Robust Content-Based Recommendation Distribution System with Gaussian Mixture Model*

Dat Nguyen Van[1], Van Toan Pham[1], and Ta Minh Thanh[1,2]

[1] *Research and Development Dept, Sun Asterisk*, Ha Noi, Viet Nam
{nguyen.van.dat,pham.van.toan}@sun-asterisk.com
[2] *Le Quy Don Technical University*, Ha Noi, Viet Nam
thanhtm@mta.edu.vn

Abstract. Recommendation systems play an very important role in boosting purchasing consumption for many manufacturers by helping consumers find the most appropriate items. Furthermore, there is quite a range of recommendation algorithms that can be efficient; however, a content-based algorithm is always the most popular, powerful, and productive method taken at the begin time of any project. In the negative aspect, somehow content-based algorithm results accuracy is still a concern that correlates to probabilistic similarity. In addition, the similarity calculation method is another crucial that affect the accuracy of content-based recommendation in probabilistic problems. Face with these problems, we propose a new content-based recommendation based on the Gaussian mixture model to improve the accuracy with more sensitive results for probabilistic recommendation problems. Our proposed method experimented in a liquor dataset including six main flavor taste, liquor main taste tags, and some other criteria. The method clusters n liquor records relied on n vectors of six dimensions into k group ($k < n$) before applying a formula to sort the results. Compared our proposed algorithm with two other popular models on the above dataset, the accuracy of the experimental results not only outweighs the comparison to those of two other models but also attain a very speedy response time in real-life applications.

Keywords: Recommendation · Content-based · Gaussian-mixture-model (GMM) · Distribution-recommendation.

1 Introduction

Due to the proliferation of internet, it has brought tremendous chance for people's lives. On the other hand, the myriad and abundance of information on the web has determined a rapidly increasing difficulty in finding what we actually need in a way that can fit the best our requirements[7, 11, 2]. Recommendation systems can be effective way to solve such problems without requiring users provide explicit requirements[32, 34]. Instead, the system can analysis the content data of item properties, which actively recommend information on users that can satisfy their needs and interests[18, 17]. The general content-based architecture is shown in Fig. 1.

* Supported by Sun Asterisk Inc.

Content-based filtering algorithm is widely used because of its simplicity and effectiveness at the begin time of any recommendation systems. According to Pasquale $et.\ al.$ [15], there are many benefits reaped from content-based recommendation (CB) systems compared to the other Collaborative Filtering (CF) one such as user independence, transparency, cold-start problems, and so on. Beside, there are still some shortcoming existing as limited content for analyzing, over-specialization or lack of rating data of new users and adequate accuracy for some specific problems. Hangyu $et.\ al.$ [30] used GMM for CF recommendation algorithm to solve the sparse users rating data. Chen $et.\ al.$ [4] proposed a hybrid model, which combines GMM with item-based CF recommendation algorithm and predicted the ratings on items from users to improve the recommendation accuracy. Rui Chen $et.\ al.$ [3] took GMM with enhanced matrix factorization to reduce the negative effect of sparse and high dimension data. In the context of music recommender systems, Yoshii $et.\ al.$ [31] proposed a hybrid recommender system that combines collaborative filtering using user ratings and content-based features modeled via GMM over MFCCs by utilizing a Bayesian network. However, CF or hybrid systems require behaviour history of users that the reason for the need of CB. Furthermore, CB based on distribution of item features have not been solved yet. A telling of example is using CB for automatically find similar items based on distribution and distance of its features in Fig. 2. These kind of probabilistic problems in recommendation systems is quite different which cannot be solved by usual common methods. Furthermore, the description of content data of items features is sometimes unreliable, inadequate that detrimentally affect to the accuracy of CB systems [12]. Due to two problems mentioned above, we propose a new approach for solving these problems by using GMM [26] to cluster all items into different groups before applying a gaussian filter function (GFF) as a calculation similarity method for sorting results. To demonstrate our effective model, we experiment and compare to two other popular methods, Bag of Word [1] with GFF (BOW + GFF), and GMM with euclidean distance (ED) [14] (GMM + ED). Our propose model not only outperforms the accuracy of the two others, but also get better in prediction time response.

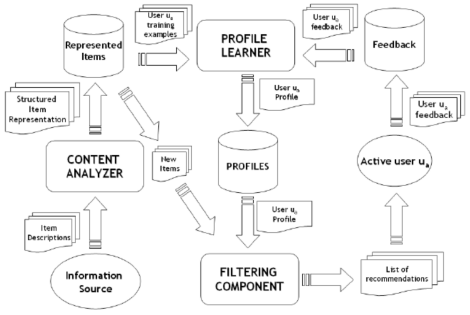

Fig. 1. High level architecture of a Content-based recommendation system.

The paper is organized as follows. Related work is introduced in Section 2 while dataset in Section 3. In Section 4, the architecture and details of proposed model is given. Experiments and evaluations are shown in Section 5. The conclusion are discussed in Section 6.

2 Related Work

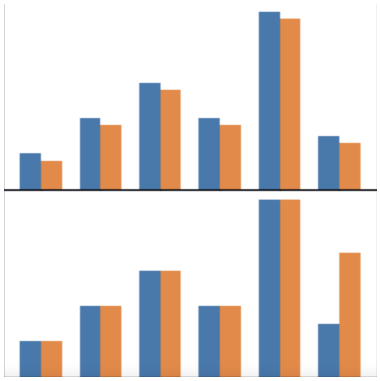

Fig. 2. An example between using distribution and distance calculation for distribution recommendation.

We introduce some preliminary knowledge that needs to be used. The following is the detailed information of them.

2.1 Content-based Recommendation

`Content-Based Recommendation Systems` is one of the most common method in building recommendation systems. The algorithm is born from the idea of using the content descriptions of each item for recommending purposes. It can be divided into two approaches: Analysing the description of item properties only, and building user profile for individuals based on feature's content of items and personal rating data [34, 15].

2.2 Popular similarities

In the `Content-based` algorithm, the similarity calculation method directly affects the accuracy of results. Some similarity calculation methods have been widely used which are listed below:

`euclidean distance`: One of the most popular methods to measure the similarity between two vectors by calculating the sum of square distance of each element respectively in those vectors. Read [14] for more information.

`Cosin`: The main idea is to measure two vectors by calculating the cosine of angle between the two vectors [22].

`Pearson`: The pearson correlation coefficient reflects the degree of linear correlation between two vectors [27],

`Jaccard`: The Jaccard Similarity is often used to compare similarity and different between two finite sample set [20],

2.3 Gaussian Mixture Model (GMM)

Gaussian Mixture Model is a function that is comprised of several Gaussians. GMM can fit any types of distribution, which is usually used to solve the case where the data in the same set contains multiple different distributions [25, 5], each identified by $k \in \{1..K\}$ where K is the number of clusters of our dataset.

GMM is defined as:

$$p(x) = \sum_{i=1}^{k} \alpha_i . N(x|\mu_i, \Sigma_i), \tag{1}$$

where $N(x|\mu_i, \Sigma_i)$ is the i^{th} component of the hybrid model, which is a probability density function of the n dimensional random vector x obeying Gaussian distribution. It can be defined as below:

$$N(x) = \frac{1}{(2\pi)^{\frac{n}{2}} \mid \Sigma \mid^{\frac{1}{2}}} \epsilon^{-\frac{1}{2}(x-\mu)^T \sum^{-1}(x-\mu)} \tag{2}$$

and

$$\sum_{i=1}^{k} \alpha_i = 1 \tag{3}$$

We assume that a sample set $D = \{x_1, x_2, x_3, ..., x_m\}$ is given that obey gaussian distribution mixture distribution. We use the random variable $z_j \in \{1, 2, ..., k\}$ to represent the mixed component of the generated sample x_j, whose value is unknown. It can be seen that the prior probability $P(z_j = i)$ of z_j corresponds to $\alpha_i (i = 1, 2, 3, ..., k)$. According to Bayes' theorem [13], we can get the posterior probability of z_j as follows:

$$\begin{aligned} p(z_j - i|x_j) &= \frac{P(z_j = i).p(x_j|z_j = i)}{p(x_j)} \\ &= \frac{\alpha_i.N(x_j|\mu_i, \Sigma_i)}{\sum_{l=1}^{k} \alpha_l.N(x_j|\mu_l, \Sigma_l)} \end{aligned} \tag{4}$$

In the above formula, $p(z_j = i|x_j)$ represents the posterior probability of sample x_j generated by the i^{th} Gaussian mixture. Assuming $\gamma_{ij} = \{1, 2, 3, ..., k\}$ represents $p(z_j = i|x_j)$. When the model parameters $\{(\alpha_i, \mu_i, \Sigma_i)|1 \leq i \leq k\}$ in the equation (4) are known, the GMM clusters divide the sample set D into k clusters $C = \{C_1, C_2, ..., C_k\}$ [25]. The cluster label λ_j of each sample x_j can be determined according to equation below:

$$\lambda_j = \arg max_{i \in 1,2,3,...,k} \gamma_{ji}$$

We get the cluster label λ_j to which x_j belongs and divide x_j into cluster C_{λ_j}. The model parameters $\{(\alpha_i, \mu_i, \Sigma_i)|1 \leq i \leq k\}$ is solved by appling EM algorithm [16].

3 Dataset

Our proposed model is implemented on a dataset about liquor, more specifically, about sake which is one of the most prevalent kind of liquor in Japan. The dataset was collected from Sakenowa[3] being one of the most well-known and reputed website selling the sake[4]. The dataset totally contains 1072 records characterized by 19 properties such as liquor name, liquor brand, year of manufacture, liquor images, liquor flavour tags, liquor six axis flavour taste ($f_1, f_2, ..., f_6$ stands for fruity, mellow, rich, mild, dry and light(Fig. 3). Noticeably, liquor six axis flavour taste and liquor flavour taste would play more important role than the others. The range value of six flavour taste ($f_1 - f_6$) axis is in [0, 1], meanwhile the dominant parts belong to [0.2, 0.6]. The text fields in

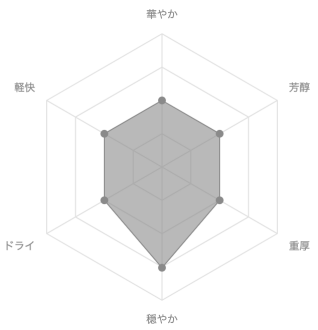

Fig. 3. A visualization about 6-axis flavour taste

[3] https://sakenowa.com
[4] https://en.wikipedia.org/wiki/Sake

the dataset all is written behind Japanese form.
However, this is a real challenging dataset due to
lack of many fields that lead to sparse in data, especially in six main fields $f_1, ..., f_6$. Therefore, our task of recommendation become more difficult and be negatively affect the recommendation results. More specifically, a disappearance or null value of 6-axis fields is greater than 30%, a nearly $2\ \%$ of null value flavour tags. Further more, many tags value is unreliable, untrust and incorrect that need to be clean and pre-processing (Tab. 1).

4 Proposed model

We introduce and explain our proposed model more in detail. As it was mentioned in previous part, we have to return the most similar products based on 19 metadata fields. In particular, 6-axis flavour taste and flavour tags are the main factors mostly affecting to the results both in the sensibility and accuracy side. Therefore, we just select 6-axis flavour taste and flavour tags for better results. The more similar in 6-axis flavour taste, the better results will be.

Table 1. Dataset blank fields statistic

$\mathbf{f}_{1..6}$	**Flavour tags**	**Product name(en)**
Float	***String***	***String***
30.4 %	1.77 %	13.4 %

More detail, we initially use Gaussian mixture model to cluster all items into $K = \{1, 2, ..., k\}$ group, then sorting results in each group with each item. Whenever finding top similar items of a item, we just jump up to the group the item belongs to and sort the group's items to return top m similar items. To sort the results, it is also possible to use some popular similarity calculation such as `cosine` or `euclidean distance`, but for better accuracy, we use a equation that calculate the distribution weight between two vectors obeying Gaussian distribution (normal distribution). The results illustrate that the more similar in 6-axis amongst items the bigger weights will be germinated. The flow of our proposed model is shown in Fig. 4.

4.1 Data pre-processing

It is a fact that text mining is very important in every text-related problems, and CB is not an exception. Previously mentioned, we only choose flavour tags and 6-axis flavour taste as the features for compute the similar between items. The flavour tags are the set of text document written behind Japanese form which require to be cleaned. We convert 6-axis into float and need to do some pre-process techniques for such flavour tags text fields like tokenization, stemmings, stop word removal, find and replace synonyms, lemmatization, and so on [23, 6, 24] before utilizing it. Moreover, the flavour tags field has been splitted into different semantic words, so we disregard the tokenization step and move forward with the other steps.

4.2 Clustering

As we recognize that the final recommendation items depend too much on 6-axis flavour taste and flavour tags. In the common and traditional way, there is a way to build a vector representing for all properties of each item, then utilizing a similarity calculation method like cosine or euclidean to sort and return top m results. However, in some case, the flavour tags are not enough adequate and precise that adversely affect to the final recommendation. Moreover, there is always an unseen problem of using cosine or euclidean that a compensate between each element of 6-axis flavour taste($f_1 - f_6$) leads to unequal among those elements($f_1 - f_6$) of results. Therefore, we decide to group all items based on it's distribution 6-axis flavour taste into different clusters to ensure items which have the same distribution will be in the same cluster that is the foundation for sorting afterwards (Fig. 5).

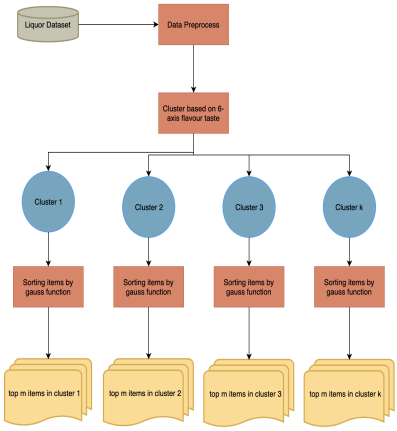

Fig. 4. The model Activities Diagram

4.3 Gaussian function for sorting

As we have $K = \{1, 2, ..., k\}$ clusters, we assume a query item is the center of the cluster we want to find. Our destination is figure out top m items that have the same distribution as much as possible, so Gaussian filter function (GFF) is the better choice than cosine or euclidean. The Gaussian function equation is defined as follows:

$$G_{kl}(f_{il}, f_{jl}) = \exp - \frac{(f_{il} - f_{jl})^2}{2\sigma_{kl}^2} \quad (5)$$

where $G_k(f_{il}, f_{jl})$ is considered as a weight between each pair of element l^{th} in 6-axis flavour taste of two different items (i, j) in cluster $k, l = \{1, 2, ..., 6\}$, and σ_{kl} is the standard deviation of the l^{th} element in 6-axis flavour taste in group k. Equation for σ_{kl} is defined as below:

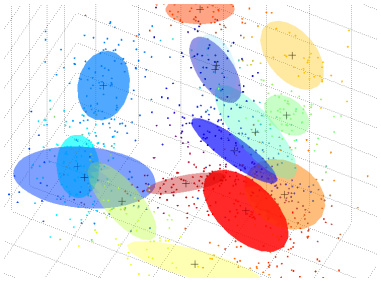

Fig. 5. GMM Visualization

$$\sigma_{kl} = \sqrt{\frac{\sum_{i=1}^{n_k} (f_{ilk} - \mu)^2}{n_k - 1}} \quad (6)$$

where n_k is the number of items belong to cluster k, f_{ilk} is the value of l^{th} of f in six flavour taste of ith item and μ is the mean value of all f_l, in group k. We calculate $G(x, y)$ 6 times for 6 field $f_1 - f_6$ for each pair items over all items of a group to sort in descending to find top best results.

4.4 Levenshtein distance for comparison

The flavour tags also play a quite significant role in the final results. We treat tags as vital as each element in 6 flavour taste. To compare and measure the similarity between two tags of string type, we use a good of levenshtein distance to solve it [8], [33]. The equation of the levenshtein distance is defined below:

$$lev_{a,b}(i,j) = \begin{cases} max(i,j), \text{if min}(i,j)=0 \\ min = \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1, \text{otherwise} \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} \end{cases} \tag{7}$$

4.5 Final sorting formula

Combine weight calculating function for 6-axis flavour taste and tags comparison with levenshtein distance (LD), we establish a equation for sorting to get final results as below:

$$S(i,j) = \sum_{k=1}^{K} \sum_{l=1}^{6} G_{kl}(i,j) + lev_{tags}(i,j) \tag{8}$$

where G_{kl} is the weight function corresponding l^{th} in 6-axis flavour taste between $item_i$ and $item_j$ in cluster k (k = {1, ..., K} K groups), $lev_{tags}(i,j)$ is the levenshtein function to compare tags similarity of those two items. We determine that the bigger $S(i,j)$, the better similar between those two items, so we sort by descending order all items of a cluster and return top m items having bigger S(i, j) value.

4.6 Proposed model pseudo code

For clearly, we give the proposed model its algorithm execution process to help readers more easily visualize and imagine our entire process. Let see pseudo code below:

Algorithm 1: Framework model proposal

Input: number of clusters k
Output: Top m other similar items of each item
Data: Dataset L
1. Data pre-processing for text fields
2. Build a matrix for 6-dimension vectors representing for six flavour taste $(f_1 - f_6)$
3. Taking the matrix as an input of GMM to train and save corresponding cluster of each item into dataset
4. **for** *item in dataset* **do**
   - Get cluster number of item
   - Find all items that have the same clusters
   - Applying equation $S(i,j)$ "(8)" for each pairs items
   - Return top m similar items by sorting decreasingly

end

5 Experiments

To prove the validity of our proposed model, we compare our proposed model to two other popular algorithms widely used in CB systems such as BOW+GFF and GMM+ED. We also illustrate the impact of GMM cluster into the accuracy and the efficiency of Gaussian filter equation in sorting results rather than those of cosine or euclidean distance.

5.1 Evaluation method

The evaluation method of recommendation systems commonly used is Root Mean Square Error (MSE) that is the average of the square errors [28]. It is defined as:

$$MSE = \frac{1}{N}\sum_{1}^{n}(r_i - \widehat{r}_i)^2, \tag{9}$$

where r_i is the predicted representing vector item, and $\widehat{r}_i$ is the original representing vector item.

We also use the recommendation results in Sakenowa as the standard measure to compare with our three algorithms because the sake website has so much reputation, popularity, being well-known for commercial purpose in Japan for many years. Beside, the recommendation results of the Sakenowa is also very impressive.

5.2 Experimental analysis

Some experiments will be conducted to verify the impact of GMM, GFF on probabilistic recommendation problem. Our main proposed model was implemented through such steps as data statistic, data cleaning, data missing value filling, clustering all items into different clusters and eventually using Gaussian filter + levenshtein distance to sort the results. To verify the effective impact of GMM and GFF on better prediction, we divided our experiments into three parts. Firstly, we use Bag-of-Word(BOW) [1] algorithm on some properties like flavour tags before applying GFF for sorting results. In the second way, we apply GMM + ED to clarify the influence of GMM. Finally, we implemented our main proposed model to prove the impact of GMM+GFF then give some comparison. All experiments will be unraveled in detail below:

Experiment 1: BOW+GFF The reason for this experiment is to verify the impact on result accuracy of GMM compared to BOW algorithm. Therefore, in the experiment, we will implement BOW algorithm comprised with GFF used for sorting on our liquor dataset. Firstly, we do some data preprocessing for text data like stemming, replace synonyms, filling missing data, etc [23]. As it was mentioned above, all important text fields were written behind Japanese form, so we use some tools offered for Japanese preprocessing like Ginza [9], Janome [10], JapaneseStemmer [19] was inspired by Porter Stemming Algorithm [29], etc. Before using GFF for sorting, we use BOW on these preprocessed properties to find the vector matrix representing for the item. The next step, we feed the vector matrix into K-nearest neighbors (k-NN) algorithm using unsupervised k-NN Scikit-Learn [21] to find top similar items based on these vectors. In these top items, we apply equation S (8) to get the best similar items.

Experiment 2: GMM+ED To demonstrate the impact of GMM, firstly, we still apply some preprocessing steps for text fields as the experiment above. After that, we build a matrix of 6 dimensions representing for 6 flavour taste, then feed it into GMM for training, save all cluster result for each item. Next step, we convert a collection of text flavour tags into a matrix of token counts using CountVectorizer of Scikit-Learn [21] and concatenate along same axis with the matrix of 6 dimensions for sorting. Finally, to return the best similar items of a given item, we just jump up to the cluster containing it and apply ED for sorting the results and get top best similar items of the given item.

Experiment 3: GMM+GFF Our two above experiment to prove the important role of GMM and GFF in our proposed model. In this experiment, in the first place, we also do preprocess for text fields as same steps in two previous experiments. After that, we build a matrix of 6 dimensions representing for 6 flavour tastes and feed the matrix into GMM for training purpose, then save cluster results for each items. To find top best similar items of a given item, we jump up into the cluster the query item lied in, consider the query item as center then apply (8) equation pair in pair with all items in the cluster, then sorting discerningly to return the top best similar items.

5.3 Experimental results and comparison

In this section, we compare our proposed algorithm with the results from the Sakenowa website and two other popular CB algorithms. The recommendation results from Sakenowa for each item are returned from an api [5]; therein, $f_{1...6}$ in the api are the value for each flavour taste, respectively. We conclude that our result accuracy outweights the Sakenowa and these two algorithm counter parts. Let see some charts below:

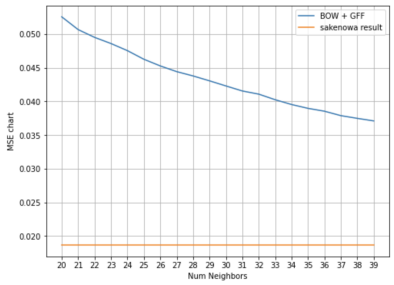

Fig. 6. MSE applied BOW+GFF

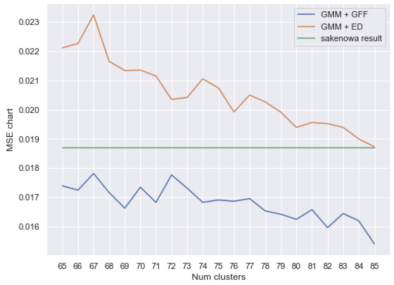

Fig. 7. MSE applied GMM+GFF and GMM+ED

All three experiments return top ten(10) best similar items for each item in dataset. In Fig. 6, list values of MSE are shown through an array of number of neighbors ranging from [25-39] in k-NN algorithm. Despite the tendency of decrease, but it is insignificant and the time response is extremely slow due to bigger number of neighbors.

[5] https://sakenowa.com/api/v1/brands/flavor?f=0&fv=f$_1$, f_2, f_3, f_4, f_5, f_6

In Fig. 7 the gap of MSE between GMM+ED and GMM+GFF is shown. It is very clearly seen that GMM+GFF generate better results than the other that verify the effect of GMM in sorting results. Both these two experiments show the effect of the number of clusters of GMM ranging from [65-85]. In Fig. 8, we compare our prediction results of all items in dataset to the recommendation results from the Sakenowa and construct a list of similarity proportion affected by the number of clusters.

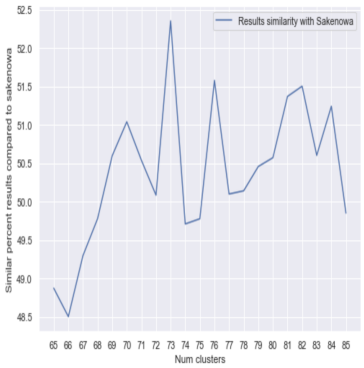

Fig. 8. Similar percent statistic compare to Sakenowa

Table 2. MSE affected by number of clusters

N clusters	GMM+ED	GMM+GFF	Sakenowa results
65	0.02211	**0.01739**	0.01868
70	0.02135	**0.01734**	0.01868
75	0.02074	**0.01691**	0.01868
80	0.01939	**0.01625**	0.01868
85	0.01873	**0.01541**	0.01868

In Tab. 2 and Tab. 4, we build a table of statistic of MSE generated from GMM+ED, BOW+GFF, GMM+GFF and recommendation results from Sakenowa. It is matter of fact that our GMM+GFF algorithm outperforms all the others method that demonstrate the effective of our algorithm. Further more, our time response in Tab. 3 also beat these two others, GMM+ED and BOW+GFF.

Table 3. Time response per query

BOW+GFF	GMM+ED	GMM+GFF
0.1856s	0.0174s	**0.0156s**

Table 4. MSE affected by number of neighbors

Num neighbors	BOW+GFF	Sakenowa results
20	0.05254	0.01868
25	0.04624	0.01868
30	0.04228	0.01868
35	0.03895	0.01868
39	0.03709	0.01868

6 Conclusion

We have proposed an very effective algorithm for recommendation system using content-based features with GMM. We have applied our proposed method for solving liquor recommendations. Further, our probabilistic-based recommendation systems not only acquire a remarkable prediction accuracy, but also has very speedy prediction time response for real-time application.

References

[1] Sounak Bhattacharya and Ankit Lundia. “MOVIE RECOMMENDATION SYSTEM USING BAG OF WORDS AND SCIKIT-LEARN”. In: 2019.

[2] Dirk Bollen et al. “Understanding choice overload in recommender systems”. In: Jan. 2010, pp. 63–70. DOI: 10.1145/1864708.1864724.

[3] Rui Chen et al. “A Hybrid Recommender System for Gaussian Mixture Model and Enhanced Social Matrix Factorization Technology Based on Multiple Interests”. In: *Mathematical Problems in Engineering* 2018 (Oct. 2018), pp. 1–22. DOI: 10.1155/2018/9109647.

[4] Kong Fan-sheng. “Hybrid Gaussian pLSA model and item based collaborative filtering recommendation”. In: *Computer Engineering and Applications* (2010).

[5] Dilan Görür and Carl Rasmussen. “Dirichlet Process Gaussian Mixture Models: Choice of the Base Distribution”. In: *J. Comput. Sci. Technol.* 25 (July 2010), pp. 653–664. DOI: 10.1007/s11390-010-9355-8.

[6] Vairaprakash Gurusamy and Subbu Kannan. “Preprocessing Techniques for Text Mining”. In: Oct. 2014.

[7] Ido Guy and David Carmel. “Social Recommender Systems”. In: Jan. 2011, pp. 283–284. DOI: 10.1145/1963192.1963312.

[8] Rishin Haldar and Debajyoti Mukhopadhyay. “Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach”. In: *Computing Research Repository - CORR* (Jan. 2011).

[9] Mai Hiroshi and Masayuki. “” In: *25* (2019). URL: http://www.anlp.jp/proceedings/annual_meeting/2019/pdf_dir/F2-3.pdf.

[10] Janome$_p y$. *Janome*. 2019. URL: https://github.com/mocobeta/janome.

[11] Shah Khusro, Zafar Ali, and Irfan Ullah. “Recommender Systems: Issues, Challenges, and Research Opportunities”. In: Feb. 2016, pp. 1179–1189. ISBN: 978-981-10-0556-5. DOI: 10.1007/978-981-10-0557-2_112.

[12] Shah Khusro, Zafar Ali, and Irfan Ullah. “Recommender Systems: Issues, Challenges, and Research Opportunities”. In: Feb. 2016, pp. 1179–1189. ISBN: 978-981-10-0556-5. DOI: 10.1007/978-981-10-0557-2_112.

[13] Dar-Shyang Lee, Jonathan Hull, and B. Erol. “A Bayesian framework for Gaussian mixture background modeling”. In: vol. 3. Oct. 2003, pp. III–973. DOI: 10.1109/ICIP.2003.1247409.

[14] Leo Liberti et al. “Euclidean Distance Geometry and Applications”. In: *SIAM Review* 56 (May 2012). DOI: 10.1137/120875909.

[15] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. “Content-based Recommender Systems: State of the Art and Trends”. In: Jan. 2011, pp. 73–105. DOI: 10.1007/978-0-387-85820-3_3.

[16] Yang Lu, Xuemei Bai, and Feng Wang. “Music Recommendation System Design Based on Gaussian Mixture Model”. In: *ICM 2015*. 2015.

[17] Linyuan Lü et al. “Recommender systems”. English. In: *Physics Reports* 519.1 (Oct. 2012), pp. 1–49. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2012.02.006.

[18] Prem Melville and Vikas Sindhwani. “Recommender Systems”. In: Jan. 2011, pp. 829–838. DOI: 10.1007/978-0-387-30164-8_705.

[19] MrBrickPanda. *Japanese Stemmer*. 2019. URL: https://github.com/MrBrickPanda/Japanese-stemmer.

[20] Suphakit Niwattanakul et al. “Using of Jaccard Coefficient for Keywords Similarity”. In: Mar. 2013.

[21] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python”. In: *Journal of machine learning research* 12.Oct (2011), pp. 2825–2830.

[22] Simon Philip, Peter Shola, and Ovye Abari. “Application of Content-Based Approach in Research Paper Recommendation System for a Digital Library”. In: *International Journal of Advanced Computer Science and Applications* 5 (Oct. 2014). DOI: 10.14569/IJACSA.2014.051006.

[23] Reza Rahutomo et al. “Preprocessing Methods and Tools in Modelling Japanese for Text Classification”. In: Aug. 2019. DOI: 10.1109/ICIMTech.2019.8843796.

[24] Martin Rajman and Romaric Besançon. “Text Mining: Natural Language techniques and Text Mining applications”. In: *Proceedings of the 7th IFIP Working Conference on Database Semantics (DS-7)* (Jan. 1997). DOI: 10.1007/978-0-387-35300-5_3.

[25] Carl Rasmussen. “The Infinite Gaussian Mixture Model”. In: vol. 12. Apr. 2000, pp. 554–560.

[26] Douglas Reynolds. “Gaussian Mixture Models”. In: *Encyclopedia of Biometrics* (Jan. 2008). DOI: 10.1007/978-0-387-73003-5_196.

[27] Philip Sedgwick. “Pearson’s correlation coefficient”. In: *BMJ* 345 (July 2012), e4483–e4483. DOI: 10.1136/bmj.e4483.

[28] Guy Shani and Asela Gunawardana. “Evaluating Recommendation Systems”. In: vol. 12. Jan. 2011, pp. 257–297. DOI: 10.1007/978-0-387-85820-3_8.

[29] Karen Sparck Jones and Peter Willett, eds. *Readings in Information Retrieval*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997. ISBN: 1558604545.

[30] Hangyu Yan and Yan Tang. “Collaborative Filtering based on Gaussian Mixture Model and Improved Jaccard Similarity”. In: *IEEE Access* PP (Aug. 2019), pp. 1–1. DOI: 10.1109/ACCESS.2019.2936630.

[31] Kazuyoshi Yoshii et al. “Hybrid Collaborative and Content-based Music Recommendation Using Probabilistic Model with Latent User Preferences.” In: Jan. 2006, pp. 296–301.

[32] Bo Zhu, Jesus Bobadilla, and Fernando Ortega. “Reliability quality measures for recommender systems”. In: *Information Sciences* (May 2018).

[33] B. Ziolko et al. “Modified Weighted Levenshtein Distance in Automatic Speech Recognition”. In: Jan. 2010.

[34] Harry Zisopoulos et al. “Content-Based Recommendation Systems”. In: (Nov. 2008).