

Implementasi dan Analisis *Online-Updating Regularization Kernel Matrix Factorization Model* pada Sistem Rekomendasi

Analysis and Implementation *Online – Updating Regularization Kernel Matrix Factorization Model* on Recommender System

Kadek Byan Prihandana Jati¹, Agung Toto Wibowo, ST., MT.², Rita Rismala, ST., MT.³

^{1,2,3}Prodi S1 Teknik Informatika, Fakultas Teknik, Universitas Telkom

byanjati@gmail.com, ² atwbox@gmail.com, ³ritaris@telkomuniversity@ac.id

Abstrak

Faktorisasi Matriks adalah salah satu metode yang digunakan pada Sistem Rekomendasi untuk membuat sebuah model prediksi rating. Salah satu jenisnya adalah *Regularized Matrix Factorization* yang mampu memberikan kualitas rekomendasi yang tinggi pada sebuah sistem rekomendasi. Akan tetapi, teknik - teknik Faktorisasi Matriks bermasalah jika model pada sistem rekomendasi berupa model yang statik. Permasalahan performansi terjadi, karena proses learning data pada Faktorisasi Matriks membutuhkan waktu yang lama. Model Online dari Faktorisasi Matriks merupakan hal yang dapat memperbaiki model sebelumnya, dengan model online, waktu yang dibutuhkan untuk melakukan proses prediksi untuk user dan item yang baru, lebih cepat dibandingkan dengan model offline faktorisasi matriks. Penelitian ini berfokus dalam menganalisis dan mengimplementasikan model online dari *Regularized Matrix Factorization* pada sebuah sistem rekomendasi. Hasil yang diperoleh adalah kualitas prediksi rating dengan metode *online – update RKMF* mengungguli kualitas prediksi rating dengan metode *full – retrain RKMF* dengan perbedaan nilai RMSE 2% pada kondisi terbaik, dan dengan waktu prediksi yang sangat singkat.

Kata kunci : *faktorisasi matriks, online updating, recommender performance*

Abstract

Matrix Factorization is one of many method in Recommendation System which used to create a prediction model. *Regularized Matrix Factorization* is the extended method that could give recommendation with high quality accuracy of model. But, these Matrix Factorization technique mostly have a performance problem on it's system. This problem is caused by the learning process at this method need a sum of time. Matrix Factorization with Online Model could help this problem to be vanished, with this model, the time which needed for learning could be reduced if compared with the Offline Model. This Research have a focus in analyzing and to implement the Online Model of *Regularized Matrix Factorization* on Recommender System. The result which obtained are *online – update RKMF* could beat the *full – retrain RKMF* in quality of rating prediction with difference of RMSE value is 2% at the best performance and with a short time prediction.

Keywords: *matrix factorization, online updating, recommender performance*

1. Pendahuluan

Berkembangnya teknologi internet pada zaman ini mengakibatkan tingginya kecepatan arus informasi dan banyaknya konten - konten di internet yang dimuat. Manusia sebagai pengguna internet, sering kali dipaksa untuk memilah informasi yang berguna dari tumpukan informasi yang berskala sangat besar. Diperlukan waktu yang lama untuk memilah informasi tersebut, sehingga diperlukan sistem untuk memberikan rekomendasi kepada pengguna. Sistem rekomendasi umumnya diimplementasikan pada website yang memiliki item dan pengguna, seperti E - commerce, website informasi film, website musik, website berita, dan website sejenis lainnya. Dan, website - website tersebut umumnya dipublikasikan secara online serta dapat melayani setiap pengguna secara real time.

Terdapat beberapa metode untuk sistem rekomendasi yang dinamis dalam menyediakan website online tersebut kepada pengguna, seperti metode Collaborative Filtering berbasis Co-Clustering dan CF berbasis Jointly Derived Neighborhood Interpolation Weights [10,11]. Metode Collaborative Filtering berbasis Co-Clustering dapat diimplementasikan pada Sistem rekomendasi online dan mampu memberikan rekomendasi dengan kualitas yang tinggi pada level komputasi yang rendah dibandingkan dengan metode korelasi dan SVD tradisional [9], akan tetapi metode ini belum dapat memberikan rekomendasi kepada pengguna dan item yang masuk ke dalam sistem dengan cepat. Sementara, CF berbasis Jointly Derived Neighborhood Interpolation menggunakan metode k-NN yang dapat menjadi metode terbaik untuk diaplikasikan ke dalam sistem rekomendasi yang berbasis pada item[3].

Faktorisasi Matriks merupakan salah satu metode yang tangguh untuk menghasilkan rekomendasi jika dibandingkan dengan metode k-NN klasik [2]. Pada Faktorisasi Matriks, data User dan Item dibentuk ke dalam sebuah bentuk matriks R yang kemudian mengalami fase training untuk menghasilkan sebuah rekomendasi. Permasalahan pada faktorisasi matrix adalah metode tersebut memiliki komputasi yang tinggi ketika melakukan learning, sehingga metode ini belum dapat diimplementasikan pada mode online.

Fokus pada penelitian ini adalah mengimplementasikan metode Regularized Kernel Matrix Factorization ke dalam sistem online dan menganalisis performansi serta kualitas prediksi yang dihasilkan.

2. Landasan Teori

2.1 Faktorisasi Matriks

Faktorisasi Matriks adalah metode yang paling sukses dari model laten faktor [2]. Faktorisasi Matriks mengkararakteristikan user dan item dengan vektor faktor yang disimpulkan dari item rating, sehingga user dan item dapat menghasilkan sebuah informasi yang mengindikasikan bahwa keduanya memiliki korespondensi yang tinggi dan saling mempengaruhi. Dari korespondensi yang kuat antara item dan user akan sangat berpengaruh ke dalam rekomendasi yang akan diberikan.

Ide pada Faktorisasi Matriks adalah faktor laten sejumlah k yang digunakan untuk menjelaskan bagaimana seorang user memberikan rating kepada sebuah item. Sebagai contohnya, dua orang user akan memberikan rating yang tinggi terhadap sejumlah film tertentu jika mereka berdua menyukai artis dari film tersebut atau genre film ada *action* , atau kesamaan lainnya. [2]

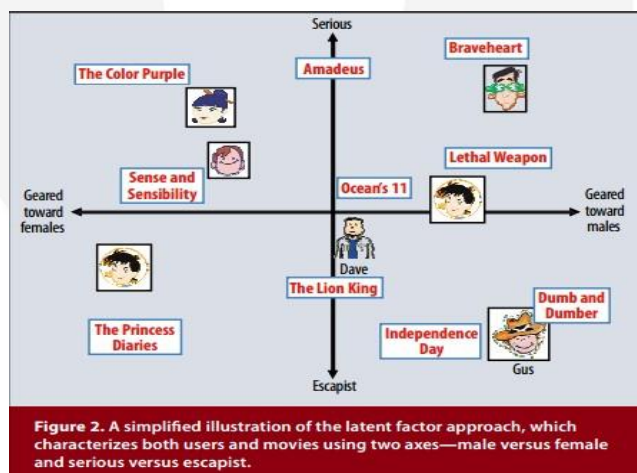


Figure 2. A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

Gambar 1 Pendekatan Model Faktor Laten

Dalam Faktorisasi Matriks, sebuah matriks akan didekomposisi menjadi 2 faktor W dan H sehingga berlaku persamaan berikut ini :

$$R = W \cdot H^T \tag{1}$$

Matriks R akan didekomposisi menjadi bentuk Matriks yang menggunakan fitur di dalamnya [9]. Dimana W sekarang merupakan matriks $W_u \times f$ dan H adalah matriks $f \times H_i$, f adalah jumlah dari fitur yang digunakan. Fitur

- fitur tersebut mendeskripsikan matriks user (W) dan matriks item (H). Dekomposisi Matriks tersebut disertai dengan Inisialisasi seluruh nilai elemen pada Faktor Matriks. Terdapat beberapa metode khusus untuk proses Inisialisasi, pada NMF (Non – Negative Matrix Factorization), proses Inisialisasi dilakukan berdasarkan formula berikut :

$$w_{ij} = h_{ij} + \sqrt{\frac{g - r_{max}}{r_{max} - r_{min}}} + noise \tag{2}$$

Keterangan :

w_{ij} dan h_{ij} adalah elemen Faktor Matriks User dan Faktor Matriks Item.

g adalah global average rating pada dataset training

$r_{max}; r_{min}$ adalah nilai maksimal rating dan nilai minimal rating pada dataset

$noise$ adalah nilai random uniform di sekitar 0

Pada proses Inisialisasi RKMF dengan Kernel Linear dan Logistic dilakukan dengan pemberian semua nilai elemen Faktor Matriks dengan nilai disekitar nilai 0. Inisialisasi tersebut dilakukan untuk membawa model yang akan dibentuk ke nilai Error Rate yang seminimal mungkin, sehingga proses Learning yang akan dilakukan dapat dipercepat.

Sistem Rekomendasi pada umumnya akan memberikan prediksi dengan formula sebagai berikut :

$$r_{ij} = a + c + K(w_i, h_j) \tag{3}$$

Formula tersebut merepresentasikan bahwa prediksi terdiri dari komponen a (bias), c (konstanta rescale), dan K(w_i, h_j) yang merupakan nilai interaksi dari user-item. [2]

Nilai Interaksi User-Item pada MF dinyatakan dalam perhitungan *dot product* dari setiap user dan item, Sebagai berikut [1]:

$$r_{ij}^* = \sum_{k=1}^n w_{ik} \cdot h_{kj} \tag{4}$$

Keterangan :

$$\begin{aligned} w_{ij} &= [w_{i1} \ w_{i2} \ \dots \ w_{in}] \\ h_{ij} &= [h_{1j} \ h_{2j} \ \dots \ h_{nj}] \\ h_{ij} &= [h_{1j} \ h_{2j} \ \dots \ h_{nj}] \end{aligned}$$

Error dari setiap rekomendasi rating inilah yang akan diminimalisir pada aktifitas learning Faktorisasi Matriks, sehingga diperoleh nilai Matriks R^* yang berkualitas. Perhitungan Error pada learning dinyatakan dengan sebagai berikut [1]:

$$E(r_{ij}^*) = E(r_{ij}) = \sum_{i,j} (r_{ij} - r_{ij}^*)^2 \tag{5}$$

Keterangan :

S adalah Jumlah keseluruhan rating yang dipantau

W adalah Matriks User

H adalah Matriks Item

r_{ij} adalah Nilai rating user-item

r_{ij}^* adalah Nilai prediksi rating user-item

Contoh Matriks Awal pada Faktorisasi Matriks, sebelum melalui learning :

Tabel 1 Contoh Matriks Awal

	D1	D2	D3	D4
U1	5	3	0	1
U2	4	0	0	1
U3	1	1	0	5
U4	1	0	0	4

Contoh Matriks Akhir pada Faktorisasi Matriks, setelah melalui proses learning :

Tabel 2 Matriks hasil learning

	D1	D2	D3	D4
U1	4.97	3	2.16	0.98
U2	3.98	2.4	1.8	0.80
U3	1.05	0.9	5.32	4.93
U4	1	0.9	4.55	3.94

2.2.1 Non - Negative Matrix Factorization

Metode Non – Negative Matrix Factorization mirip dengan metode Matrix Factorization, akan tetapi nilai semua elemen pada NMF ini memiliki sebuah batasan yaitu semua nilai tersebut tidak dapat bernilai kurang dari 0. Ini bertujuan untuk menghilangkan pengaruh interaksi elemen User dan Item yang bernilai negatif.

2.2 Gradient Descent

Algoritma Gradient Descent adalah algoritma yang digunakan untuk mengoptimisasi *Empirical Risk* [12]. Setiap iterasi pada proses Learning yang dilakukan, algoritma ini memperbarui *weights* w pada basis dari gradient

$$w_{i,j} + 1 = w_{i,j} - \gamma \sum_{k=1}^1 \nabla_w Q(w_{i,j}, r_{i,j}) \quad (6)$$

γ merupakan parameter gain yang mengatur proses pembaruan nilai weight. Proses memperbarui *weights* w ini akan mencapai pada sebuah titik konvergen yang disebut titik *linear convergence*. Pada titik tersebut, sebuah fungsi berada pada *Empirical Risk* dengan nilai terendah.

2.3 Teknik Kernel

Metode Kernel adalah metode yang terbukti sukses diimplementasikan pada sejumlah metode Learning (SVM, Gaussian Processes, Regularization Networks) [6]. Kernel Trick sering digunakan pada banyak kasus *learning* data, karena pada kasus - kasus tersebut tidak semua data dapat dipisahkan secara linier. Hal tersebut sering dinamakan non-linearly separable, dan metode Kernel dapat menjadi solusi terhadap permasalahan tersebut.

Fungsi kernel diperkenalkan pada Faktorisasi Matrix untuk mengintegrasikan informasi tambahan mengenai baris dan kolom matriks, yang sangat penting digunakan untuk membuat prediksi rating [4]. Beberapa metode kernel yang populer pada Kernel Matrix Factorization adalah sebagai berikut [1]:

$$K(x_i, h_i) = (x_i, h_i) \tag{8}$$

$$K(x_i, h_i) = (1 + \frac{(x_i, h_i)^d}{\|x_i - h_i\|^2}) \tag{9}$$

$$K(x_i, h_i) = \exp(-\frac{(x_i, h_i)}{2\sigma^2}) \tag{10}$$

$$K(x_i, h_i) = \exp(x_i + h_i) \tag{11}$$

Berdasarkan formula (2) setiap kernel memiliki pengaturan nilai parameter yang berbeda. Pada Kernel Linear, representasi nilai prediksi memiliki parameter a = bias, c = 1, dan $K(w_u, h_i) = w_u \cdot h_i$. Bias pada Kernel Linear direpresentasikan dengan formula berikut :

$$K(x_i) = \sum_{w \in S} w_i \tag{12}$$



Pada Kernel Logistik, representasi prediksi mirip dengan Kernel Linear, dengan perubahan di representasi bias menjadi seperti berikut :

$$r_{ij} = -\ln\left(\frac{h_{ij}}{1+h_{ij}}\right) \tag{13}$$

dengan cara adalah r_{ij} dan h_{ij} adalah dengan r_{ij} . Sedangkan pada metode NMF dengan Kernel Linear, Masing – masing jenis Kernel juga memiliki konfigurasi metode proses update pada Algoritma Gradient Descent yang digunakan untuk Learning. Dengan mengupdate setiap elemen matriks sehingga model aproksimasi yang dibentuk memiliki error yang paling minimum terhadap model User Item.

Secara umum, formula update yang akan digunakan semua Kernel dan NMF adalah sebagai berikut :

$$\frac{\partial L(\{r_{ij}\}, W, H)}{\partial w_{ij}} \propto (r_{ij} - h_{ij}) \cdot \frac{\partial}{\partial w_{ij}} (w_{ij}, h) + h_{ij} \tag{14}$$

$$\frac{\partial L(\{r_{ij}\}, W, H)}{\partial h_{ij}} \propto (r_{ij} - h_{ij}) \cdot \frac{\partial}{\partial h_{ij}} (w_{ij}, h) + w_{ij} \tag{15}$$

Persamaan tersebut akan diadaptasi oleh Kernel masing – masing, persamaan (12) berfungsi untuk mengubah nilai elemen matriks faktor terhadap user dan persamaan (13) berfungsi untuk mengubah nilai elemen matriks faktor terhadap item.

Akan tetapi, pada metode NMF, persamaan (12) dan (13) berubah menjadi persamaan seperti berikut :

$$w_{ij} = \max(0, w_{ij} - \frac{\partial}{\partial w_{ij}} L(\{r_{ij}\}, W, H)) \tag{16}$$

$$h_{ij} = \max(0, h_{ij} - \frac{\partial}{\partial h_{ij}} L(\{r_{ij}\}, W, H)) \tag{17}$$

Persamaan 12, 13, 15 dan 16 pada masing – masing kernel dijabarkan menjadi persamaan berikut :

- Kernel Linear

$$\frac{\partial}{\partial w_{ij}} L(r_{ij}, h_{ij}) = h_{ij}, \quad \frac{\partial}{\partial h_{ij}} L(r_{ij}, h_{ij}) = w_{ij} \tag{18}$$

- Kernel Logistik

$$\frac{\partial}{\partial w_{ij}} L(r_{ij}, h_{ij}) \propto h_{ij} \cdot \ln\left(\frac{h_{ij}}{1+h_{ij}}\right) + (1+h_{ij}) \tag{19}$$

2.4 Regularization

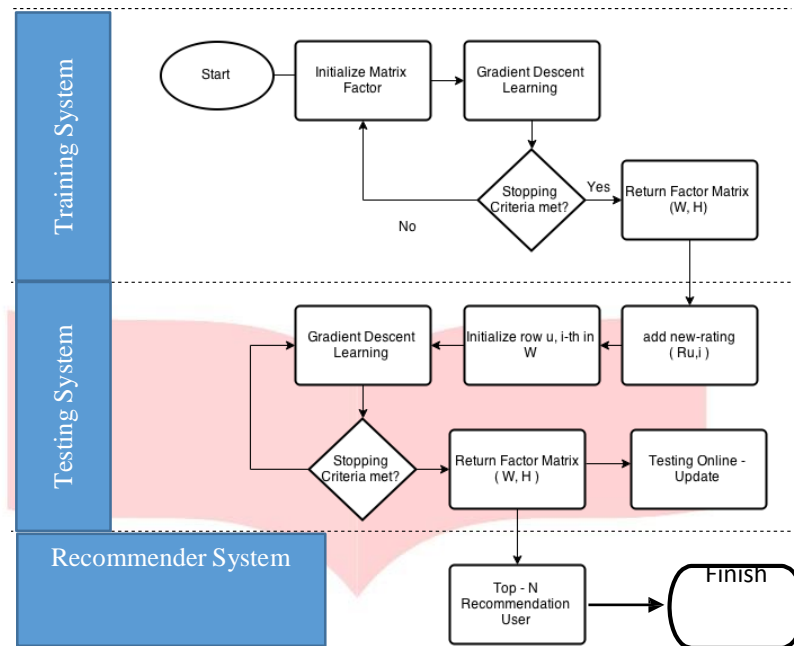
Kompleksitas data selalu berhubungan dengan besarnya jumlah dimensi fitur yang diobservasi, dan dari kompleksitas data tersebut seringkali diikuti dengan kompleksnya model yang dibuat untuk memprediksi data dengan sangat akurat. Pada dasarnya semakin kompleks model yang dibuat, semakin tidak stabil model tersebut. Regularization adalah sebuah metode untuk memodifikasi model yang kompleks dan memberikan model yang stabil pada model data yang kompleks [6].

2.5 Perancangan Sistem

Dalam penelitian ini dibuat sebuah sistem yang dapat memberikan rekomendasi rating dengan mengimplementasikan online – update RKMF. Sistem menerima masukan berupa dataset rating dan masukan rating oleh seorang user secara *real-time* dan menghasilkan sebuah rekomendasi top – n film berdasarkan prediksi rating sistem. Tahapan pertama ialah sistem melalui proses inialisasi dengan melakukan dekomposisi terhadap matriks yang dibentuk dari dataset. Semua nilai elemen faktor matriks hasil dekomposisi akan dibangkitkan secara acak dan mengikuti fungsi inialisasi setiap kernel. Tahapan kedua adalah melakukan proses training terhadap faktor matriks user dan faktor matriks item menggunakan dataset movielens 100k yang mempunyai 100.000 rating. Error model pada saat training akan dikurangi dalam setiap iterasi. Tahap ketiga adalah proses testing yang bertujuan untuk mengukur bagaimana performansi sistem online – update RKMF

dari segi kualitas dan *runtime* prediksi yang dihasilkan dan perbandingan performansi tersebut terhadap metode full – retrain RKMF. Pada tahap berikut, faktor matriks yang telah melewati proses learning akan digunakan untuk prediksi rating baru yang masuk ke dalam sistem online – update RKMF. Dengan skenario testing yang telah diimplementasi pada sistem testing, akan diukur kualitas prediksi sistem online – update RKMF yang berupa nilai RMSE dan *runtime* prediksi yang berupa ukuran waktu millisecond. Setelah dilakukan semua

tahapan tersebut, tahapan terakhir adalah memberikan n item rekomendasi kepada user yang telah memberikan rating film kepada sistem. Untuk semua rating yang telah diprediksi, diambil n item berupa data film yang akan disampaikan kepada user.



Gambar 2 Diagram Blok Sistem

3. Hasil Pengujian dan Kesimpulan

Tabel 3 menunjukkan performansi sistem online – update RKMF dari segi kualitas prediksi yang dihasilkan dan runtime pada saat memproses prediksi rating.

Tabel 3 Hasil Prediksi Struktur Sekunder RNA

Kasus	Kernel	Online – Update RKMF	Full – Retrain RKMF
new-user	Linear	0.928	0.935
	Logistik	0.929	0.950
	Linear+NMF	0.938	0.937
new-item	Linear	0.904	0.893
	Logistik	0.916	0.932
	Linear+NMF	0.90	0.89

Pengujian terhadap sistem online – update RKMF dilakukan dengan menggunakan kernel linear dan kernel logistic serta metode Non – Negative Matrix Factorization. Dari pengujian diperoleh bahwa untuk semua kernel dan metode NMF, sistem online – update RKMF menghasilkan prediksi yang memiliki akurasi melebihi dari standar akurasi pada kompetisi Netflix Prize berupa nilai RMSE yang kurang dari 0.95. Dan selain itu kualitas prediksi untuk masing – masing kernel hanya memiliki perbedaan dalam kasus yang paling buruk sebesar 2% jika dibandingkan dengan metode full – retrain RKMF. Perbedaan ini tidak berarti metode online – update RKMF tidak lebih akurat dibandingkan dengan metode full – retrain RKMF. Dapat dilihat bahwa nilai prediksi online – update RKMF pada kernel logistic lebih akurat 2% dibandingkan dengan metode full – retrain RKMF.

Pada kasus new – item, prediksi yang dihasilkan juga hampir sama dengan sistem pada kasus new – user. Semua Kernel yang diuji coba pada kasus new-user Item dapat menembus RMSE 0.95 pada akhir iterasi. Pada kasus new-item, RMSE pada mode Online Update dapat menyaingi nilai RMSE Full Retrain, dengan perbedaan nilai sebesar 1 – 1.6% pada semua kernel. Hal ini membuktikan bahwa Metode RKMF Online – Update ini merupakan metode umum yang dapat diimplementasikan pada 2 kasus yang simetris yaitu kasus new - user dan new- item serta dapat diimplementasikan pada kernel linear,logistik dan metode NMF.

Metode online – update RKMF dapat menandingi metode full – retrain RKMF pada simulasi testing tersebut disebabkan oleh proses inisialisasi sebelum simulasi testing dimulai, yaitu dengan melakukan full – retrain RKMF pada matriks dataset. Sehingga hasil faktor matriks dari full – retrain RKMF tersebut sudah dapat membentuk model prediksi rating yang optimal. Hasil faktor matriks dengan model prediksi yang optimal inilah yang digunakan sebagai acuan untuk membentuk prediksi berikutnya pada metode online – update RKMF. Semua prediksi oleh online – update RKMF tidak dimulai dari awal lagi, karena hal tersebut.

Daftar Pustaka:

- [1] S. Rendle and S. - T. Lars, "Online-updating regularized kernel matrix factorization models for large-scale recommender systems," in *Proceedings of the 2008 ACM conference on Recommender systems*, 2008.
- [2] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295)*, 2001.
- [3] Gonen, Mehmet and S. Kaski, "Kernelized Bayesian matrix factorization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, pp. 2047--2060, 2014.
- [4] Y. Koren, R. Bell and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, pp. 30--37, 2009.
- [5] J. Kivinen, A. J. Smola and R. C. Williamson, "Online learning with kernels," *Signal Processing, IEEE Transactions on*, vol. 52, pp. 2165-2176, 2004.
- [6] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting*, vol. 22, pp. 679-688, 2006.
- [7] T. Chai and R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?," *Geoscientific Model Development Discussions*, vol. 7, pp. 1525-1534, 2014.
- [8] Ott and Patrick, *Incremental matrix factorization for collaborative filtering*, Citeseer, 2008.
- [9] R. M. Bell and Y. Koren, "Scalable collaborative filtering with jointly derived neighborhood interpolation weights," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, 2007.
- [10] T. George and S. Merugu, "A scalable collaborative filtering framework based on co-clustering," in *Data Mining, Fifth IEEE International Conference on*, 2005.
- [11] L. Bottou, *Large-scale machine learning with stochastic gradient descent*, Springer, 2010, pp. 177-186.