1. **Pengertian *Recommender Systems***

*Recommender* *Systems* (RS) merupakan sebuah fitur yang memberikan saran beberapa konten yang mungkin relevan atau disukai oleh pengguna. Saran-saran tersebut berhubungan dengan berbagai proses pengambilan keputusan, seperti barang apa yang harus dibeli, musik apa yang harus didengarkan, atau berita apa yang harus dibaca. “item” adalah istilah umum yang digunakan untuk menunjukan apa yang direkomendasikan sistem kepada pengguna [1].

Dewasa ini, hampir semua aplikasi atau Web yang menyajikan konten seperti Google, Facebook, Twitter, Amazon, Spotify, Netflix, dan sejenisnya telah memiliki fitur tersebut. Bagi beberapa perusahaan, keberadaan *recommender system* menjadi penting untuk membantu menjalankan tugas dan mencapai tujuan perusahaan sebagaimana mestinya [2]. Kenyataannya, implementasi *recommender system* yang baik mampu menghasilkan *traffic* yang sangat tinggi bagi perusahaan. Misalnya, Spotify mengungkapkan bahwa lebih dari 8.000 musisi mendapat *listener* bulanan melalui fitur Discover Weekly [3]. Hal yang sama juga diungkapkan oleh Netflix bahwa 75% film yang disaksikan pengguna didapat dari *recommender system* [4].

1. ***Data* dan *Knowledge Source***

RS merupakan sistem pemrosesan informasi yang aktif mengumpulkan berbagai jenis data untuk membangun rekomendasi-rekomendasi. Data utama yang dibutuhkan adalah data *items* yang akan ditawarkan dan *user* yang akan menerima tawaran tersebut. Semenjak data yang tersedia untuk RS sangat beragam, maka bisa atau tidaknya data dimanfaatkan bergantung pada teknik rekomendasi yang digunakan [1].

Secara umum, terdapat teknik rekomendasi dengan sedikit pengetahuan karena hanya menggunakan data yang sederhana, seperti rating yang diberikan oleh *user* terhadap *item* [5], [6]. Teknik lain bergantung pada pengetahuan yang lebih, misalnya menggunakan deskripsi ontologis dari *user* dan *item* [7], hubungan sosial, dan tingkah laku *user* dalam sistem [8]. Bagaimanapun, data yang digunakan oleh RS biasanya mengacu pada tiga objek, yaitu *user, item* dan *transaction* [1].

***Item***adalah objek yang direkomendasikan. Kompleksitas dan nilai atau kegunaan suatu *item* dapat dijadikan sebagai karakteristik *item.* Suatu *item* dapat bernilai positif jika *item* tersebut berguna untuk *user*, atau bernilai negatif jika *item* tersebut tidak sesuai dengan *user*.

***User***mengakses sistem untuk berbagai macam tujuan dan karakteristik. biasanya data *user* dimodelkan sesuai dengan teknik rekomendasi yang digunakan. Misalnya, pada *collaborative filtering,* *user* dimodelkan sebagai matriks *user item* yang berisi rating yang diberikan oleh *user* untuk beberapa *item*.

***Transaction*** merupakan *log-like data* yang menyimpan informasi penting yang dihasilkan selama interaksi manusia-komputer dan berguna untuk membangun generasi algoritma rekomendasi yang digunakan. *Data Transaction* juga dapat berisi feedback eksplisit yang disediakan untuk diisi oleh *user* sebagai penilaian terhadap suatu *item*, seperti *rating*. Pada kenyataannya, *rating* adalah bentuk data transaksi paling populer yang digunakan pada RS. *Rating* dapat dikumpulkan secara eksplisit maupun implisit.

Dalam koleksi eksplisit, *user* diminta untuk memberikan pendapatnya terhadap suatu *item* pada skala tertentu. Menurut [9], beberapa jenis *rating* sebagai berikut:

* *Numerical rating*, seperti *rating* dengan skala 1-5 yang disediakan untuk rekomendasi buku oleh Amazon.com
* *Ordinal rating*, seperti “sangat setuju, setuju, netral, tidak setuju, sangat tidak setuju” biasanya melalui kuisioner.
* *Binary rating* yang menjadi model pilihan dimana *user* diminta untuk memilih *item* tertentu baik atau buruk.
* *Unary rating* yang menunjukan bahwa *user* telah mengamati atau membeli *item.*

Bentuk lain dari evaluasi pengguna adalah tag yang diasosiasikan oleh *user* dengan *item* yang disajikan dalam sistem. Pada Movielens RS, tag mewakili bagaimana perasaan pengguna Movielens tentang film, misalnya “terlalu lama” atau “akting” [8].

Dalam koleksi implisit, sistem bertujuan untuk menyimpulkan pendapat pengguna berdasarkan tindakan pengguna, seperti *click, typing,* dan *searching* yang dilakukan *user* dalam sistem.

1. **Metode-metode *Recommender Systems***

Seperti yang telah disebutkan diatas, bahwa data yang tersedia untuk RS sangat beragam. Maka dalam perkembangannya, RS menggunakan teknologi atau metode yang beragam. Kita dapat mengklasifikasikan sistem ini kedalam dua kelompok besar [10], yaitu sebagai berikut :

1. Content-based Recommender Systems

*Content-based* merekomendasikan *item* berdasarkan karakteristik *item* [10]. Misalnya, pengguna Netflix telah menonton banyak film koboi maka rekomendasi film yang akan diberikan oleh pengguna tersebut adalah daftar film yang bergenre “koboi”.

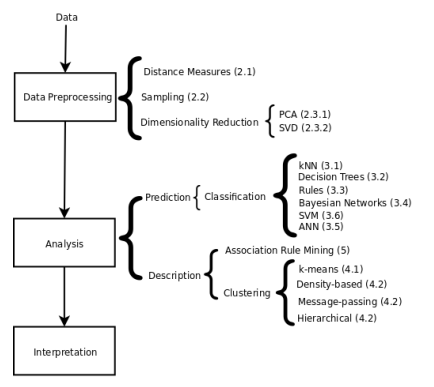
1. Collaborative Filtering Recommender Systems

*Collaborative filtering* merekomendasikan *item* berdasarkan kesamaan antar pengguna, antara pengguna dan *item*, atau antar *item* [10]. *Item* yang akan direkomendasikan kepada pengguna adalah *item* yang disukai oleh pengguna yang memiliki kesamaan. metode ini dapat menggunakan dasar pencarian *similarity* dan *clustering* namun dasar tersebut masih belum cukup memadai untuk melakukan rekomendasi sehingga banyak algoritma baru yang muncul dan terbukti efektif untuk RS.

1. **Data Mining for *Recommender Systems***

RS biasanya menerapkan teknik dan metodologi dari area terdekat seperti *Human Computer Interaction* (HCI) atau *Informasi Retrieval* (IR). Namun, sebagian besar sistem ini mengandung inti algoritma yang dapat dipahami oleh teknik *Data Mining* (DM). Bahkan, sebagian besar tantangan dalam DM [11] juga merupakan tantangan dalam *recommender systems*.

DM menggunakan metode dan teknik yang diambil dari *machine learning, artificial intelligence, statistics,* dan  *database systems*. Namun sebagian besar teknik "tradisional" ini perlu disesuaikan untuk memperhitungkan dimensi tinggi dan heterogenitas data yang meresap dalam masalah *data mining*. Proses DM biasanya terdiri dari 3 langkah yaitu, *Data Preprocessing* [12], *Data Analysis*, and *Result Interpretation* (lihat Gambar 1).



Gambar 1 Langkah-langkah Utama dan Metode dalam *Data Mining Problem*

1. *Data Preprocessing*

Data didefinisikan sebagai kumpulan objek dan atributnya. Dimana atribut didefinisikan sebagai properti atau karakteristik dari suatu objek. Ada berbagai jenis data dengan atribut-atribut yang bervariasi. Selain itu, *reallife* *data* biasanya perlu dipreprocess (misalnya cleansed, filtered, transformed) sebelum digunakan pada teknik *machine learning* dalam langkah analisis. Pemrosesan data ini dilakukan untuk menhilangkan adanya *missing point, duplicated data*, atau *noise* dalam *instance*. Jika dikaitkan dengan RS, dalam pemrosesan data terdapat tiga masalah yang sangat penting untuk membangun *recommender systems* [13] yaitu, meninjau *similarity* atau mengukur jarak antar data atau kumpulan titik data, penerapan *sampling* sebagai cara untuk mengurangi jumlah *item* dalam koleksi data yang sangat besar namun tetap mempertahankan karakteristik utamanya, dan penerapan teknik yang paling umum untuk mengurani dimensi data.

1. *Similarity Measures*

*Similarity* didefinisikan sebagai ukuran numerik. Sering jatuh dalam kisaran [0,1] yang menunjukkan bagaimana kesamaan antar *item*.Memiliki fungsi kesamaan yang sesuai adalah masalah utama

untuk banyak algoritma *data mining*. Biasanya, untuk sebuah ukuran numerik tentang bagaimana dua *item* yang berbeda mengacu pada fungsi jarak. Fungsi jarak yang paling umum adalah *euclidean distance* [13].



Dimana n adalah jumlah dimensi, xk dan yk adalah k atribut (komponen) dari masing-masing objek data x dan y. Perhatikan bahwa untuk menghitung *euclidean distance*, perlu untuk menormalkan data pada skala data yang berbeda. *Minkowski distance* adalah generalisasi dari *euclidean distance* [13].



dimana r adalah derajat jaraknya. *Minkowski distance* dibedakan berdasarkan jumlah r nya : Untuk r = 1, *the city block distance* (Manhattan, taxicab atau L1 norm) ; Untuk r = 2, *euclidean distance*; Untuk r → ∞, *supremum distance* (Lmax norm atau L∞ norm). *Mahalanobis distance* didefinisikan sebagai [13]:



Dimana σ adalah matriks kovarians dari data. Pendekatan lain yang sangat umum adalah mempertimbangkan item sebagai vektor dokumen dari suatu ruang n-dimensi dan menghitung kemiripannya sebagai cosinus dari sudut yang mereka bentuk [13]:

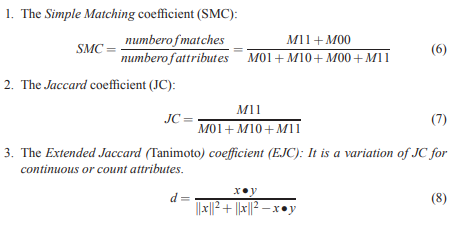


Dimana • menunjukkan *vector dot product* dan || x || adalah panjang vektor x. *Similarity* ini dikenal sebagai *cosine similarity* atau L2 Norm. *Similarity* antara item juga dapat dilihat melalui korelasi antar objek yang dibentuk. Meskipun ada beberapa koefisien korelasi yang dapat diterapkan, *pearson correlation* adalah yang paling umum digunakan [13]:



di mana Σ adalah kovarian dari titik data x dan y dan σ adalah standar deviasinya.

Beberapa *similarity measures* juga telah diusulkan untuk *items* yang memiliki atribut biner. Pertama, = jumlah atribut dimana x sama dengan 0 dan y sama dengan 1, = jumlah atribut dimana x sama dengan 1 dan y sama dengan 0, = jumlah atribut dimana x sama dengan 0 dan y sama dengan 0, dan = jumlah atribut dimana x sama dengan 1 dan y sama dengan 1. Dari jumlah tersebut dapat dihitung [13]:



Dalam membangun *recommender systems* khususnya dengan pendekatan *collaborative filtering* biasanya digunakan metode pengklasifikasian k-*Nearest Neighbor* [13]. Seperti metode pengklasifikasian dan pengelompokan lainnya, metode ini sangat bergantung pada pengukuran *similarity* yang sesuai. RS yang tradisional telah menggunakan *cosine similarity* atau *pearson correlation* untuk mengukur *similarity*. Walaupun, perhitungan *similarity* lain yang telah disebutkan diatas juga memungkinkan dalam konteks ini. [14] melakukan penelitian berskala besar untuk mengevaluasi enam *similarity measures* yang berbeda dalam konteks Orkut social network. Meskipun hasilnya mungkin bias untuk setiap *setting* tertentu dari eksperimen yang telah dilakukan, menarik untuk dicatat bahwa *cosine similarity* dapat memberikan respon terbaik dalam rekomendasi. Sedangkan penelitian yang serupa dilakukan oleh [15] menyimpulkan bahwa dalam kasus umum, ketepatan prediksi rekomendasi tidak dipengaruhi oleh *similarity measures* yang terpilih. Faktanya, menggunakan *similarity measures* secara acak kadang-kadang menghasilkan prediksi yang lebih baik dari pada menggunakan salah satu *similarity measure* yang terkenal.

1. *Sampling*

*Sampling* adalah teknik utama dalam *data mining* untuk memilih data yang relevan dari kumpulan data yang relatif besar. Teknik ini digunakan baik dalam *data preprocessing* maupun *final data interception*. Pengambilan sampel perlu digunakan karena komputasi akan sangat mahal jika kumpulan data yang ada diproses semuanya. *Training* dan *testing dataset* juga dapat dibuat dengan melakukan *sampling*. Biasanya *training dataset* digunakan untuk mempelajari parameter atau konfigurasi algoritma yang digunakan dalam tahap analisis, sedangkan *testing dataset* digunakan untuk mengevaluasi model yang diperoleh dalam tahap pelatihan. Dengan demikian dapat dipastikan kinerja dari model yang dibuat.

Teknik pengambilan sampel yang paling sederhana adalah *random sampling*, yaitu pemilihan *item* harus memiliki probabilitas yang sama. Namun teknik tersebut tidak dapat menjamin bahwa sampel yang diambil secara acak mampu merepresentasikan data secara tepat. Sehingga pendekatan yang lebih canggih dibutuhkan, Seperti *cross validation* (CV). Pendekatan ini membagi dataset menjadi dua subset, yaitu data *training* sebagai proses pembelajaran bagi model yang digunakan dan data *validation* sebagai proses validasi model [16]. Selanjutnya pemilihan jenis CV dapat didasarkan pada ukuran dataset. Biasanya *n-fold cross validation* (mengulangi proses CV sebanyak n kali) digunakan karena dapat mengurangi waktu komputasi dengan tetap menjaga keakuratan estimasi. Namun menurut [17], CV mungkin tidak dapat diandalkan kecuali dalam dataset yang jumlahnya cukup besar.

1. *Reducing Dimensionality*

Umumnya penambahan atribut pada suatu data akan membuat proses klasifikasi menjadi lebih mudah. Tetapi jika jumlah atribut terus bertambah, sedangkan jumlah sampel data terbatas, maka akurasi klasifikasi ini pada titik tertentu akan menurun. Fenomena inilah yang disebut dengan *“Curse of dimensionality”* [18]. Teknik *dimentionality reduction* sebagai solusi untuk fenomena tersebut dengan mengubah dimensi asli yang tinggi ke dimensi yang lebih rendah. Dalam RS, Sparsity dan *curse of dimensionality* adalah masalah yang selalu muncul. Bahkan dalam pengaturan yang paling sederhana, kita cenderung memiliki matriks yang jarang dengan ribuan baris dan kolom (yaitu pengguna dan item), sebagian besar adalah nol. Oleh karena itu, pengurangan dimensi pada RS telah menjadi langkah awal yang perlu dilakukan dalam mendesain RS. Algoritma yang paling relevan digunakan untuk mengurangi dimensi pada RS adalah *principal component analysis* (PCA) dan *singular value decomposition* (SVD) [13]. Kedua algoritma tersebut umumnya sering dipakai pada metode *collaborative filtering*. Goldberg dkk. mencoba menerapkan pendekatan PCA pada model *collaborative filtering* untuk merekomendasikan lelucon online [19]. Awalnya matriks yang digunakan adalah matriks penilaian pengguna terhadap *item*. Lalu mereka memilih subkumpulan item yang semua penggunanya memiliki peringkat. matriks baru tersebut kemudian digunakan untuk menghitung matriks korelasi global dengan menerapkan standar PCA 2 dimensi. Sistem mereka, yang dikenal sebagai Eigentaste 2, menghasilkan kualitas yang standar. Kemudian SVD digunakan untuk meningkatkan model *collaborative filtering*. penelitian yang dilakukan oleh [20], menerapkan SVD dengan dua cara yang berbeda untuk membangun RS. Pertama, SVD dapat digunakan untuk mengungkap hubungan laten antara pelanggan dan produk.

Untuk mencapai tujuan ini, mereka terlebih dahulu mengisi angka nol dalam matriks item-pengguna dengan rata-rata peringkat item dan kemudian menormalkan dengan mengurangkan rata-rata pengguna. Matriks ini kemudian difaktorkan menggunakan SVD dan dekomposisi yang dihasilkan bisa digunakan (setelah beberapa operasi sepele) langsung untuk menghitung prediksi. Kedua, menggunakan ruang dimensi rendah yang dihasilkan dari SVD untuk meningkatkan pembentukan ketetanggaan yang nantinya akan digunakan pada pendekatan kNN.

Seperti yang dijelaskan oleh [21], salah satu keuntungan besar SVD adalah algoritma inkremental untuk menghitung dekomposisi yang memungkinkan untuk menerima pengguna atau peringkat baru tanpa harus mengkomputasi ulang model yang telah dibangun dari data yang ada sebelumnya. Ide yang sama kemudian diperpanjang dan diformalkan oleh [22] menjadi model SVD online. Penerapan SVD incremental baru-baru ini menjadi pendekatan yang diterima oleh publik setelah keberhasilannya dalam Netflix Prize 3. Perlu dicatat bahwa varian yang berbeda dari Matrix Factorization (MF), seperti *Non-Negative Matrix Factorization* (NNMF) juga telah digunakan [23]. Algoritma ini, pada dasarnya, mirip dengan SVD. Ide dasarnya adalah untuk menguraikan matriks peringkat menjadi dua matriks, yang salah satunya mengandung fitur yang menggambarkan pengguna dan yang lainnya berisi fitur yang menjelaskan item. MF lebih baik daripada SVD dalam menangani nilai-nilai yang hilang dengan memperkenalkan istilah bias untuk model. Namun, SVD juga dapat menangani hal tersebut pada langkah preprocessing dengan mengganti nol dengan rata-rata rating pada *item*. Perhatikan bahwa keduanya SVD dan MF cenderung overfitting. Namun, ada varian MF, seperti Regularized

Kernel Matrix Factorization yang dapat menghindari masalah secara efisien. Masalah utama yang muncul pada MF dan SVD adalah terletak pada ketidakpraktisannya untuk menghitung ulang faktorisasi tersebut setiap kali matriks diperbarui karena kerumitan komputasi. Namun, [24] mengusulkan metode online yang memungkinkan untuk memperbarui *factorized approximation* tanpa menghitung ulang seluruh model.

1. Classification

klasifikasi adalah pemetaan antara ruang fitur dan ruang label, di mana fitur mewakili karakteristik dari elemen untuk mengklasifikasikan dan label mewakili kelas. Pada Sistem rekomendasi restoran, misalnya, dapat diimplementasikan oleh sebuah *classifier* yang mengelompokkan restoran menjadi satu dari dua kategori (baik, buruk) berdasarkan

sejumlah fitur yang menggambarkannya (misalnya, kualitas makanan dalam skala mulai 1 hingga 10, atmosfer dalam skala dari 1 hingga 10, dll.). Restoran R tertentu akan diwakili oleh vektor fitur FVR = <f v1, fv2, fvn>. Dalam contoh khusus ini, klasifikasi adalah biner karena hanya menghasilkan dua label: baik atau buruk [13]. Terdapat banyak sekali metode klasifikasi yang dikenalkan. Meskipun secara umum, metode tersebut dikelompokan menjadi dua, yaitu *supervised* dan *unsupervised*. Namun untuk jelasnya, perlu dibahas secara rinci metode-metode klasifikasi sebagai berikut:

1. *Nearest* *Neighbors*
2. *Decision Trees*
3. *Rule-based Classifier*
4. *Bayesian Classifier*
5. *Artificial Neural Network*
6. *Support Vector Machine*
7. *Ensembles of Classifier*
8. *Evaluating Classifier*
9. Cluster Analysis

*Clustering* [25], juga disebut sebagai *unsupervised learning* dengan mengelompokan item-item yang mirip kedalam satu cluster.

1. Rusult Interception

[1] F. Ricci, L. Rokach, and B. Shapira, “Introduction to recommender systems handbook,” in *Recommender systems handbook*, Springer, 2011, pp. 1–35.

[2] D. Jannach, P. Resnick, A. Tuzhilin, and M. Zanker, “Recommender systems—beyond matrix completion,” *Commun. ACM*, vol. 59, no. 11, pp. 94–102, 2016.

[3] E. V. Buskirk, “The Most Streamed Music from Spotify Discover Weekly,” 2016. [Online]. Available: https://insights.spotify.com/us/2016/07/07/top-music-discover-weekly/ . [Accessed: 03-Feb-2018].

[4] X. Amatriain and J. Basilico, “Netflix Recommendation Beyond 5 Stars (Part 1),” 2012. [Online]. Available: https://medium.com/netflix-techblog/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429. [Accessed: 20-Feb-2018].

[5] H. Ahn, K. Kim, and I. Han, “Mobile advertisement recommender system using collaborative filtering: MAR-CF,” in *Proceedings of the 2006 conference of the Korea society of management information systems*, 2006, pp. 709–715.

[6] E. A"\imeur, G. Brassard, J. M. Fernandez, and F. S. M. Onana, “A lambic: a privacy-preserving recommender system for electronic commerce,” *Int. J. Inf. Secur.*, vol. 7, no. 5, pp. 307–334, 2008.

[7] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.

[8] S. Berkovsky, T. Kuflik, and F. Ricci, “Mediation of user models for enhanced personalization in recommender systems,” *User Model. User-adapt. Interact.*, vol. 18, no. 3, pp. 245–286, 2008.

[9] J. Ben Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative filtering recommender systems,” in *The adaptive web*, Springer, 2007, pp. 291–324.

[10] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*. Cambridge university press, 2014.

[11] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[12] D. Pyle, “Data Preparation for Data Mining (The Morgan Kaufmann Series in Data Management Systems).” San Francisco: Morgan Kaufmann, 1999.

[13] X. Amatriain and J. M. Pujol, “Data mining methods for recommender systems,” in *Recommender systems handbook*, Springer, 2015, pp. 227–262.

[14] E. Spertus, M. Sahami, and O. Buyukkokten, “Evaluating similarity measures: a large-scale study in the orkut social network,” in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, pp. 678–684.

[15] N. Lathia, S. Hailes, and L. Capra, “The effect of correlation coefficients on communities of recommenders,” in *Proceedings of the 2008 ACM symposium on Applied computing*, 2008, pp. 2000–2005.

[16] A. Wibowo, “10 FOLD-CROSS VALIDATION,” 2017. [Online]. Available: https://mti.binus.ac.id/2017/11/24/10-fold-cross-validation/. [Accessed: 25-May-2018].

[17] A. Isaksson, M. Wallman, H. Göransson, and M. G. Gustafsson, “Cross-validation and bootstrapping are unreliable in small sample classification,” *Pattern Recognit. Lett.*, vol. 29, no. 14, pp. 1960–1965, 2008.

[18] A. Nugroho S, “Curse of dimensionality (Kutukan Dimensi),” 2007. [Online]. Available: https://asnugroho.wordpress.com/2007/12/31/curse-of-dimensionality-kutukan-dimensi-tinggi/. [Accessed: 25-May-2018].

[19] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, “Eigentaste: A constant time collaborative filtering algorithm,” *Inf. Retr. Boston.*, vol. 4, no. 2, pp. 133–151, 2001.

[20] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Application of dimensionality reduction in recommender system-a case study,” 2000.

[21] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Incremental singular value decomposition algorithms for highly scalable recommender systems,” in *Fifth International Conference on Computer and Information Science*, 2002, pp. 27–28.

[22] M. Brand, “Fast online svd revisions for lightweight recommender systems,” in *Proceedings of the 2003 SIAM International Conference on Data Mining*, 2003, pp. 37–46.

[23] M. Wu, “Collaborative filtering via ensembles of matrix factorizations,” in *Proceedings of KDD Cup and Workshop*, 2007, vol. 2007.

[24] S. Rendle and L. Schmidt-Thieme, “Online-updating regularized kernel matrix factorization models for large-scale recommender systems,” in *Proceedings of the 2008 ACM conference on Recommender systems*, 2008, pp. 251–258.

[25] J. A. Hartigan, “Clustering algorithms (probability & mathematical statistics).” John Wiley & Sons Inc New York, 1975.