# PURE: A PUBMED ARTICLE RECOMMENDATION SYSTEM BASED ON CONTENT-BASED FILTERING

TAKASHI YONEYA[1,2] HIROSHI MAMITSUKA[1]
t-yoneya@kirin.co.jp mami@kuicr.kyoto-u.ac.jp

[1] *Bioinformatics Center, Kyoto University, Gokasho Uji, 611-0011, Japan*
[2] *Discovery Research Laboratories, Kirin Pharma Co. Ltd., 3 Miyahara, Takasaki, Gunma 370-1295, Japan*

We have developed a PubMed article recommendation system, PURE, which is based on content-based filtering. PURE has a web interface by which users can add/delete their preferred articles. Once articles are registered, PURE then performs model-based clustering of the preferred articles and recommends the highly-rated articles by the prediction using the trained model. PURE updates the PubMed articles and reports the recommendation by email on daily-base. This system will be helpful for biologists to reduce the time required for gathering information from PubMed. PURE is downloadable under GPL license, via www.bic.kyoto-u.ac.jp/pathway/mami/out/PURE.tar.gz.

*Keywords*: recommendation; content-based filtering; PubMed; EM algorithm.

## 1. Introduction

MEDLINE/PubMed [16] is one of the largest public databases on biological and medical sciences [14, 16, 17] and updated daily with thousands of new papers. Biologists devote a considerable amount of time to checking PubMed to find papers relevant to their interests. To reduce their heavy burden, we develop a system, which we call PURE (standing for a PUbmed article REcommendation system), that automatically captures the preference of a user by using this user's response to the presented papers. Using the acquired preferences, PURE then reports relevant papers with scores to this user by email. This type of system is generally called a "recommendation system" [4]. Existing methods for this system can be classified into two types: collaborative filtering [8] and content-based filtering [7]. Collaborative filtering utilizes users with similar preferences. That is, as has been done in amazon.com, item X is recommended to a user who is buying item Y, if customers who bought item Y also bought item X. Thus, the performance of collaborative filtering depends on whether there exists a user with similar taste or not. A major drawback of collaborative filtering is that this filtering requires many users to find users having similar preferences. On the other hand, content-based filtering uses the content of items which are highly rated by a user and tries to find the preference of a user more directly. That is, this filtering can be completed by a single

user only, without using other users' information. The performance depends on the quality of the content of items, and if we can retrieve the item content sufficiently, we can expect that content-based filtering works well enough. PubMed provides a lot of contents of each article, divided into the title and the abstract, etc, preferably with a very small number of errors. Thus content-based filtering would be a better approach for recommending PubMed articles to a biologist than collaborating filtering, and we chose it in our system, PURE.

A user has to only register preferred articles to PURE. The preference patterns are then extracted from the registered articles by using model-based clustering, in which probabilistic parameters of a mixture model are estimated based on an EM (Expectation-Maximization) algorithm. PURE then downloads new articles from PubMed daily, which are ranked by using the likelihoods given by the trained probabilistic model to the newly downloaded articles. Finally PURE presents the highly rated articles to a user, with an interface at which a user can register preferred articles. A user can add any preferred articles at any step of the above procedure, which is iterated every time when new articles are added.

Some automatic services for recommending interesting articles are available currently, e.g. [2, 5, 9, 10, 13, 15], based on searching interesting articles by keywords. We emphasize that PURE is different from them, which present only the articles containing search keywords. PURE captures the preference pattern of the articles which are registered by a user and then recommends new articles which are the most relevant to the acquired patterns. By using this approach, PURE might find preferred papers which do not include keywords as well as those containing keywords. Another feature of PURE is that a user only has to input preferred articles by using a user-friendly and easy-to-use interface of PURE.

In experiments, we evaluated the performance of the learning method used in PURE in a two-class supervised learning manner. Experimental results indicated that PURE (or the method used in PURE) can provide a considerably favorable performance, say precision of 70% at 10% recall, even with only a very small number (e.g. 20) of training articles.

## 2. Implementation

Fig. 1 shows the whole scheme of the article recommendation by PURE, which has the following five steps that are iterated in this order: 1) Preferred articles are registered by a user out of their own findings or those presented by PURE. The selected articles are stored in a database of PURE. 2) A probabilistic model is trained using the stored articles to learn the preference of a user. 3) New PubMed articles are downloaded daily. 4) Prediction is done by ranking the downloaded articles using the trained model. 5) Finally the highly rated articles are presented to a user.

PURE is implemented in a client-server manner. That is, a user uses PURE at a client computer through a web interface, and the computations in the above five

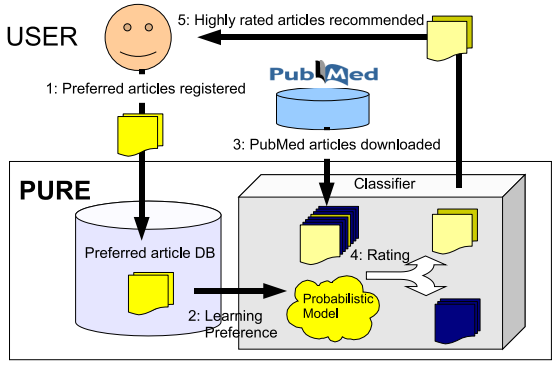



Fig. 1. Recommendation scheme of PURE.

steps are basically conducted by a central server computer. The reason why this system was adopted by PURE is the following two: 1) A large number, say normally thousands, of articles are accumulated in PubMed every day, and they must be downloaded daily by PURE. It is naturally inefficient if they are downloaded by each user, and it would be more computationally efficient that they are downloaded by one central computer only. 2) Learning from preferred documents and ranking downloaded documents are both relatively heavy loads. So it would be efficient and favorable to finish these tasks at a central computer at a convenient time, say at night, in a day.
We describe the detail of the above five steps below.

### 2.1. *Registration and storage of preferred articles to a PURE database*

We designed two different web interfaces for registering preferred articles, and a user can use both interfaces.
One is an interface by which a user can directly register articles (PubMed IDs). As an example of this interface, Fig. 2 shows a text box[a] at the top, in which PubMed IDs can be input by a user, and under this box, a list of articles which are already registered by this user. A user can write PubMed IDs in the box and click "submit" located at the bottom, and then a list of registered articles is shown under the input box as in Fig. 2. If a user wants to remove a registered article in the list, the checkbox in the left side of this article can be clicked and then "submit" can be clicked to delete the article.
The other is an interface by which a user can select articles out of those recommended by PURE. Fig. 3 shows an example of this interface, showing five recommended articles. A user can choose an article out of the recommended articles by

[a]For a first-time user, only this text box is displayed.

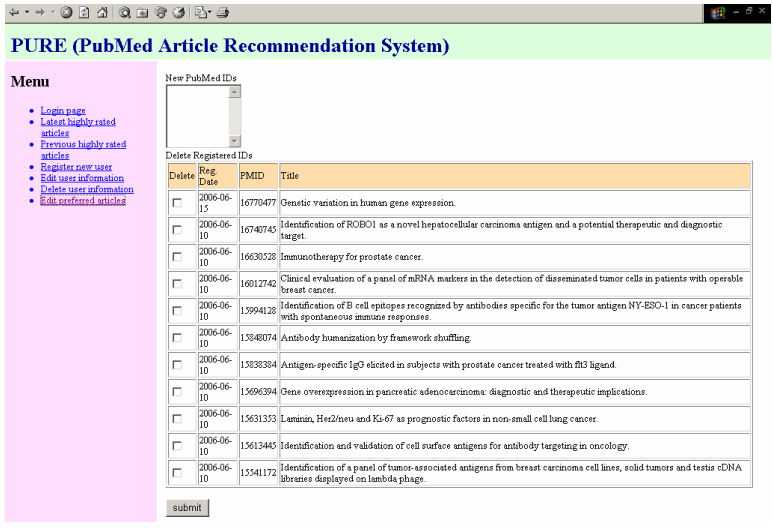



Fig. 2. A web interface for editing a list of preferred articles.

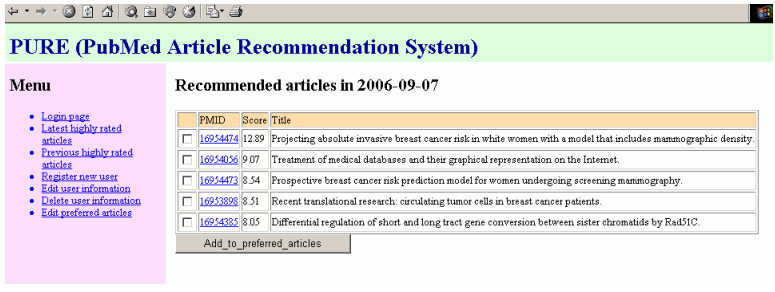



Fig. 3. A web interface for showing recommendations to a user which can be registered in this interface.

clicking the checkbox in the left side of the corresponding article, and click "submit" to register.

We note that in both interfaces, the input of PURE is PubMed ID only and no other items like keywords are required. The article which is newly registered by the first interface is downloaded from PubMed, while that by the second interface is not since it was already downloaded. The detail of the way to download articles from PubMed in PURE will be described in a later section.

### 2.2. *Training of a probabilistic model using stored articles to learn a user's preference*

This step is divided in the following two parts which are performed in this order.

#### 2.2.1. *Selecting Words and Assigning Initial Scores*

We treat a PubMed article as a set of words when it is rated or used for training a model. In doing so, we first generate a set of stop words to be removed from articles. The stop words can be classified into two types: those which are generated from PubMed articles and not.

Stop words in the first type are generated in the following manner, when PURE is installed on a computer. We first randomly download a large number (e.g. 10,000) of articles from PubMed and compute the *df* (document frequency) and the *tf-idf* (term frequency - inverse document frequency) [3] of each word. We note that for a word, the document frequency is defined as the number of documents having this word, and the term frequency is defined as the number of appearances of this word. Then a word with a high *df* or with a low *tf-idf* is a stop word.

Stop words in the second type are pre-defined in PURE, and they belong to at least one of the following three categories: 1) A word of less than three letters, e.g. I, IV. 2) A word of no alphabets, e.g. 10%, 10.3. 3) A word appearing in Journal of Business Research from Jan. 2005 to Apr. 2006, since they must be unrelated to biological and medical sciences.

Out of each downloaded article, we delete all these stop words to obtain a set of words for this article, and then for each word of the word set, the *tf* (term frequency) is computed to be assigned as the initial word score.

#### 2.2.2. *Learning a Probabilistic Model*

The selected words and initial scores in preferred articles are used to learn a probabilistic model, which is used to give a score to a new article. We use the preferred articles only for learning the preference, meaning that unsupervised learning, more precisely soft clustering of preferred documents (and words), is conducted. In this learning, we note that a user only has to show preferred articles. That is, the rating of each article is not required. This is the advantage of this method in terms of easy handling of preferred articles. We describe the detail of our probabilistic model and the learning process below.

We denote $d$ as an article, $z$ as a latent variable corresponding to a cluster, $s$ as a *field*, e.g. the title or the abstract, and $w$ as a word, and $n_{s,d}(w)$ as the count of word $w$ occurring in field $s$ of document $d$. We model the probabilistic structure of an article (a set of words) by using a finite mixture model [6] as follows:

$$p(d) = \sum_z p(d, z) = \sum_z p(z) \prod_{s \in d} \prod_{w \in s} p_s(w|z), \tag{1}$$

We then train probability parameters $p(z)$ and $p_s(w|z)$ from preferred articles. This is a so-called mixture model which is often used for clustering. An important feature which makes this model different from a normal mixture model is that we distinguish the fields, meaning that preferences in each field can be captured independently. In fact, we used two fields, the abstract and the title, independently. The probability

parameters are trained by an EM (Expectation-Maximization) algorithm [1], which repeats the following E-and M-steps alternately until some stopping condition is satisfied.

**E-step:**

$$p(z|d) = \frac{p(d,z)}{\sum_z p(d,z)}$$

**M-step:**

$$\hat{p}(z) = \frac{\sum_d p(z|d)}{\sum_{z,d} p(z|d)}$$

$$\hat{p}_s(w|z) = \frac{\sum_d n_{s,d}(w)p(z|d)}{\sum_w \sum_d n_{s,d}(w)p(z|d)}$$

This parameter estimation is carried out by a cron script, every time when a table of preferred articles is modified.

### 2.3. *Daily download of PubMed articles*

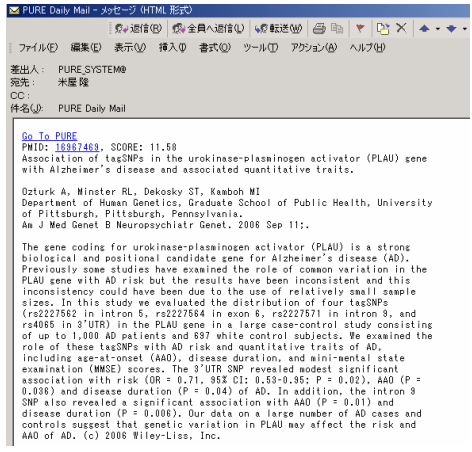
New articles of PubMed are retrieved every day in a MEDLINE format through the "Entrez Data (EDAT)" field by a cron-scheduled script, which is written by using an eFetch script [11] as a reference tool. Daily download of PURE is performed in the night time of Eastern Standard Time, i.e. the standard time at NCBI, because a large number of downloads must be done at night, as notified in [12]. Articles, downloaded from PubMed, are stored in a MySQL table with additional information like a user ID and the registered date, etc.

### 2.4. *Rating articles*

All downloaded articles are ranked by the trained probabilistic model. For each article, words for rating are extracted, and then the likelihood that an article $d$ is preferred can be computed by using Eq. (1). However, as is easily expected from Eq. (1), the larger the number of words in an article, the less the probability (or likelihood) of the article. To correct this bias, we use the $Z$-score for each article. That is, we gathered a large number of articles and grouped them into those having the same number of words. We then computed the mean $\mu$ and the standard deviation $\sigma$ of $p(d)$ in each group. Given a new article $d$, we counted the number of words of $d$ and computed the $Z$-score of $d$ using the $\mu$ and the $\sigma$ of the group containing articles with the same number of words as that of $d$, as follows: $Z = \frac{p(d)-\mu}{\sigma}$. Thus, downloaded papers are ranked by using their $Z$-scores.

### 2.5. *Presentation of highly rated articles*

A pre-defined number of highly rated articles can be recommended articles, which are stored in a MySQL table of PURE, and all other downloaded articles are discarded. The recommended articles are presented to each user by the following two


PURE Daily Mail - メッセージ (HTML 形式)

通信(R) | 全員へ返信(L) | 転送(W)

ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) アクション(A) ヘルプ(H)

差出人： PURE_SYSTEM@
宛先： 米屋 隆
CC：
件名(J)： PURE Daily Mail

Go To PURE
PMID: 16927489, SCORE: 11.58
Association of tagSNPs in the urokinase-plasminogen activator (PLAU) gene with Alzheimer's disease and associated quantitative traits.

Ozturk A, Minster RL, Dekosky ST, Kamboh MI
Department of Human Genetics, Graduate School of Public Health, University of Pittsburgh, Pittsburgh, Pennsylvania.
Am J Med Genet B Neuropsychiatr Genet. 2006 Sep 11;.

The gene coding for urokinase-plasminogen activator (PLAU) is a strong biological and positional candidate gene for Alzheimer's disease (AD). Previously some studies have examined the role of common variation in the PLAU gene with AD risk but the results have been inconsistent and this inconsistency could have been due to the use of relatively small sample sizes. In this study we evaluated the distribution of four tagSNPs (rs2227562 in intron 5, rs2227564 in exon 8, rs2227571 in intron 9, and rs4065 in 3'UTR) in the PLAU gene in a large case-control study consisting of up to 1,000 AD patients and 697 white control subjects. We examined the role of these tagSNPs with AD risk and quantitative traits of AD, including age-at-onset (AAO), disease duration, and mini-mental state examination (MMSE) scores. The 3'UTR SNP revealed modest significant association with risk (OR = 0.71, 95% CI: 0.53-0.95; P = 0.02), AAO (P = 0.036) and disease duration (P = 0.04) of AD. In addition, the intron 9 SNP also revealed a significant association with AAO (P = 0.01) and disease duration (P = 0.006). Our data on a large number of AD cases and controls suggest that genetic variation in PLAU may affect the risk and AAO of AD. (c) 2006 Wiley-Liss, Inc.

Fig. 4. A notification email of recommended articles.

ways. One is through a web interface, and Fig. 3 shows an example. As in this figure, the title of a recommended article is presented with its score ($Z$-score), and the whole abstract of an article can be displayed by clicking the corresponding PubMed ID[b]. If a user finds an article to be added to this user's preferred article list, this user only has to click the "check box" of this article and then click the "submit" button. The time period to keep the recommended articles in PURE can be set by a user, and expired articles are removed automatically. The other way of presenting recommended articles is using an email notification service, and Fig. 4 is an example of a notification email. A user can click "Go To PURE" at the top of this email to login to PURE and can see a web page showing a list of recommended articles, which is in Fig. 3.

## 3. Results and Discussion

### 3.1. *Recommendation of articles relevant to cancer diagnostics*

PURE is a web-based software with a user-friendly interface. A user can easily check the articles recommended already as well as those currently, and maintain preferred articles through the easy-to-use web-interface of PURE.

We show an actual result obtained by using PURE. We here assume that a user's interests are in diagnostics and target discovery for cancer therapy. Based on these interests, PubMed IDs of this user's preferred papers are registered to PURE. Fig. 2 shows a resultant interface at which eleven preferred papers are already shown. You can see that they are actually relevant with the above topic of this user's interests. The probabilistic model in PURE is trained by using the eleven registered articles to

[b]Previously recommended articles can be found just by clicking "Previously highly rated articles" in the menu as shown in the left column of Fig. 3.

capture the patterns of this user's interests. Articles, which are newly downloaded from PubMed, are then ranked by using the trained probabilistic model, and only highly ranked articles are presented to this user. Fig. 3 shows an actual interface by which five articles are recommended. In these five articles, the top article, which has a significantly high score, is an article on risk prediction of breast cancer, which we believe, is well suited to this user's original interests of cancer therapy.
You might think that this function of PURE is very similar to a function of PubMed which given an article, tries to find related articles. However, we emphasize that they are totally different from each other, because this function of PubMed is to find articles related with only one article, whereas the input of PURE is not only one article but multiple articles. PURE captures the pattern of interests from the input articles and presents articles relevant to this pattern. We note that any number of inputs can be allowed and the performance of PURE will be improved more by inputting a larger number of inputs.

### 3.2. *Evaluation of the performance of PURE with various training sets*

We then measured the performance of PURE in terms of how many examples we need to obtain satisfactory recommendations, using the probabilistic model in PURE. In this experiment, we used a two-class supervised learning manner to check the performance of PURE. The performance is measured by the precision at a low recall, exactly 10% recall, because only a small number of highly rated articles are recommended by PURE. Precision and recall are defined as $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$, respectively, where $TP$, $FP$ and $FN$ represent true positives, false positives and false negatives, respectively.

#### 3.2.1. *Data*

Each dataset in this experiment was obtained in the following manner. We first retrieved articles from PubMed by inputting a key word pattern. These articles are positive examples, which were divided into two groups: 500 for test (rating) and the remaining examples for training. Then for test, 4,500 negative examples were randomly collected so that the test set contains 10% (500) of positive articles and 90% (4,500) of negatives[c]. We repeated the division for positives ten times randomly and averaged the results over the ten runs.
We conducted the above experiment five times using five different keyword patterns. Table 1 shows the five keyword patterns we used and the number of articles retrieved from PubMed.

[c]The reason why we used this setting is that a biologist must be interested in probably only 10% or less articles out of all articles.

Table 1. Five datasets used in our experiments.

| Name | Keyword pattern | # articles |
|---|---|---|
| marker | cancer AND marker AND gene AND expression | 1593 |
| structure | protein AND compound AND structure | 1360 |
| rheumatoid | (autoimmune OR rheumatoid) AND drug | 5618 |
| cancer | cancer AND compound | 5167 |
| solubility | (compound OR small molecule OR drug) AND solubility | 3729 |

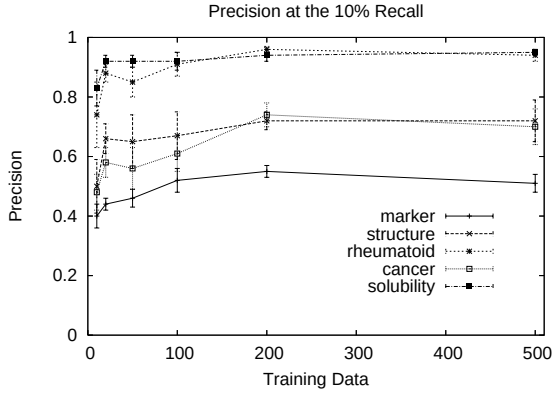



Fig. 5. Precisions at 10% recall with various training datasets

Table 2. Average precisions at 10% recall with changing the size of training articles.

| Training Articles | 10 | 20 | 50 | 100 | 200 | 500 |
|---|---|---|---|---|---|---|
| Average Precision (%) | 59.0 | 69.6 | 68.7 | 72.6 | 78.2 | 76.4 |

### 3.2.2. *Results*

Fig. 5 shows the precision at 10% recall for five datasets, which was obtained by increasing the number of training (positive) articles. Although the precision values at the same number of training articles were different depending on keyword patterns, we can see that the precisions were saturated at a relatively small number, e.g. twenty, of training articles. This result implies that nearly the best performance will be obtained simply by inputting a relatively small number of articles as preferred articles. Table 2 shows the average precision at 10% recall over the five datasets, when the number of training articles varied. This result indicates that our method can achieve an accuracy of roughly 70% when the top 10% articles are recommended, even if the number of training articles is only twenty. From these results, we can say that PURE must be useful for automatically finding articles which are relevant to the interests of a biologist.

## 4. Conclusions

We developed a PubMed article recommendation system, PURE, based on a content-based filtering. The results obtained by our various experiments imply that this system is useful in automatically finding articles which are relevant to a user's interest. A key feature of this system is an easy handling of preferred articles. That is, a user only has to input preferred articles into the system, which then captures the preference of this user from the inputs. Then by using the captured preferences, the system recommends the articles, which are, as shown in our experiment, highly ranked among the new articles downloaded from PubMed daily. Thus, this system will be helpful for finding preferred articles without using any other information such as keywords. PURE is downloadable under GPL license, via www.bic.kyoto-u.ac.jp/pathway/mami/out/PURE.tar.gz.

## 5. Acknowledgments

The authors would like to thank Ichigaku Takigawa, Raymond Wan, Shanfeng Zhu and Motoki Shiga of Kyoto University and Takayuki Onuma and Reina Nishida of Kirin Pharma for fruitful discussions and valuable comments.

## References

[1] Dempster, A., Laird, N., and Rubin, E., Maximum likelihood from incomplete data via the EM algorithm (with discussion), *J. Royal Stat. Soc. B.*, 39:1–38, 1997.
[2] Hokamp, K. and Wolfe, K.H., PubCrawler: keeping up comfortably with PubMed and GenBank, *Nucl. Acids Res.*, 32(Web Server issue):W16–W19, 2004.
[3] Joachims, T., A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, *ICML-97*, 143–151, 1997.
[4] Leavitt, N., Recommendation technology: Will it boost e-commerce?, *IEEE Computer*, 39:13–16, 2006.
[5] Marchin, M., Kelly, P.T., and Fang, J., Tracker: continuous HMMER and BLAST searching, *Bioinformatics*, 21:388–389, 2005.
[6] McLachlan, G. and Peel, D., Finite Mixture Models, Wiley, 2000.
[7] Mooney, R. and Roy, L., Content-based book recommending using learning for text categorization, *ACM DL-2000*, 195–204, 2000.
[8] Resnick, P. and Varian, H., Recommender systems, *Communications of the ACM*, 40:56–58, 1997.
[9] Shultz, M. and De Groote, S.L., MEDLINE SDI services: how do they compare?, *J. Med. Libr. Assoc.*, 91:460–467, 2003.
[10] `http://biomail.sourceforge.net/biomail/`
[11] `http://eutils.ncbi.nlm.nih.gov/entrez/query/static/efetchseq_help.html`
[12] `http://eutils.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html`
[13] `http://www.amedeo.com/`
[14] `http://www.info.scopus.com/`
[15] `http://www.leaddiscovery.co.uk/PubMed-dailyupdates.html`
[16] `http://www.ncbi.nlm.nih.gov/entrez/`
[17] `http://www.sciencedirect.com/`