A Content-Based Matrix Factorization Model for Recipe Recommendation

Chia-Jen Lin, Tsung-Ting Kuo, and Shou-De Lin

Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
heartherlin@gmail.com, {d97944007,sdlin}@csie.ntu.edu.tw

Abstract. This paper aims at bringing recommendation to the culinary domain in recipe recommendation. Recipe recommendation possesses certain unique characteristics unlike conventional item recommendation, as a recipe provides detailed heterogeneous information about ingredients and cooking procedure. Thus, we propose to treat recipes as an aggregation of features, which are extracted from ingredients, categories, preparation directions, and nutrition facts. We then propose a content-driven matrix factorization approach to model the latent dimension of recipes, users, and features. We also propose novel bias terms to incorporate time-dependent features. The recipe dataset is available at http://mslab.csie.ntu.edu.tw/~tim/recipe.zip

Keywords: Recipe recommendation, content-based recommendation, matrix factorization.

1 Introduction

With the prevalence of the Internet, people share huge amounts of recipes online, be a family recipe passed down through generations or one bright idea put into action in one afternoon. Currently there are over 10,000 cooking websites [1] providing various forms of information (e.g., texts, dish photos, cooking videos), as well as useful functions for searching and filtering by certain criteria. Conceivably, discovering appropriate recipes from such overwhelming database can be time-consuming. A recommendation system for recipes offers a desirable solution.

The task of recommending recipes does present several unique challenges. First, each recipe can be considered as a combination of several ingredients together with some contextual information such as cooking process and nutrition facts, or even certain meta-information such as its order in a course meal, type of cuisine, etc. As a result, a suitable recommendation system should take such profound and heterogeneous information into consideration. Second, there is no limit on the number of ingredients that can be used in a recipe, and generally recipes are not rated by as many viewers as movie or music does, we are facing a serious sparse rating and cold start problem. As shown in Table 1, the density of a recipe rating matrix is much lower than that of a movie rating. Such challenges can bring serious problems for traditional collaborative filtering models as these models rely heavily on the correlation among ratings to identify the latent connection between users and items.

V.S. Tseng et al. (Eds.): PAKDD 2014, Part II, LNAI 8444, pp. 560–571, 2014.

Table 1. Statistics of Netflix and FOOD.COM

Data	Netflix	FOOD.COM
User	480189	24741
Item	17770 (movies)	226025 (recipes)
Rating	100480507	956826
Sparsity	1.18%	0.02%
Average rating/per user	5654.50	4.23
Average rating/per item	209.25	38.67

Taking advantage of content information can be a solution to address the data sparsity and cold start problems. Unfortunately, such approach also has its own limitation in recipe recommendation since it fails to model the relationship among different features (e.g., different ingredients). An example is that the opinion of a user for an ingredient can be dramatically different depending on the type of dish to be prepared.

For instance, raw fish is a signature Japanese cuisine called Sashimi, but does not fit well with fries in traditional British fish and chips recipe. Therefore we cannot simply determine the usefulness of an ingredient without considering its correlation with other ingredients or preparation methods. This imposes a serious challenge for a content-based recommender.

In this paper, we propose a collaborative filtering approach called *content-driven temporal-regularized matrix factorization* (CTRMF), which aims at integrating heterogeneous content information into a Matrix Factorization (MF) model for a recipe recommendation system. The reason to choose an MF-based model is two-fold. First, MF-based models have been proven empirically as one of the most effective approaches for recommendation systems [2] [3]. Second, MF-based models allow us to exploit the latent correlation among objects, which is critical for recipes which include set of ingredients, preparation methods, and other meta-information. To incorporate the heterogeneous information of a recipe into an MF model, we propose to work on the *feature-matrix* instead of the original user-rating matrix. Feature matrix encodes the latent information about ingredients, categories, preparation directions, nutrition facts, and authors. We introduce several temporal biases into our model, including a novel idea to exploit the concept of Recency-Frequency-Monetary in different context.

1. We propose a content-driven MF-based model that incorporates the heterogeneous information of a recipe, including ingredients, dietary facts, preparation methods, serving order, cuisine type, and occasion. To our knowledge, this is the first proposal on using heterogeneous content information to perform recipe recommendation. Our experiments demonstrate decent improvement over the state-of-the-art models.

2. We propose a set of novel bias terms using the concept of Recency-Frequency-Monetary in different context. Such bias terms can potentially be applied to design recommendation systems in other domains.

3. Several works have been proposed on recipe recommendation. However, no benchmark test has been conducted to compare the performance of the proposed model with that of other competitors. This paper extracts real-life data from FOOD.COM to compare our model with two competitors to establish the performance benchmark on recipe recommendation.

2 Related Work

Personalized recommendation is important in consumer industry with huge variety of applications. Two common set of approaches are exploited for recommendation. (1) *Content-based filtering* is a paradigm that has been used mainly in the context of recommending items, for which informative content descriptors exist. Standard machine learning methods (e.g., SVM) have been used in this context. (2) *Collaborative filtering* exploits correlations between ratings across a population of users by finding users most similar to the active user and forming a weighted vote over these neighbors to predict unobserved ratings [11].

Recipe recommendation tasks have only been tackled by a small amount of researchers. Svensson et al. [4] propose a recipe recommendation system based on a user's explicit and implicit feedbacks through social interactions. Sobecki et al. [5] present a hybrid recommendation system, using fuzzy reasoning to recommend recipes. The above methods treat a recipe as a whole item, and require the social network between users for recommendation. In contrast, we break a recipe down into individual features, and need only the ratings but not social information to make recommendations.

There are also some recipe recommendation systems using content based techniques. Zhang et al. [6] construct a learning model using knowledge sources (e.g., WordNet) and a classifier (kNN) to make recommendations by finding similar recipes. Wang et al. [7] utilize NLP technique to parse preparation directions of recipes, and represent the recipes as cooking graphs consist of ingredients and cooking directions. They demonstrate that graph representations can be used to characterize Chinese dishes, by modeling the flow of cooking steps and the sequence of added ingredients. However, their work models the occurrence of ingredients and cooking methods but fails to take into account the relationships between ingredients. Neither do they consider users' preferences on specific recipes or ingredients. The main drawback of such language-dependent methods lies in the limited generality to non-Chinese recipes.

Freyne et al [8] proposes an Intelligent Food Planning (IFP) system, which breaks a recipe into core ingredients and gives each ingredient a weight. Then, IFP uses the weights of the ingredients to predict the rating of a new recipe. However, IFP does not take other information such as cooking style into account.

Forbes et al [9] propose content-boosted matrix factorization (CBMF), which is an extension of the matrix factorization model, to model hidden factors between users and ingredients.

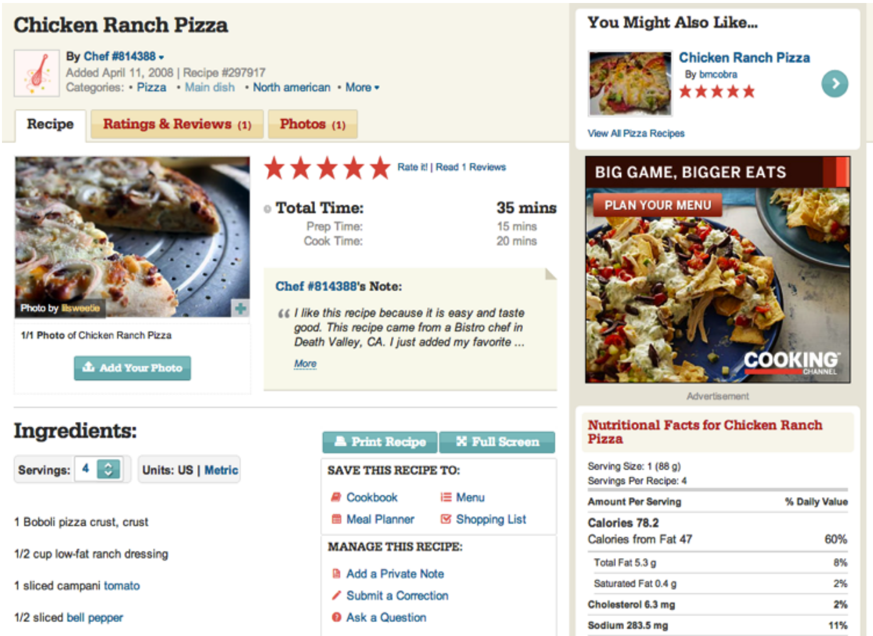

Fig. 1. A recipe from FOOD.COM

Although CBMF incorporates content information using linear constraints and proves the potential usefulness in the experiment, it only considers ingredients and does not use other information. Unlike our model, CBMF does not use temporal-regularized bias to improve accuracy. Experiment shows that our proposed model outperforms the CBMF model significantly. In this paper, we implemented IFP and CBMF as benchmark to compare with our proposed model.

3 Dataset and Features

3.1 Data Source

We collect data from 2000/2/25 to 2012/3/9 from FOOD.COM (www.food.com), one of the largest online recipe sharing communities. Figure 1 shows a sample recipe on FOOD.COM, which includes detail information such as ingredients, preparation directions, categories added by users, and the nutrition facts of this recipe. Our goal is to construct a recommendation engine that takes into consideration the profound types of information available.

We first filter out recipes that are rated no more than 3 times, as well as the users who rate no more than 5 times. Table 1 compares the statistics of FOOD.COM data with the Netflix data. We have found that this data is much sparser than the Netflix dataset. The average ratings per users/items are also much smaller. Such sparsity limits the effectiveness of a conventional collaborative filtering model and justifies the needs of adding content or meta-information into the recommendation system.

Table 2. Statistics of ingredient features in FOOD.COM data after data cleaning

Statistics	Value
Total ingredients counts in all recipes	2,131,207
Maximum ingredients number in a recipe	82
Minimum ingredients number in a recipe	1
Average ingredients number in a recipe	9
Maximum appearance of an ingredient on recipes	91,560
Minimum appearance of an ingredient on recipes	3
Average appearance of an ingredient on recipes	419

Table 3. Statistics of other features in FOOD.COM data

Statistics	Value
Positive features count in all recipes	3,087,494
Maximum number of features in a recipe	67
Minimum number of features in a recipe	3
Average number of features in a recipe	14
Maximum used times of a feature	220,775
Minimum used times of a feature	3
Average frequency for a feature being used	6,366

3.2 Features

We try to extract diverse features for each recipe. Originally, the dataset consists of 576,292 distinct ingredients, which requires certain level of data cleaning. We first correct some typos, and then merge ingredients of similar constituent, usually with different modifiers. For instance, “big red potato” and “small white potato” are both changed to “potato”. We then remove ingredients used no more than 3 times to obtain 5,365 binary ingredients features. Those features cover about 99.8% of all the ingredients used in the recipes. Table 2 shows the statistics of ingredient features.

Besides ingredients, we extract features from categories, preparation directions, and nutrition facts to create the profile of a recipe. We group these features into 6 groups:

- *Main Ingredient*: Ingredient with maximum weight in recipe, excluding water/stock/bouillon.
- *Dietary*: Based on the FDA reference daily intake (RDI) [100], healthy terms such as low-fat (i.e., Recipes only contains 2% of fat), high fiber (i.e., 20% or more for fiber) are defined as binary features.
- *Preparation*: Describe the preparation process of a recipe, such as ways of cooking (stir-fry, oven bake, etc.). Note that we only choose terms with sufficiently high TFIDF values as binary features.
- *Courses*: describe the order of the dish being served in a coursed meal. For instance, appetizers, main dish, or desserts.
- *Cuisines*: describe style of food in terms of countries, such as Italian, Asian, etc.
- *Occasion*: describe the situation of food being served (e.g., brunch, dinner party)

Table 4. Top 10 features in six groups

Main Ingredients	Courses	Preparation	Cuisines	Occasion	Dietary
Meat	Main dish	Time to make	North U.S.	Taste/mood	Low fat
Vegetables	Dessert	Easy	U.S.	Dinner party	Low sodium
Fruit	Side dishes	Equipment	European	Holiday/event	Healthy
Eggs/dairy	Lunch/snacks	< 60 minutes	Asian	Comfort food	Low carb
Pasta, rice & grains	Appetizers	Number of servings	Italian	Seasonal	Low cholesterol
Poultry	One dish meal	< 30 minutes	Southern U.S.	To go	Low calorie
Chicken	Salads	< 4 hours	Mexican	Weeknight	Vegetarian
Beef	Breads	< 15 minutes	Canadian	Brunch	Low protein
Cheese	Breakfast	3 steps or less	South west pacific	Potluck	Low sat. fat
Seafood	Cookies and brownies	5 ingredients or less	Southwestern U.S.	Summer	Kid friendly

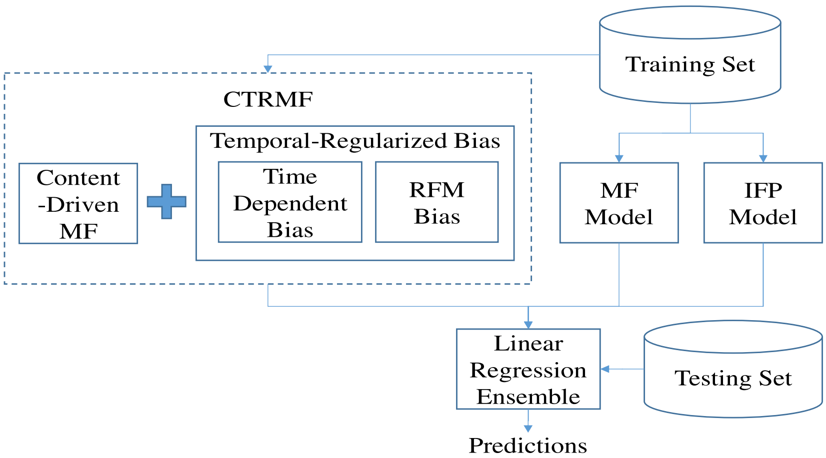

Fig. 2. Flowchart of our methodology

Here we obtain 485 additional features (not counting the original 5,635 ingredients) from FOOD.COM. Finally we merge highly similar features and remove extremely frequent, indiscriminative features such as salt and sugar. Finally we choose 5,538 features, 5,073 ingredients and 465 additional features. The statistics of those features are shown in Table 3. We list top 10 most frequent features in each group in Table 4.

4 Methodology

Figure 2 shows the flow chart of our proposed framework for recommendation. The heart of this system is the CTRMF engine, which will be described in section 4.1 and 4.2. As have been suggested by several researchers [2] [3] that the ensemble of models usually leads to the better results, we then linearly combine results from CTRMF with two diverse models, MF and IFP, to show that CTRMF can further improve the performance.

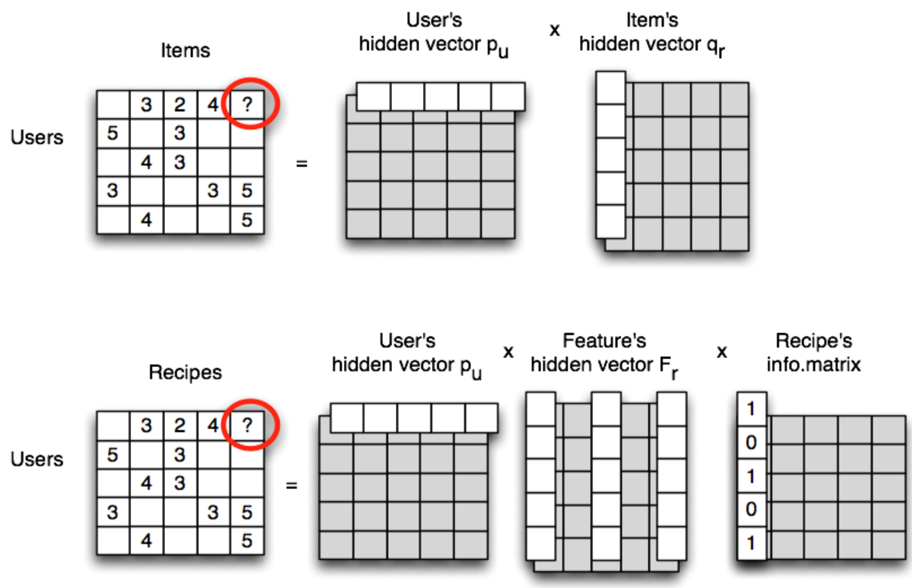

Fig. 3. Traditional MF and CTRMF

4.1 Content-Driven Matrix Factorization

We first define some notations:

S: training set	$r_{u,r}$: rating from user u to recipe r	μ: average rating in S
b_u: user bias	n_r: total number of recipes	n_u : number of users
b_r: recipe bias	n_f: total number of features	n_h: size of the hidden vector
P: a user-hidden matrix of dimension n_u * n_h, where each column represents a hidden vector of eash user		
F: a hidden-feature matrix of dimension n_h * n_f, where each row represents a hidden vector of each feature		
R: a recipe-feature matrix of dimension n_r * n_f, where each element is 1 if the recipe contains the corresponding feature, 0 otherwise		

Traditional MF tries to model hidden factors by decomposing the original user-item matrix into two low-dimensional matrices as below:

$$\hat{r}_{(u,r)} = p_u^T q_r$$

It models the interaction between latent user feature vector and item feature vector. That is, if a user likes a specific latent factor and an item has that factor, we conjecture that the user likes the item.

However, such model does not consider other useful information. Here we assume that recipes of common latent features are favored by certain group of users having similar latent features. Therefore our model predicts the rating using the following equation:

$$\hat{r}_{(u,r)} = p_u^T q_r^T R$$

Here p is a user-latent matrix, q represents latent-feature information, and R is the feature-recipe mapping. Note that p and q are learned from data and R is a matrix that encodes the heterogeneous information of each recipe. Figure 3 compares MF and CTRMF. Different from CBMF which does not include bias terms, here we add user bias and item bias; both are proven to be effective in our experiments. The objective function can then be defined as follows:

$$\min_{p^*,b^*,S} \sum_{(u,i)} (r_{u,r} - \mu - b_u - b_r - p_u^T F^T R_r)^2 + \lambda(\|p_u\|^2 + \|F\|^2 + b_u^2 - b_r^2)$$

The update function used in training can be derived as the follows:

$$e_{ur} \stackrel{def}{=} (r_{u,r} - \mu - b_u - b_r - p_u{}^T F^T R_r) \qquad p_u \leftarrow p_u + \eta(\sum_{r \in S_u} e_{ur} F^T R_r - \lambda p_u)$$

$$F \leftarrow F + \eta(\sum_{r \in S_u} e_{ur} R_r p_u{}^T - \lambda F) \qquad b_u \leftarrow b_u + \eta(e_{ur} - \lambda b_u)$$

$$b_r \leftarrow b_r + \eta(e_{ur} - \lambda b_r$$

4.2 Temporal-Regularized Bias

Bell et al. [2] have discovered from the Netflix data that there generally are some temporal patterns among ratings that can be exploited for better prediction accuracy. We also find similar patterns among the most active users in the FOOD.COM dataset. As shown in Figure 4, during the early days of the website, more than 30% of the ratings are relatively low (1, 2 and 3 in a five-star rating system). As the website becomes more mature, the percentage of low rates decreases to about 10%. Based on such observation, we add a time-aware bias to both users and recipes. We further propose to use the idea of Recency-Frequency-Monetary (RFM) Bias into our model. RFM is a concept proposed for analyzing customer behavior in customer relationship management (CRM). It is commonly used in database marketing and has received high attention in retail domain. The three main components of RFM are:

1. *Recency*: whether the customer purchased something recently?
2. *Frequency*: whether the customer purchased something frequently?
3. *Monetary*: whether the customer spends lots of money on something?

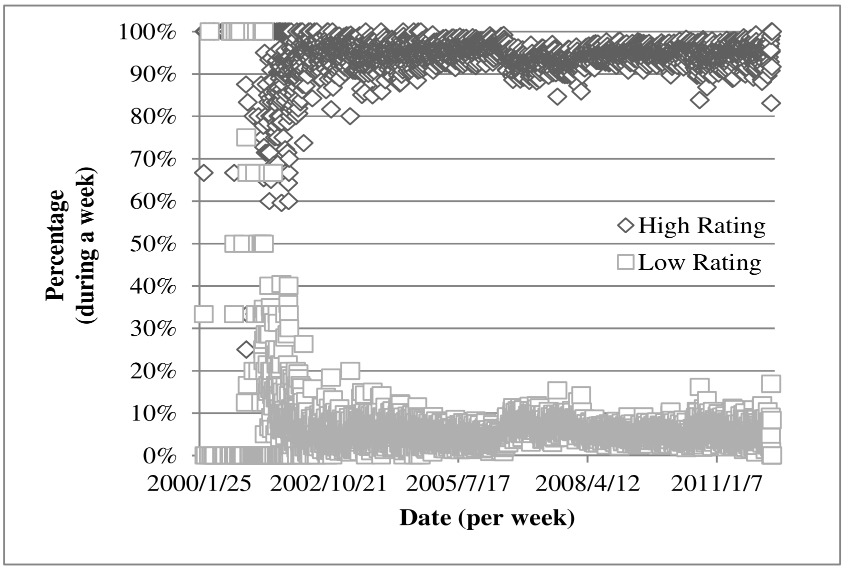

Fig. 4. Rating percentage distribution by week

We adopted the concept of RFM to incorporate more temporal biases into our model. For a certain user, *R*, *F*, and *M* become three binary variables indicating a user's Recent, Frequent, and Monetary rating behaviors. These three binary values then categorize the users into 8 different groups, and we assign each group a bias value to be learned. Similarly, items and authors are also divided into 8 groups, each correspond to a bias term. For each group, we try to learn a different bias value. Below we define the meaning of each group for users, recipes and authors.

4.2.1 User

First, from users' perspective, RFM of a user *u* can be defined as:

1. *Recency*: whether *u* rates a recipe more recently than *u*'s average rating recency in the past?
2. *Frequency*: whether *u* rates a recipe more frequently than *u*'s average rating frequency in the past?
3. *Monetary*: whether the most recent rating *u* provided rates higher than *u*'s average rating?

Figure 5 is an example showing that *u* had provided a rating of 3 on May 1st, 3 on May 8th, 4 on May 15, and 5 on May 19. In this example, the current Recency value is 21-19=2, lower than the average past Recency ((8–1) + (15–8) + (19–15)) / 3=6. Similarly, the current Frequency 5/21 (rated 5 times in 21 days) is higher than the average of user *u*'s past Frequency, 4/19 (rated 4 times in 19 days). For Monetary term, the last rating provided, a score of 5, is higher than user *u*'s past average rating, (3+3+4) / 3=3.3.

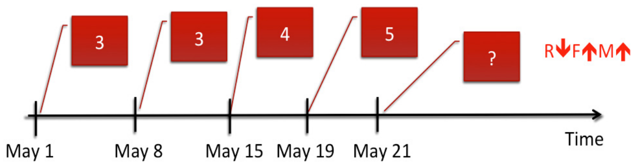

Fig. 5. Example of RFM in user side

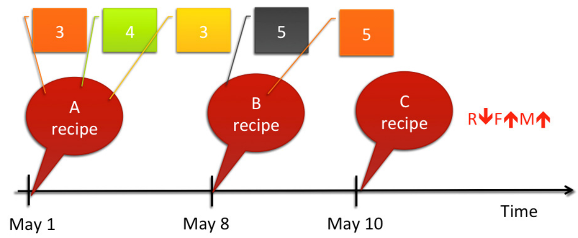

Fig. 6. Example of RFM in author side

Therefore, this user is assigned to group {R=0, F=1, M=1} and the corresponding bias terms is imposed. Such grouping allows us distinguish *hot users* from *cold users*.

4.2.2 Recipe

Similarly, from recipes' perspective, the RFM of a recipe *r* can be defined as:

1. *Recency*: whether *r* is rated more recently than its average recency of rating?
2. *Frequency*: whether *r* is rated more frequently than average frequency of rating?
3. *Monetary*: whether the most recent rating of *r* is higher than its average rating?

Similar to users, the recipes can now be divided into eight groups and each group is assigned a bias value to be learned. Such bias helps us distinguish *hot recipes* from *cold recipes*.

4.2.3 Author

From authors' perspective, RFM of an author *a* can be defined as:

1. *Recency*: does *a* create new recipe more recently than *a*'s average recency?
2. *Frequency*: does *a* create new recipe more frequently than *a*'s average frequency?
3. *Monetary*: does *a*'s last recipe received higher rating than *a*'s average rating received?

Note that the definition of Monetary here is slightly different from those of users and recipes. Figure 6 is an example showing that author *a* created the first recipe A on May 1st, second recipe B on May 8th, and recipe C on May 10th.

Table 5. RMSE results of baseline, our method, and ensemble

Method	RMSE	Method	RMSE
IFP	0.6186	CTRMF (without RFM)	0.5931
MF	0.6015	CTRMF (with RFM)	0.5901
CBMF	0.6233	Linearly Regression Ensemble	**0.5813**
Content-Driven MF	0.6013		

In this example, the current recency is 2, lower than the average past recency, 7 / 1=7. Similarly, the current frequency 3/10 is higher than the average frequency of author *a*, 2/8. The last recipe created received an average rating of 5 which is higher than the average ratings received by recipes posted by author *a*, (3+3+4) / 3=3.3. Each author is assigned to one of the eight groups with its associated bias term. This bias helps us distinguish *hot authors* and *cold authors*.

Combining the three perspectives identified from FOOD.COM dataset, our final objective function is defined as follows:

$$\min_{p_u, b_u, F} \sum_{(u,i)\in k} (r_{u,r} - \mu - bu_{Time(t)} - br_{Time(t)} - b_{rfm(u,r,a,t)} - p_u^T F^T R_r)^2$$
$$+\lambda(\|p_u\|^2 + \|F\|^2 + (bu_{Time(t)})^2 + (br_{Time(t)})^2 + b^2_{rfm(u,r,a,t)})$$

Note that b_{rfm} term is the multiplication of three terms, user, recipe, and author biases, defined above. And, the update functions are as follows:

$$e_{ur} \stackrel{def}{=} (r_{u,r} - \mu - bu_{Time(t)} - br_{Time(t)} - b_{rfm(u,r,a,t)} - p_u^T F^T R) \qquad p_u \leftarrow p_u + \eta(\sum_{r\in S_u} e_{ur} F^T R_r - \lambda p_u)$$

$$F \leftarrow F + \eta(\sum_{r\in S_u} e_{ur} R_r p_u{}^T - \lambda F) \qquad bu_{Time(t)} \leftarrow bu_{Time(t)} + \eta(e_{ur} - \lambda bu_{Time(t)})$$

$$br_{Time(t)} \leftarrow br_{Time(t)} + \eta(e_{ur} - \lambda br_{Time(t)}) \qquad b_{rfm(u,r,a,t)} \leftarrow b_{rfm(u,r,a,t)} + \eta(e_{ur} - \lambda b_{rfm(u,r,a,t)})$$

Here we train our model using stochastic gradient decent (SGD). We set λ to 0.01, η to 0.001, and the number of hidden factors to 100.

5 Experiments

We randomly select 4/5 of data from the users' ratings as training data, and use the rest as testing data. We compare our model (CTRMF) with IFP, standard MF, and CBMF models. The results showing in Table 5 reveal that the content-driven MF (introduced in Section 4.1) is better than CBMF, proving that the bias terms are useful. CTRMF has significant improvement over the existing methods with better RMSE. Also, adding RFM bias terms can improve CTRMF. Then we use linear regression to create an ensemble of IFP, MF, and our method. We divide the training data into training and validation to learn the parameters (i.e., the testing data remains unseen during ensemble). The ensemble RMSE can be further boosted to 0.5813.

6 Conclusion

This paper, to our knowledge, is the first ever attempt that incorporates 6 different types of content information, main ingredient, dietary, preparation, course order, cuisine type, and occasion, with user ratings for recipe recommendation. Such data will be released and become the only benchmark data so far for recipe recommendation. We also proposed the CTRMF model which is the first recommendation model that adopts the concept of RFM-based bias for recommendation, which can be potentially applied to domains other than recipe recommendation. Finally, this paper is the first to provide empirical comparison on different state-of-the-art models. For the future, we intent to extend the recommendation into a set of courses, such as appetizer, main dish, soup, dessert, and so on.

References

1. Alexa, http://www.alexa.com/topsites/category/Top/Home
2. Bell, R.M., Koren, Y., Volinsky, C.: The BellKor solution to the Netflix Prize. Technical Report, AT&T Labs Research (2007)
3. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer (2009)
4. Svensson, M., Laaksolahti, J., Höök, K., Waern, A.: A recipe based on-line food store. In: IUI 2000: Proceedings of the 5th International Conference on Intelligent User Interfaces, pp. 260–263 (2000)
5. Sobecki, J., Babiak, E., Słanina, M.: Application of hybrid recommendation in web-based cooking assistant. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006, Part III. LNCS (LNAI), vol. 4253, pp. 797–804. Springer, Heidelberg (2006)
6. Zhang, Q., Hu, R., Namee, B., Delany, S.: Back to the future: Knowledge light case base cookery. Technical report, Technical report, Dublin Institute of Technology (2008)
7. Wang, L., Li, Q., Li, N., Dong, G., Yang, Y.: Substructure similarity measurement in Chinese recipes. In: Proceeding of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, pp. 979–988. ACM, New York (2008)
8. Freyne, J., Berkovsky, S.: Intelligent food planning: personalized recipe recommendation. In: Proceeding of the 14th International Conference on Intelligent User Interfaces, IUI 2010, pp. 321–324. ACM, New York (2010)
9. Peter, F., Zhu, M.: Content-boosted matrix factorization for recommender systems: experiments with recipe recommendation. In: Proceedings of the Fifth ACM Conference on Recommender Systems. ACM (2011)
10. http://en.wikipedia.org/wiki/Reference_Daily_Intake, http://en.wikipedia.org/wiki/Dietary_Reference_Intake, http://www.fda.gov/downloads/Food/GuidanceRegulation/UCM265446.pdf
11. Basilico, J., Hofmann, T.: Unifying collaborative and content-based filtering. In: ICML (2004)