Analisis dan Implementasi Prediksi Rating pada Memory-based Collaborative Filtering dengan Menggunakan Smoothing

Analysis and Implemantation Rating Prediction om memory-based Collaborative Filtering with Smoothing

Hafiz Dewanto[1], Agung Toto Wibowo,ST.MT,[2]
[1,2]Fakultas Informatika Universitas Telkom, Bandung
[1]hafizdewanto@gmail.com, [2]atwbox@yahoo.com,

Abstrak

Teknik *Collaborative Filtering* (CF) telah dikenal sebagai salah satu teknik yang paling sukses didalam *Recommender System*, dimana teknik ini memanfaatkan informasi dan preferensi dari *user* atau *item* lain untuk memberikan rekomendasi *item*. Ada dua tipe algoritma CF, yaitu *memory-based* dan *model-based* yang memiliki kelebihan dan kekurangan masing-masing. Pada penelitian ini, digunakan algoritma *memory-based* CF dengan teknik *smoothing*, dimana teknik *smoothing* mampu membantu kelemahan *memory-based* CF dalam hal kekurangan data rating yang kosong atau disebut *sparsity*.

Berdasarkan hasil pengujian, algoritma *memory-based* CF dengan teknik *smoothing* mampu menurunkan error sistem yang diukur dengan *Mean Absolute Error* (MAE) dari 0,8581 menjadi 0,8483 atau menurun sebesar 1,14% dibandingkan dengan menggunakan algoritma *memory-based* saja.

Kata kunci *:Collaborative Filtering, Recommender System, Memory-based* CF, *smoothing*, MAE

Abstract

Collaborative Filtering technique has been proved to be one of the most successful techniques in Recommender System, where this technique utilizes the information and the preference of other users or items to provide item recommendations. There are two types of algorithm for CF : memory-based CF and model-based CF. On this research we use memory-based CF with smoothing technique which can handle sparsity.

Based on research, the Collaborative Filtering recommendation method memory-based CF with smoothing technique can reduce system error measured by Mean Absolute Error (MAE) from 0,8581 to 0,8483 or decline by 1,14% compared to memory-based algorithm only.

Keywords : *Collaborative Filtering, Recommender System, Memory-based CF, smoothing, MAE*

1. Pendahuluan

Di era digital seperti saat ini, pertumbuhan konten-konten digital begitu pesatnya bertambah dan berkembang setiap harinya. *Item-item* digital seperti berita, video, film, buku, musik, bahkan sampai media social seperti twitter dan instagram tidak luput peningkatan yang sangat pesat. Hal ini memungkinkan *user* untuk berhadapan pada banyaknya pilihan *item-item* tersebut, yang secara langsung atau tidak langsung, memaksa *user* untuk memilih *item* tersebut [5]. Pertumbuhan jumlah film yang sangat pesat membuat ketersediaan informasi film menjadi semakin banyak dan menyulitkan user dalam mendapatkan informasi mengenai film yang sesuai dengan kegemarannya. Untuk mengatasi hal tersebut, dibutukan *recommender system*, yaitu sebuah sistem rekomendasi yang dapat memberikan rekomendasi sebuah atau beberapa film kepada user sesuai dengan karakteristik kegemaran user tersebut.

Collaborative filtering (CF) merupakan salah satu metode yang digunakan dalam *recommender system*, yang memiliki 2 algoritma utama yaitu *memory-based* CF dan *model-based* CF [8]. *Memory-based* CF mampu mengidentifikasi similiaritas dari dua user dengan membandingkan rating yang mereka berikan kepada suatu set item, *memory-based* CF memiliki dua kelemahan yaitu *sparsity* dan *scallability*. *Sparsity* adalah keadaan dimana banyaknya data yang kosong, sedangkan *scallability* adalah keadaan dimana *memory-based* tidak mampu bekerja optimal dikarenakan jumlah user dan *item* terlalu besar. Teknik *smoothing* hadir untuk menjawab permasalahan *sparsity*. Dalam

penelitian ini, akan digunakan algoritma *memory-based* CF dengan teknik *smoothing*. Pada *memory-based* CF digunakan algoritma *user-based* untuk menganalisis persamaan diantara user dengan membandingkan rating yang mereka lakukan dan film yang mereka gemari, kemudian mengisi rating yang kosong dari matriks *user-item* dengan prediksi rating sementara, hal ini disebut data *smoothing*. Kemudian matriks yang penuh hasil data *smoothing* dicari kemiripan antar item untuk kemudian dihasilkan prediksi nilai rating yang final, tahap ini disebut juga dengan algoritma *item-based*.

Hasil penelitian menunjukkan bahwa algoritma *memory-based collaborative filtering* dengan teknik *smoothing* mampu memberikan performansi yang lebih baik bila dibandingkan dengan algoritma tradisional biasa, yaitu *memory-based* saja [1].

2. Landasan Teori

2.1 Sistem Rekomendasi

Sistem rekomendasi membantu kita dalam mengatasi masalah *information overload* dengan menyediakan saran-saran yang bersifat personal berdasarkan pada riwayat perilaku pengguna sebelumnya. Ada dua pendekatan dalam membangun sistem rekomendasi, yaitu *Collaborative Filtering* (CF) dan *Content-Based* (CB).Pada metode CB, rekomendasi dibuat dengan menganalisis deskripsi setiap item untuk mengidentifikasi item mana yang mempunyai ketertarikan khusus dari user. Deskripsi ketertarikan user diproleh dari profile user yang didasarkan atas penilaian user terhadap item tersebut pada sistem rekomendasi, misalnya dengan cara memberikan rating [8].

2.2 Collaborative Filtering

Sebagai salah satu pendekatan yang paling sukses untuk membangun sistem rekomendasi, CF menggunakan preferensi/penilaian yang dilakukan oleh user atau item lain. Dalam hal ini, dikenalkan tentang CF dan tantangan-tantangan yang ada, seperti data *sparsity* (kelangkaan data), *scalability* (performansi), *synonym* (kesamaan data), *gray sheep*, serangan manipulasi, proteksi privasi, dan lain-lain. Teknik (algoritma) dari CF ada 3 yaitu, *memory-based*, *model-based*, dan *hybrid* (kombinasi, contoh : *memory-based* dan *model-based*) [8,9]. CF pada sistem rekomendasi dengan menggunakan teknik-teknik tersebut, mampu memprediksi rating pada item dari user.

Table 1 Contoh data pada CF

	Film			
User		The Hobbits	Batman	Ironman
	Budi	2	3	4
	Joko	3	?	1
	Wisnu	?	5	?
	Tuti	4	?	2
	Siti	1	2	?

Sebagai contoh pada Tabel 1, user Joko memberikan rating pada film The Hobbits dan Ironman, tapi tidak pada film Batman. Dengan perhitungan tertentu, sistem dapat memprediksi preferensi Joko terhadapat film Batman dengan mempertimbangkan rating-rating yang telah diberikan user lain pada film Batman tersebut.

Terdapat 2 algoritma pada CF yang juga digunakan dalam Tugas Akhir ini, yaitu :

a. User-based CF

User-based CF merupakan bagian dari *memory-based* CF, yang menghitung *similarity* antar user. Terdapat beberapa algoritma *similarity* yang dapat digunakan, yaitu *Pearson correlation, vosine vector similarity, adjusted cosine vector similarity, mean-squared difference and Spearman correlation.* Contoh algoritma user-based CF yang dipakai pada Tugas Akhir ini adalah *Pearson correlation* seperti pada persamaan 1:

$$(\quad) \quad \frac{\Sigma \quad (\quad)(\quad)}{\sqrt{\Sigma \quad (\quad)} \, \Sigma \quad (\quad)} \qquad (1)$$

Ri,c adalah rating item C oleh user i. Ai adalah rata-rata rating dari user i terhadap semua co-rated item. Iij adalah kumpulan ite yang dirating oleh user i dan user j.

b. Item-based CF

Item-based CF merupakan bagian dari *model-based CF, yang menghitung similarity* antar item. Algoritma item-based CF yang dipakai pada Tugas Akhir ini adalah *Pearson* correlation untuk menghitung *similarity* antar item, seperti pada persamaan 2 :

$$(\quad) \quad \frac{\Sigma \quad (\quad)(\quad)}{\sqrt{\Sigma \quad (\quad)} \, \Sigma \quad (\quad)} \qquad (2)$$

Rit adalah rating dari user i untuk item t. Rir adalah rating dari user i untuk item r. At adalah rata-rata rating yang diberikan oleh co-rated user terhadap item t. Ar adalah rata-rata rating yang diberikan co-rated user terhadap item r. m adalah jumlah co-rated user yang merating item t dan item r.

2.3 MAE

Mean Absolute Error (MAE) adalah nilai rata-rata error dari hasil prediksi. MAE dapat dihitung menggunakan persamaan 3 berikut :

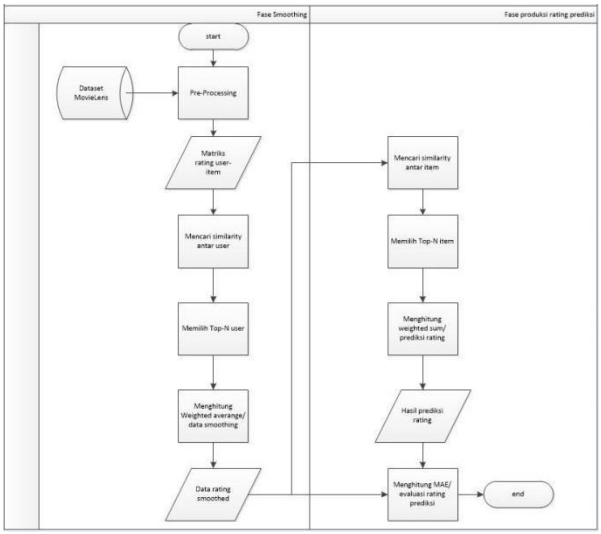
Σ _____ (3)

Dimana n adalah user, p adalah rating prediksi dan q adalah rating asli.

3. Analisa dan Perancangan Sistem

3.1 Deskripsi dan Analisa Sistem

Berikut adalah gambar yang menunjukkan gambaran umum sistem :



Gambar 1 Gambaran Umum Sistem

Langah pertama adalah me-*preprocessing* data sesuai kebutuhan sistem yaitu mengubahnya menjadi matriks *user-item*. Proses selanjutnya adalah menghitung user rating similiarity menggunakan Pearson correlation, kemudian mendapatkan top-N user yang memiliki nilai similiarity terbesar, lalu dilanjutkan dengan mengisi nilai-nilai rating yang masih kosong dari target user ke target item dengan menggunakan persamaan *weighted average*, tahapan ini disebut juga sebagai proses data *smoothing*. Selanjutnya data yang sudah di*smoothing* tadi masuk keproses MAE dan proses perhitungan kemiripan antar item. Pada proses perhitungan antar user, digunakan matriks yang telah penuh dengan teknik data *smoothing* untuk dicari kemiripan antar item dengan persamaan *Pearson correlation*. Setelah didapatkan nilai kemiripan antar semua item, dipilihlah top-N item, yaitu item-item yang memiliki nilai kemiripan paling besar diantara item-item lainnya, kemudian dengan persamaan *weighted sum* maka didapatkanlah hasil prediksi rating tiap item. Analisis performansi dilakukan dengan menggunakan persamaan MAE, hasil MAE dari *memory-based* saja kemudian akan dibandingkan dengan hasil MAE *memory-based* dengan teknik data *smoothing*.

3.2 Perancangan Data

Dataset yang digunakan pada penelitian ini adalah dataset dari MovieLens yang memiliki rating sejumlah 100 000 yang berasal dari 1000 user yang merating 1680 film. Dataset tersebut terbagi 2 yaitu dataset untuk data traning dan dataset untuk data testing, yang sudah kita dapatkan dari MovieLens. Dari dataset tersebut, dipilih secara acak user yang sudah memberikan rating minimal 100, dan juga dipilih secara acak item yang sudah diberikan rating minimal 100 rating.. Dataset MovieLens yang diberikan berupa tabel raing dengan user id, item id, dan nilai rating sebagai kolomnya.

Table 2 Contoh tabel data rating MovieLens

User_id	Item_id	Rating
1	1	5
1	4	4
2	1	3
3	3	5
4	3	1

Tabel 2 merupakan contoh tabel data rating yang memiliki user_id dari 1 sampai 4 dan item_id 1 sampai 4.

Table 3 Contoh Matriks user-item

		Item			
User_id		1	2	3	4
	1	5	0	0	4
	2	3	0	0	0
	3	0	0	5	0
	4	0	0	1	0

Untuk memudahkan dalam pemrosesan, tabel 2 diubah kedalam bentuk matriks user-item seperti pada tabel 3 dengan user_id sebagai baris matriks, item_id sebagai kolom matriks dan isi sel matriks adalah rating. Jika user belum memiliki rating pada item tertentu, nilai rating 0.

4.1 Pengujian Sistem

Bagian ini akan menjelaskan mengenai tujuan pengujian untuk menjawab permasalahan yang diangkat dan menjelaskan strategi pengujian yang dibuat untuk memecahkan permasalahan tersebut.

4.1.1 Strategi Pengujian

Untuk menjawab tujuan pada penelitian ini, maka dibuat strategi pengujian yang dibagi menjadi dua buah skenario pengujian.

4.1.2.1 Skenario 1 – Mendapatkan nilai MAE untuk *memory-based* saja

Skenario ini dilakukan bertujuan untuk mendapatkan nilai MAE dari algoritma *memory-based*, yaitu *user-based* CF, dimana nilai MAE ini akan menjadi tolak ukur performansi dari algoritma *memory-based* dengan *smoothing*.

Seperti yang sudah dijelaskan pada subbab perancangan data, jumlah user yang digunakan berjumlah 943 user, jumlah item yang digunakan sejumlah 1682 item. Sedangkan parameter peubah untuk bisa mendapatkan parameter yang optimal adalah nilai top-N, dimana nilai top-N akan dimulai dari n=5 sampai n=15 dengan pertambahan 5 setiap percobaan, sehingga didapatkan sejumlah 3 buah hasil pengujian.

Konfigurasi parameter optimal didapatkan dengan melihat nilai MAE yang paling kecil dari 3 buah hasil pengujian tersebut. Hasil dari mendapatkan konfigurasi optimal di skenario 1 ini akan digunakan dan dianalisis kembali pada skenario 2.

4.1.2.2 Skenario 2 – Mendapatkan nilai MAE untuk algoritma *memory-based* dengan teknik *smoothing*

Skenario ini dilakukan bertujuan untuk mendapatkan nilai MAE dari algoritma *memory-based* dengan teknik *smoothing*..

Seperti yang sudah dijelaskan pada subbab perancangan data, jumlah user yang digunakan berjumlah 943 user, jumlah item yang digunakan sejumlah 1682 item. Sedangkan parameter peubah untuk bisa mendapatkan parameter yang optimal adalah nilai top-N, dimana nilai top-N akan dimulai dari n=5 sampai n=15 dengan pertambahan 5 setiap percobaan, sehingga didapatkan sejumlah 3 buah hasil pengujian.

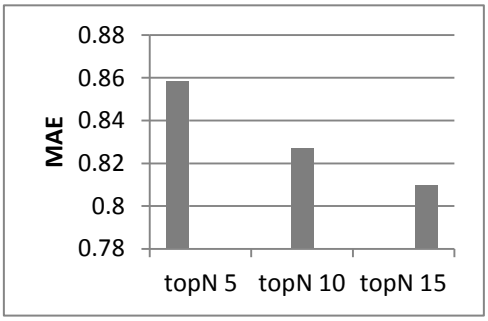
Konfigurasi parameter optimal didapatkan dengan melihat nilai MAE yang paling kecil dari 3 buah hasil pengujian tersebut. Hasil MAE dari pengujian skenario 2 ini kemudian akan dibandingkan dengan hasil MAE pengujian skenario 1.

4.2 Analisis dan Hasil Pengujian

Bagian ini akan menjelaskan tentang analisis hasil pengujian dari setiap skenario pengujian sistem yang sudah dirancang.

4.2.1 Analisis dan Hasil Pengujian Skenario 1

Skenario ini dilakukan dengan menggunakan algoritma *memory-based* dengan pergantian parameter top-N user mulai dari n=5 sampai dengan n=15 dengan kelipatan 5.

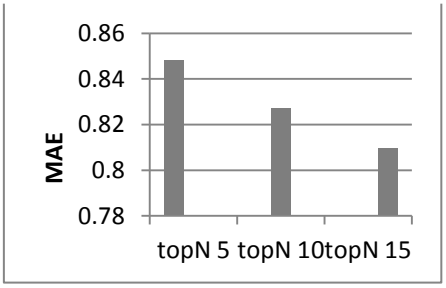


Gambar 2 Chart hasil MAE

Didapatkan nilai MAE terendah berada pada top-N bernilai 15, hal ini dikarenakan ketika top-N user bernilai 15, informasi preferensi user sudah cukup baik sehingga mampu menghasilkan nilai MAE yang kecil. Sebaliknya pada saat top-N bernilai 5 dan 10, banyaknya user similar yang dipilih masih terlalu sedikit atau belum cukup, sehingga sistem masih kekurangan informasi preferensi user yang didapat dari rating user.

4.2.2 Analisis dan Hasil Pengujian Skenario 2

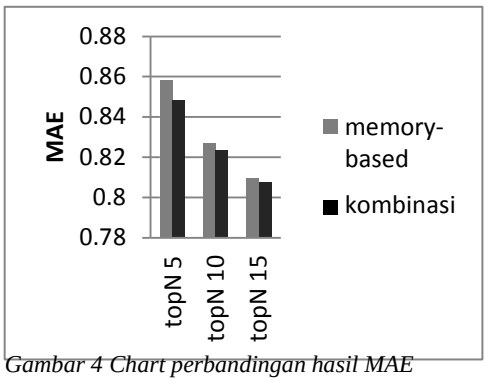
Skenario ini dilakukan dengan menggunakan algoritma kombinasi *memory-based* dan *item-based* dengan pergantian parameter top-N item mulai dari n=5 sampai dengan n=15 dengan kelipatan 5.



Gambar 3 Chart hasil MAE

Didapatkan nilai MAE terendah berada pada top-N bernilai 15, hal ini dikarenakan ketika top-N item bernilai 15, informasi preferensi item sudah cukup baik sehingga mampu menghasilkan nilai MAE yang kecil. Sebaliknya pada saat top-N bernilai 5 dan 10, banyaknya item similar yang dipilih masih terlalu sedikit atau belum cukup, sehingga sistem masih kekurangan informasi preferensi item yang didapat dari rating user.

Kemudian perbandingan hasil MAE dari 2 algoritma diatas ditunjukkan pada gambar dibawah ini.



Gambar 4 Chart perbandingan hasil MAE

Didapatkan hasil bahwa algoritma *memory-based* dengan teknik *smoothing* mampu menurunkan nilai MAE dari 0.8581 menjadi 0.8483 atau turun sebesar 1,14%.

5.1 Kesimpulan

Berdasarkan analisis dan hasil pengujian yang dilakukan pada penelitian ini, dapat disimpulkan **:**

1. Strategi *memory-based* dengan teknik *smoothing* mampu menurunkan nilai MAE sebesar 1,14% dibandingkan dengan teknik *memory-based* CF saja.

2. Pemilihan top-N berpengaruh terhadap nilai MAE, tidak ada pengukuran yang pasti mengenai pemilihan top-N, oleh karena itu sebaiknya penelitian menggunakan top-N sebanyak-banyaknya untuk kemudian dapat ditentukan top-N ke-berapa yang paling baik menghasilkan nilai MAE yang bagus.

5.2 Saran

Berikut adalah saran yang dapat dijadikan pertimbangan untuk penelitian selanjutnya :

1. Walaupun algoritma *memory-based* dengan teknik *smoothing* mampu menghasilkan MAE yang lebih kecil dibandingkan dengan tradisional CF, namun penurunan nilai MAE tidak terlalu siginifikan jumlahnya, yaitu hanya 1,14%, oleh sebab itu bisa ditambahkan beberapa strategi seperti *user/item clustering*, *fusion*, dan lain sebagainya.
2. Bisa ditambahkan algoritma pembandingnya, seperti algoritma *memory-based* CF lainnya, yaitu *item-based* CF saja.

Daftar Pustaka

[1] Kunegis, J., & Albarak, S. (2007). Adapting Ratings in Memory-Based Collaborative Filtering using Linear Regression. *in 'IRI', IEEE Systems, Man, and Cybernetics Society* , 49-54.

[2] Lee, J., Sun, M., & Lebanon, G. (2012). A Algorithms.

[3] Ma, H., King, I., & Lyu, M. R. (2007). Effective Missing Data Prediction for Collaborative Filtering.

[4] O'Connor, M., & Herlocker, J. (1999). Clustering Items for Collaborative Filtering. *In Proceedings of the ACM SIGIR Workshop on Recommender Systems: Algorithms and Evaluation, 22nd International Conference on Research and Development in Information Retrieval* , 11-14.

[5] Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (2010). *Introduction to Recommender Systems Handbook.* Springer.

[6] Sarkar, M., & Leong, T. Y. (2001). Fuzzy K-means Clustering with Missing Values. *in Proceedings - AMIA Symposium, Journal of the American Medical Informatics Association* .

[7] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. *Proceedings of the 10th international conference on World Wide Web* .

[8] Su, X., & Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence* .

[9] Gong, Ye, & Tan. (2009). Combining Memory-based and Model-based Collaborative Filtering in Recommender System. *Pacific-Asia Conference on Circuits Communications and System.*

[10] Tang, T. Y., & McCalla, G. (2003). Mining Implicit Ratings for Focused Collaborative Filtering for Paper Recommendations. *in Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems, 18th International Joint Conference on Artificial Intelligence* .

[11] Ungar, L. H., & Foster, D. P. (1998). Clustering Methods for Collaborative Filtering. *in 'Proceedings of the Workshop on Recommendation Systems*.