechanism, specifically based on an existing generalization of Bayesian predictive distribution using the α-divergence. We also propose a novel hybrid method of collaborative and content-based recommendations, under the proposed framework. We demonstrate the effectiveness of the proposed method by using two kinds of real datasets; one is the Movielens benchmark collection related to recommendation, and the other is a real log dataset of shared electronic devices related to a particular problem of user modeling.

Keywords – Bayesian predictive distribution, Collaborative prediction, Movielens benchmark

1. Introduction

In real-world applications of machine learning or data mining methods, one often should deal with multiple statistical prediction tasks simultaneously. Traditional approaches usually treat each task separately from the others by ignoring the potential relationship of these tasks, where the prediction will be made for each individual task based only on the knowledge about the particular task. In recent widespread areas of such applications, however, there often be difficult situations where each single task cannot be successful by itself due to limitation of data sizes. To overcome such difficulties, utilizing potential relationship among the tasks has recently been appeared as one of the key issues.

One serious example of such situations is in the context of recommendation systems [1], where each user's explicit ratings over items are usually too lacking to know their true preferences over huge numbers of items. The popularly-used *collaborative filtering* (CF) [15, 5] methods predict the unknown ratings of a particular user according to the rating patterns of other users who seem to have similar preferences to the target user, assuming that users share some common preference (rating) patterns over items. In more general contexts of *user modeling* [16], which aims at predicting users' behaviors on target systems, there usually exists inactive users or new users who do not provide enough data, and thus the prediction about these users cannot be successful.

One simple idea to utilize the task relationship is to train multiple task's predictors simultaneously, with making them share the knowledge across the tasks in the training. Such approaches are actually used in model-based CF methods [10, 12, 13]. From a more general perspective, this is the basic idea of *multi-task learning* (MTL) [7, 4, 3, 17]. On the other hand, there is an alternative approach, in which the task-specific predictors are separately trained in advance, and then integrated at the time of prediction. We call this latter approach as *collaborative prediction*, or simply as *collaboration*, in the rest of this article. Conventional memory-based CF methods can be regarded as being in this category, even though they do not use any model explicitly. Some ensemble methods such as Bagging [6] are closely related to this approach, while they usually consider integration of multiple predictors each for a same task (but [18] has reported the usefulness even in a multi-task setting).

In this article, we propose a general framework of collaborative prediction, by extending our previous study [9]. In [9], we proposed a Bayesian formulation of collaborative prediction, with particular interests in its application to user modeling. In this article, we extend that study in the following two points. First, we investigate an alternative principle to realize the collaboration. In particular, we use a generalization of Bayesian predictive distribution [8, 2] in this study. Second, we apply our approach into a new context: hybrid of collaborative and content-based recommendation. We first formulate our problem and propose the general framework of our collaborative prediction in Sec. 2. Then, we describe our approach in a particular application to the recommendation case in Sec. 3. Experimental results using two real datasets, each of which is related to recommendation or user modeling, are presented in Sec. 4. Finally, some discussions are presented in Sec. 5.

2. Proposed Method

2.1 Problem formulation

Consider that there are multiple prediction tasks simultaneously, each of which contains the same kinds of input and target variables, each defined over a domain that is common across the tasks. That is, we have a set of U tasks, each of which aims to predict the value of a target $t_u \in \mathcal{T}$, given a new instance of an input vector $\boldsymbol{x}_u \in \mathcal{X}$, provided that a set of N_u observations $D_u = \{(t_u^n, \boldsymbol{x}_u^n) \mid n = 1, 2, \ldots, N_u\}$ has been given in advance. The subscript u indicates that these quantities are related to the u-th task. The superscript n is the index of samples. Note that neither the target domain $\mathcal{T}$ nor the input domain $\mathcal{X}$ has the task indices. In this study, we assume the target in each task is a single variable, but the extension to the cases of multiple targets is straightforward.

Given the U datasets, $D_1, D_2, \ldots, D_U$Ctask-specific predictors, i.e., usually regressors or classifiers, are first constructed, each of which is trained by using only the corresponding single dataset. We assume that not only the predicted values but also the distribution of the task's target variable in each task is available. The predictor for each task thus should be a probabilistic one, which may be a naive Bayes classifier, a logistic regressor, a Bayesian network, etc. By the traditional approach which do not consider the relationship of the tasks, the predicted value of a new target instance t_u^{new} given a new input $\boldsymbol{x}_u^{\text{new}}$ can be obtained from the trained task-specific model $p_u(t_u \mid \boldsymbol{x}_u)$. This can be done, for example, by taking its mode:

$$\hat{t}_u^{\text{new}} = \text{argmax}_{t_u}\, p_u(t_u \mid \boldsymbol{x}_u = \boldsymbol{x}_u^{\text{new}}). \tag{1}$$

This type of prediction is referred to as *individual prediction* in the rest of this article. On the other hand, the aim here is to present a principled way of incorporating the knowledge of other tasks into the prediction of a single task.

2.2 A key trick for collaborative prediction

Now we have U task-specific predictors which have already been trained:

$$p_1(t_1 \mid \boldsymbol{x}_1), \quad p_2(t_2 \mid \boldsymbol{x}_2), \quad \ldots, p_U(t_U \mid \boldsymbol{x}_U). \tag{2}$$

If nothing is done, the next prediction when given a new input for a certain task u becomes simply an individual prediction (1), since the variable $\boldsymbol{x}_u$ only appears in the u-th predictor. To overcome this, the key trick here is to replace the variables explicitly as follows:

$$p_1(t_s \mid \boldsymbol{x}_s), \quad p_2(t_s \mid \boldsymbol{x}_s), \quad \ldots, p_U(t_s \mid \boldsymbol{x}_s), \tag{3}$$

for the prediction of a specific task $s \in \{1, 2, \ldots, U\}$. Such a replacement is possible under the assumption of shared domains across the task variables. The idea underlying this replacement is that the task-specific predictors could have similar properties with each other.

With this trick, we then turn out to have an ensemble of U predictors. A naive way to utilize this ensemble to predict about a single task s is to take an average of probabilistic output of these predictor, given a new input $\boldsymbol{x}_s$. This is similar to the idea of Bagging. However, unlike the bootstrapped ensemble utilized in Bagging, the predictors in our context can be much heterogeneous. Thus, an appropriate weighting of each predictor is essentially required.

In our previous study [9], we proposed a Bayesian collaboration method, which was simply an application of the Bayesian predictive distribution with a special trick above. Using the notations: $T_s = \{t_s^n \mid n = 1, 2, \ldots, N_s\}$ and $X_s = \{\boldsymbol{x}_s^n \mid n = 1, 2, \ldots, N_s\}$, the posterior distribution $\pi_s(u \mid D_s)$ over the U models can be given as

$$\pi_s(u \mid D_s) \propto p_u(T_s \mid X_s)\pi_s(u) = \prod_{n=1}^{N_s} p_u(t_s^n \mid \boldsymbol{x}_s^n)\pi_s(u), \tag{4}$$

where the normalization constant is omitted. $\pi_s(u)$ is the prior belief about the occurrence or reliability on the U task-specific models, which is assumed simply as being proportional to the number of original datasets, $N_1, N_2, \ldots, N_U$, in this study. With this posterior (4), the Bayesian predictive distribution is given as

$$\bar{p}_s(t_s \mid \boldsymbol{x}_s) \equiv \sum_{u=1}^{U} \pi_s(u \mid D_s) p_u(t_s \mid \boldsymbol{x}_s). \tag{5}$$

The multiple predictors are thus naturally integrated under a task-specific weight $\pi_s(u \mid D_s)$ in the Bayesian framework. Intuitively speaking, the posterior weights of other tasks than s roughly reflect their similarity to the task s; the similarity will relatively decrease with increasing in the data number of the task s, since the likelihood $p(T_s \mid X_s)$ becomes large and confident.

2.3 α-Bayesian collaboration

The optimality of Bayesian predictive distribution have been stated in [11] in terms of the Kullback-Leibler (KL) loss for specified distributions. In our context, Eq. (5) can be seen as the optimal distribution that minimizes the following risk functional:

$$R\left[q\left(t_s\right)\right] = \sum_{u=1}^{U} \int dT_s \, \pi_s(u) p_u(T_s) \mathrm{KL}\left[p_u\left(t_s\right) \| \, q\left(t_s \mid T_s\right)\right], \tag{6}$$

where we have omitted the dependences on $\boldsymbol{x}_s$ in $p_u(t_s \mid \boldsymbol{x}_s)$ and on X_s both in $p_u(T_s \mid X_s)$ and $q(t_s \mid D_s) = q(t_s \mid T_s, X_s)$ for notational simplicity. $\mathrm{KL}\left[p(t) \| q(t)\right]$ is the KL diveregence defined as $\int dt \, p(t) \log \frac{p(t)}{q(t)}$. Note that the integrals should be replaced by summations in discrete-variable cases. The Bayesian collaboration thus can be seen as minimizing the expected KL divergence between the individual predictor (with replaced variables) and the resultant collaborative one; the expectation is with respect to the current belief about the random quantities, i.e., u and T_s, after the trick of variable replacements.

This previous study exemplifies that the property of the Bayesian collaboration depends on what kind of loss functional we employ. Since it is not clear the KL-loss is the best one for our applications, we investigate in this study a generalized Bayesian [8, 2] approach which utilizes a more general class of divergence, called the α-divergence, as the loss functional. The generalized Bayesian, or the α-*Bayesian* [2], predictive distribution is given by minimizing the following risk functional:

$$R_\alpha\left[q\left(t_s\right)\right] = \sum_{u=1}^{U} \int dT_s \, \pi_s(u) p_u(T_s) \mathcal{D}_\alpha\left[p_u\left(t_s\right) \| \, q\left(t_s \mid T_s\right)\right], \tag{7}$$

where the α-divergence $\mathcal{D}_\alpha\left[p(t)\|q(t)\right]$ is defined by

$$\mathcal{D}_\alpha\left[p\left(t\right)\|\,q\left(t\right)\right]=\begin{cases}\int dt\, p(t)\log\frac{p(t)}{q(t)} & (\alpha=-1)\\ \int dt\, q(t)\log\frac{q(t)}{p(t)} & (\alpha=1)\\ \frac{4}{1-\alpha^2}\left\{1-\int dt\, p(t)^{\frac{1-\alpha}{2}}q(t)^{\frac{1+\alpha}{2}}\right\} & (\alpha\neq\pm1)\end{cases}. \tag{8}$$

Note that $D_{-1}\left[p(t)\|q(t)\right]=\mathrm{KL}\left[p(t)\|q(t)\right]$ and $D_1\left[p(t)\|q(t)\right]=\mathrm{KL}\left[q(t)\|p(t)\right]$. By minimizing the risk (7) with respect to q, the predictive distribution is given as its optimum:

$$\bar{p}_s^{\alpha}(t_s\mid\boldsymbol{x}_s)\propto\begin{cases}\left\{\sum_{u=1}^{U}\pi_s(u\mid D_s)p_u(t_s\mid\boldsymbol{x}_s)^{\frac{1-\alpha}{2}}\right\}^{\frac{2}{1-\alpha}} & \alpha\neq1\\ \exp\left(\sum_{u=1}^{U}\pi_s(u\mid D_s)\log p_u(t_s\mid\boldsymbol{x}_s)\right) & \alpha=1\end{cases}. \tag{9}$$

The prediction can be done, for example, by taking its maximum:

$$\hat{t}_s^{\mathrm{new}}=\mathrm{argmax}_{t_s}\,\bar{p}_s^{\alpha}(t_s\mid\boldsymbol{x}_s=\boldsymbol{x}_s^{\mathrm{new}}). \tag{10}$$

We refer to this scheme as α-*Bayesian collaboration*. Note that the case of $\alpha=-1$ is equivalent to the previous Bayesian collaboration, so the α-Bayesian collaboration is its extension.

3. A Hybrid Recommendation using α-Bayesian Collaboration

In practical contexts of rating prediction, a primal target problem for recommendation systems, not only the rating information but also additional features typically related to item contents are often available. Recently, many types of *hybrid* methods of CF and content-based prediction have been investigated (cf. [1]), which utilize both types of information to predict unknown ratings. Such hybrid approaches are important, because they potentially improve the weaknesses of CFs or content-based ones particularly in dealing with new items or new users, respectively.

In this section, we investigate how to apply the proposed approach of collaborative prediction in such a context of hybrid recommendation.

3.1 Naive Bayes classifier

One simple but effective method for content-based rating prediction is to use the naive Bayes (NB) classifier [14], which "classifies" the input features (of item contents) into rating values, which are assumed to be discrete. Suppose now the recommendation system has collected a set of N_u explicit ratings from user u. The single task here is to predict a rating value t_u about an item previously unrated by the user, based on these N_u observed ratings and also content features of the item.

The NB method first estimates the joint distribution of the rating t_u and the content features $\boldsymbol{x}_u$ for each single task (user) u. The model is given as

$$p_u(t_u,\boldsymbol{x}_u)=p_u(t_u)\prod_{i=1}^{d}p_u(x_u(i)\mid t_u), \tag{11}$$

where $x_u(i)$ $(i=1,2,\ldots,d)$ is the i-th variable of $\boldsymbol{x}_u$, representing a single content feature. We assume both of the target and the input take discrete values for simplicity. Given the task-specific dataset $D_u=\{(t_u^n,\boldsymbol{x}_u^n)\mid n=1,2,\ldots,N_u\}$, where each sample corresponds to a single rating event, these probabilities can be estimated as

$$p_u(t_u)=\frac{\sum_{n=1}^{N_u}I(t_u^n=t_u)+\delta}{\sum_{t_u'}\left(\sum_{n=1}^{N_u}I(t_u^n=t_u')+\delta\right)}, \tag{12a}$$

$$p_u(x_u(i)\mid t_u)=\frac{\sum_{n=1}^{N_u}I(x_u^n(i)=x_u(i))I(t_u^n=t_u)+\delta}{\sum_{x_u(i)'}\left(\sum_{n=1}^{N_u}I(x_u^n(i)=x_u(i)')I(t_u^n=t_u)+\delta\right)}, \tag{12b}$$

In Eq. (12), $I(\cdot)$ is an indicator function which takes one or zero if the statement is true or false, respectively. The summations should be taken over the domains defined for each variable appropriately. δ is a small positive constant to avoid zero probabilities for unobserved values. Based on these estimated probabilities, the user u' s unknown ratings are estimated based on the posterior probability $p_u(t_u \mid \boldsymbol{x}_u)$ calculated straightforwardly from Eq. (11).

3.2 Item ID as an input feature

In the above setting of content-based prediction, a special kind of feature related to each item is ignored: the *item ID*. In the n-th rating event of user u, the system actually observes the rating t_u^n, the content features of an item $\boldsymbol{x}_u^n$, and also the item's ID y_u^n. According to the simplest setting of content-based methods, this absence causes almost no trouble since the objective is always to know the rating of unrated items: even if we incorporate y_u as an input feature such that

$$p_u(t_u, \boldsymbol{x}_u, y_u) = p_u(t_u) p_u(y_u \mid t_u) \prod_{i=1}^{d} p_u(x_u(i) \mid t_u), \tag{13}$$

the posterior distribution becomes $p_u(t_u \mid \boldsymbol{x}_u, y_u) \approx p_u(t_u \mid \boldsymbol{x}_u)$ if the item y_u has never been observed (rated), since in this case $p_u(y_u \mid t_u)$ is constant regardless of t_u.

In the collaborative setting, on the other hand, the posterior weights $\pi_s(u \mid D_s)$ will actually be affected by whether the item ID is included or not. In usual recommendation, the item ID is quite an important information, then we use the NB model in the form of Eq. (13) in this study. This is in fact a key setting of successful collaboration of the NB content-based predictors.

4. Experimental Results

4.1 Movie rating prediction

For the evaluation, we used the Movielens dataset (http://www.movielens.org). This dataset originally contains one million ratings of $6,040$ users over $3,952$ unique items (movies), where all the users provide more than 20 ratings. Each rating takes one of the values in $\{1, 2, 3, 4, 5\}$. We randomly selected $1,000$ users from this dataset, and used the sub-dataset throughout the experiment.

The Movielens dataset also contains the contents information of items. For simplicity, we used only the genre information, which consists of 18 binary labels each corresponding to a single genre, such as "Action" or "Comedy," where if the label is 1, the movie is of the genre. Thus, the number of input features is $d = 19$ (item's ID and the 18 labels).

The experimental setting was as follows. In each run of the experiment, the NB models (13) for every user were first trained individually. In each training of a user, one sample (an item-rating pair) was held out, and the learning was done based on the remained dataset. After all the trained models have been obtained, both the individual and the collaborative prediction were evaluated for each user according to the held-out sample. To reduce the computational time of the collaborative prediction, the posterior weights smaller than 10^{-8} were explicitly set at zero and hence the predictions by the corresponding models were not conducted. We repeated this procedure for 20 times, where the 20 held-out samples for each user were sampled without duplications.

Through the above procedure, we obtained the prediction results of 20 test samples for each user. Each user's prediction performance was then evaluated by the mean absolute error (MAE) criterion:

$$\mathrm{MAE}(u) = \frac{1}{m} \sum_{j=1}^{m} \left| \tau_u^j - \hat{t}_u^j \right|, \tag{14}$$

where τ and $\hat{t}$ are the true and predicted ratings, respectively, and $m = 20$.

The results with various α values are summarized in Fig 1. The upper two panels show the average MAEs by the collaborative prediction in the black solid curve, with the two dashed curves indicating the errorbars ($\pm$ standard deviations). For comparison, the gray straight lines are for individual predictions. The lower tow panels show the average of decreases in MAEs by the collaborative prediction, also with the errorbars (two dashed curves).

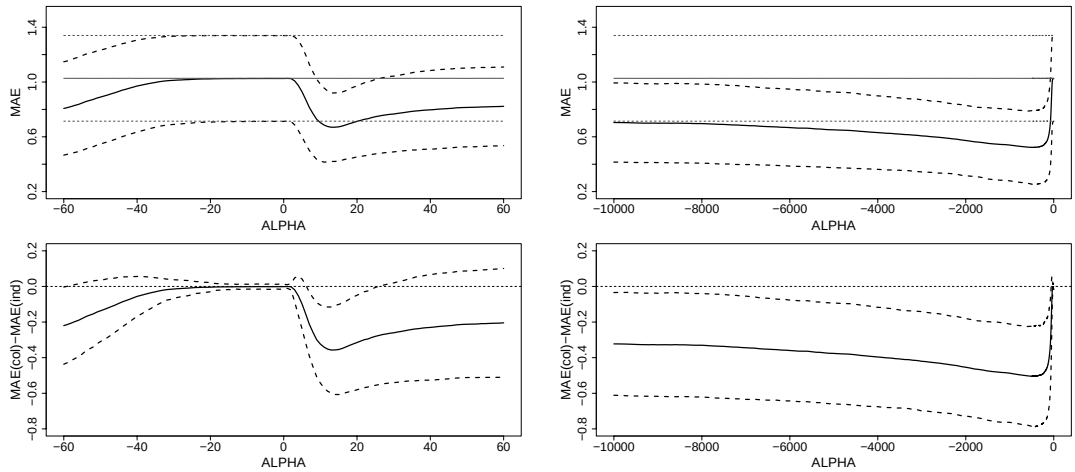

Figure 1: Comparison of MAE between individual and collaborative predictions. The horizontal axis denotes the value of α. Note that the left and right columns have different horizontal scales. Upper two panels: the average MAEs by collaborative prediction (black solid curves) and individual prediction (gray lines). Additional dashed curves represent $\pm$ standard deviations of the MAEs. Lower two panels: the average of MAE(collaborative) - MAE(individual). The two dashed curves are $\pm$ standard deviations. The horizontal dashed line just indicates zero.

The horizontal dashed line just indicates zero. The difference between the left and right panels are in the range of α as indicated by the horizontal scales. These panels show that the effect of collaboration highly depends on the α value. Interestingly, $\alpha = -1$ (normal Bayes) was not suitable for this dataset. The improvements by collaborative prediction were mainly observed in two parts: one is the very broad range as shown in the right two panels. Actually, the errors decreased by collaboration, in almost all the range in these panels. The other occurred in a relatively small region around $\alpha = 10$, as can be seen in the left two panels.

4.2 Printer usage prediction

Next, we evaluated our approach with a dataset related to a real problem of user modeling, which is referred to as a "printer usage" dataset. This is a set of electronic log data that were collected through daily uses of printer devices shared via network within a section of a company to which one of the authors belongs. The single task here is to predict each user's preferred setting of printing options based on input information. The data consisted of many records, each of which identified the user, context, and other features of a single printing job, including the values of printer settings such as the numbers of document pages or the usage of duplex/simplex.

The target and input in each task were constituted from several kinds of information summarized in Table 1. We refer to each one as an *attribute* in the following. In the pre-processing, the original values of `modelName` were replaced by anonymous ones. The values of `docPages` and `copies`, which originally took natural numbers, were quantized as shown in the table. The attributes `docExt`, `mediaSize` and `numberUp` originally took many instances, but we only used some major ones. In our setting, each of the upper four attributes in the table were considered as an input variable. That is, the input vector $\boldsymbol{x}$ took one of the configurations of (`modelName`, `docExt`, `docPages`, `copies`), for example, (Pr1, ppt, 2-5, 1). On the other hand, the combination of the lower four attributes was used as a single target variable; it took one of the values of `mediaSize`$\times$`duplex`$\times$`numberUp`$\times$`colorMode`, such as A4$\times$duplex$\times$1in1$\times$fullColor. The numbers of possible realizations were $|\boldsymbol{x}| = 625$ and $|t| = 80$. Samples containing missing values were simply removed. After the pre-processing, the total number of users was 77. The numbers of individual training data ranged from 5 to $2,110$.

As the learning model for each task, we again used the naive Bayes model. We first trained the naive Bayes models for all the 77 users, based on the originally available datasets. Based on this ensemble of 77 individual predictors (original ensemble), we evaluated the performance of our method for 27 users who have relatively large

Table 1: Eight attributes of printer logs used in this experiment. The upper four attributes were supposed to constitute the input, while the combination of lower four be the target.

Name	Description	Values
`modelName`	Anonymous forms of model names	Pr1, Pr2, Pr3, Pr4, Pr5
`docExt`	File extensions of original documents	doc, html, pdf, ppt, xls
`docPages`	Number of pages in original documents	1, 2-5, 6-20, 21-50, 51-over
`copies`	Numbers of copies	1, 2-5, 6-20, 21-50, 51-over
`mediaSize`	Media sizes	A3,A4,B4,B5,JapanesePostCard
`duplex`	Duplex or simplex printing	duplex, simplex
`numberUp`	Numbers of pages per sheet	1in1, 2in1, 4in1, 9in1
`colorMode`	Color modes	monochrome, fullColor

amounts of data, by artificially reducing the number of training data N_u for these users.

For a fixed $N_1 = N_2 = \cdots = N_{27} (= N)$, both N training and 200 test samples were randomly selected for each user from the user's original data collection. This was repeated for 50 times for each user. For the single pair of training and test dataset of a particular user s, the NB model was then trained and evaluated with the corresponding test data (individual prediction). In the collaborative prediction, the posterior weight distributions $\pi_s(u \mid D_s)$ ($u = 1, 2, \ldots, 77$) were first calculated, where for the models of the users other than the target user s, the models in the original ensemble were used. After this weight evaluation, the collaboration prediction was done and evaluated with the same test data as used in the evaluation of individual prediction.

Figure 2 shows the test accuracies by individual and collaborative predictions with $\alpha = -1, -100$, and 20. This figure collectively depicts all the results of the 27 users. The horizontal axis denotes the number of training data, $N (= 5, 20, 40, 60, 80,$ and $100)$, and the vertical the test accuracy. The accuracy was simply the fraction of the test cases in which the predicted value was equal to the true target one. The four kinds of lines show the mean accuracies by the individual and the collaborative prediction with three α settings, where the errorbars indicate $\pm$ standard deviations over the 27×50 runs.

In comparison to the accuracy of the individual prediction, the test accuracy by the collaborative prediction with a negative α value was improved especially with a relatively small N. In contrast, that with $\alpha = 20$ was degraded in the same conditions. The difference in the accuracy by the four methods are not very clear, in the cases of large N values.

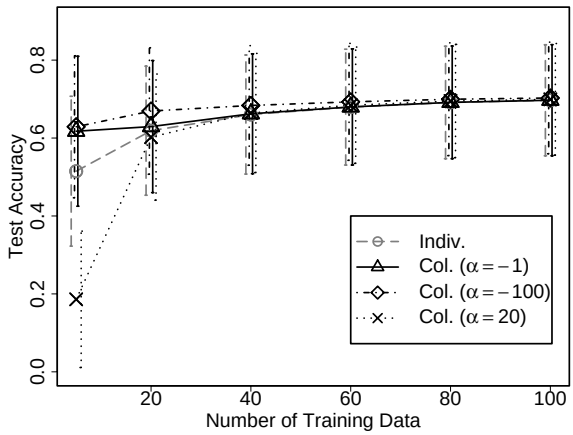

Figure 2: Test accuracies by individual and collaborative predictions with $\alpha = -1, -100$, and 20. All the results of the 27 users are collectively shown here. The horizontal axis denotes the number of training data, $N (= 5, 20, 40, 60, 80,$ and $100)$

5. Discussion

In this article, we proposed a new approach to collaborative prediction, by extending our previous study [9]. The proposed method is based on the theory of generalized Bayesian predictive distribution which employs the α-divergence as loss functional [8, 2]. The key idea of our collaborative prediction is to regard the predictors of other tasks (or users, in more specific contexts) as alternative models of a target task, by replacing the task-specific variables with the target task's ones. Our method is basically a straightforward application of the generalized Bayesian rule to this modified setting. The integration of multiple predictors for each target task was realized under a specific weight distribution computed from the corresponding dataset, and thus was tailored to be suitable for the particular task. Such a parallel use of the generalized Bayesian predictive distribution is a novel point of our current study.

In the experiment of the hybrid recommendation task, the effect of α to the prediction accuracy was interesting. The property of the α-loss in the integration of multiple predictors can be qualitatively stated as follows [2]. A smaller $\alpha(< 0)$ induces a more "optimistic" integration, in the sense that it typically assigns relatively large probability at a particular value, when at least one component predictor does so. On the other hand, a larger $\alpha(> 0)$ yields a more "pessimistic" one, because it will be strongly affected from small probability assignments by the individual predictors. Thus, the collaborative prediction with a small and a large α's would have aspects of *majority vote* or *consensus building* within a group of similar (positively weighted) users, respectively. The two successful ranges of α where the collaboration effectively improved the prediction accuracy may have reflected that both of the vote and consensus are useful for collaborative prediction; the different widths of the two ranges might suggest that making good consensus would be difficult than voting. More deep investigations about the usefulness of α-divergence in the context of recommendation remains as a topic for future studies.

References

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] S. Amari. Integration of stochastic models by minimizing α-divergence. *Neural Computation*, 19:2780–2796, 2007.

[3] B. J. Bakker and T. M. Heskes. Task clustering and gating for Bayesian multitask learning. *Journal of Machine Learning Research*, 3:261–270, 2003.

[4] J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

[5] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. 14th Conf. on Uncertainty in Artificial Intelligence*, pages 43–52, San Francisoco, CA, 1998. Morgan Kaufmann.

[6] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[7] R. Caruana. Multitask learning: A knowledge-based source of inductive bias. *Machine Learning*, 28:41–75, 1997.

[8] J. M. Corcuera and F. Giummolé. A generalized Bayes rule for prediction. *Scandinavian Journal of Statistics*, 26:265–279, 1999.

[9] J. Hirayama, M. Nakatomi, T. Takenouchi, and S. Ishii. Collaborative prediction by multiple Bayesian networks and its application to printer usage modeling. *Behaviometrika*, (submitted).

[10] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *Proc. 16th Int. Joint Conf. on Artificial Intelligence*, pages 688–693, 1999.

[11] F. Komaki. On asymptotic properties of predictive distributions. *Biometrika*, 83:299–313, 1996.

[12] B. Marlin. Modeling user rating profiles for collaborative filtering. In *Proceeding of the 17th Annual Conference on Neural Information Processing Systems*, 2004.

[13] B. Marlin and R. Zemel. The multiple multiplicative factor model for collaborative filtering. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[14] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI*, pages 187–192, 2002.

[15] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, 1994.

[16] G. I. Webb et al. Machine learning for user modeling. *User Modeling and User-Adapted Interaction*, 11(1–2):19–29, 2001.

[17] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007.

[18] Y. Zhang, S. Burer, and W. Nick Street. Ensemble pruning via semi-definite programming. *Journal of Machine Learning Research*, 7:1315–1338, 2006.

Jun-ichiro Hirayama received the B.E. degree from Kyoto University, Kyoto, Japan, in 2002 and the M.E. and Ph.D. degrees from Nara Institute of Science and Technology, Nara, Japan, in 2004 and 2007, respectively. Currently, he is a postdoctoral researcher at the Graduate School of Informatics, Kyoto University. His research interests are in machine learning and computational neuroscience.

Masashi Nakatomi received the B.E. and the M.E. degrees from the University of Tokyo in 2001 and 2003, respectively, and is currently a research engineer at Ricoh Company, Japan. His research interests include user modeling and knowledge-based systems.

Takashi Takenouchi is an assistant professor at Nara Institute of Science and Technology. He received his B.Eng. and M.Eng. degrees from the University of Tokyo, and Ph.D. degree from the Graduate University for Advanced Studies, Japan, in 1999, 2001 and 2004, respectively. His research interest has been on learning theory and discriminant analysis. He is currently working on machine learning and bioinformatics.

Shin Ishii received his B.E. in 1986, M.E. in 1988, and Ph.D. in 1997 from University of Tokyo. He is a professor of Graduate School of Informatics at Kyoto University. His current research interests are computational neuroscience, systems neurobiology and statistical learning theory.