ANALISA KINERJA METODE FAKTORISASI MATRIKS BERBASIS GRADIENT DESCENT PADA SISTEM REKOMENDASI

Endaryono, endaryono@unindra.ac.id
Program Studi Informatika, Universitas Indraprasta PGRI (Unindra),
Jl. Nangka No. 5B Jakarta Selatan 12530

Hendri Murfi, hendri@ui.ac.id
Program Studi Magister Matematika, Departemen Matematika, FMIPA UI,
Kampus UI Depok 16424

Abstrak

Metode faktorisasi matrik merupakan metode yang banyak digunakan pada sistem rekomendasi. Metode ini digunakan untuk mengekstrak variabel tersembunyi yang akan digunakan sebagai dimensi baru bagi vektor pengguna dan vektor produk. Representasi yang baru ini diharapkan akan memberikan kinerja yang lebih baik. Pada tulisan ini, kami melakukan studi eksperimen pada metode faktorisasi matriks berbasis gradient descent untuk sistem rekomendasi. Studi ini meliputi penentuan mekanisme dalam mendapatkan parameter optimal dari metode dan mengukur akurasi metode pada data riil. Dari eksperimen yang dilakukan, diperoleh parameter optimal untuk learning rate adalah 8×10^{-5} dan jumlah varibel tersembunyi adalah 3. Selanjutnya, nilai root mean square error (RMSE) pada kondisi optimal terebut adalah 0.9335. Hasil ini menunjukkan bahwa metode faktorisasi matriks berbasis gradient descent memberikan tingkat akurasi yang sangat baik pada sistem rekomendasi.

Kata Kunci: sistem rekomendasi, model varibel tersembunyi, faktorisasi matriks, gradient descent

Abstract

Matrix factorization method is a widely used method of recommendation system. This method is used to extract hidden variables that will be used as new dimensions for user vectors and product vectors. This new representation is expected to provide better performance. In this paper, we conducted an experimental study on the gradient descent-based matrix factorization method for the recommendation system. This study includes determining the mechanism in obtaining the optimal parameters of the method and measuring the accuracy of methods on real data. From the experiments conducted, the optimal parameter obtained for learning rate is 8×10^{-5} and the number of hidden variables is 3. Next, the root means square error (RMSE) value at the optimal condition is 0.9335. These results indicate that the gradient descent-based matrix factorization method provides an excellent level of accuracy in the recommendation system.

Keywords: recommendation system, hidden variables model, matrix factorization, gradient descent

1. PENDAHULUAN

Internet sebagai media bisnis memberikan kemudahan kepada masyarakat untuk dapat melakukan transaksi penjualan secara virtual. Penjualan secara *online* atau *e-commerce* adalah suatu alternatif yang sering dipilih kalangan bisnis dalam menjual produk, misal www.amazon.com, www.movielens.com. Untuk membantu konsumen dalam menentukan produk pilihannya, suatu sistem penjualan *online* biasanya dilengkapi oleh suatu sistem rekomendasi (*recommender system*).

Sistem rekomendasi adalah suatu sistem yang dapat menyaring atau memberikan informasi kepada pengguna berdasarkan kesamaan selera dari pengguna sebelumnya [1]. Secara umum, ada dua strategi yang digunakan oleh sistem rekomendasi, yaitu *content filtering* dan *collaborative filtering* [2].

Dalam *content filtering*, sistem akan mendefiniskan profil untuk masing-masing produk atau untuk masing-masing pengguna, Profil-profil ini memungkinkan sistem untuk mengasosiasikan seorang pengguna dengan produk-produk yang sesuai. Strategi berbasis konten ini membutuhkan informasi eksternal yang boleh jadi susah untuk diperoleh.

Sementara pada *collaborative filtering*, sistem bergantung hanya pada aktifitas pengguna sebelumnya, misal transaksi yang dilakukan atau pemberian *rating* pada suatu produk, tanpa perlu mendefinisikan profil secara eksplisit. Dengan kata lain, *collaborative filtering* adalah suatu teknik bagaimana membuat prediksi serta rekomendasi tentang minat seorang pengguna dengan pengumpulan informasi atau opini dari pengguna-pengguna lain [3].

Hubungan pengguna (*user-U*) dan produk (*item-I*) dalam *collaborative filtering* dapat disajikan dalam bentuk tripel (U, I, R) dan membentuk suatu matriks *rating* **R** [4]. Baris pada matriks *rating* **R** merupakan vektor pengguna dan kolom pada matriks *rating* **R** merupakan vektor produk. Selanjutnya dengan menggunakan jarak antar vektor dapat dilakukan perhitungan kemiripan antara pengguna dengan pengguna lainnya (*users similarity*) atau kemiripan antara produk satu dengan produk lainnya (*items similarity*). Kemiripan antar pengguna atau kemiripan antar produk inilah yang akan menjadi dasar perekomendasian dalam sistem rekomendasi.

Matriks *rating* **R** umumnya tidak lengkap, karena pengguna umumnya hanya memberikan *rating* pada sebagian produk saja. Sehingga, melakukan prediksi kecenderungan pengguna pada suatu produk berdasarkan kecenderungan pengguna yang mirip sering kali menjadi tidak akurat atau bahkan tidak bisa dilakukan. Salah satu cara yang sering dilakukan untuk mengatasi masalah ini adalah menggunakan model variable tersembunyi (*latent variable model*), di mana jumlah variabel tersembunyi ini biasanya jahu lebih kecil dari dimensi awal data pengguna atau produk.

Variabel tersembunyi ini akan digunakan sebagai dimensi baru dari vektor pengguna dan vektor produk. Selanjutnya, kecenderungan seorang pengguna pada suatu produk dapat dihitung berdasarkan kemiripan/jarak antara vektor pengguna dan vektor produk tersebut dalam representasi baru ini.

Dalam *machine learning*, salah satu metode yang sering digunakan untuk mengekstrak variabel tersembunyi ini adalah metode faktorisasi matriks. Karena tidak semua entri matriks *rating* **R** diketahui, maka penggunanaan metode faktorisasi matriks *singular value decomposition* (SVD) untuk mengekstrak variabel tersembunyi hanya bisa dilakukan dengan sejumlah modifikasi, misal mengganti entri yang tidak diketahui tersebut dengan nol. Akan tetapi, cara ini akan menyebabkan peningkatan jumlah data yang jika tidak dilakukan secara benar akan merusak data. Oleh karena itu diperlukan pendekatan lain sedemikian sehingga proses faktorisasi hanya dilakukan berdasarkan data yang diketahui saja.

Pendekatan yang umum digunakan saat ini adalah melakukan formulasi masalah faktorisasi matrik dalam bentuk masalah optimasi (*optimization*), yaitu diberikan suatu matriks *rating* $\mathbf{R} \in \mathrm{R}^{mxn}$ maka masalah faktorisasi matrik adalah mencari matriks $\mathbf{W} \in \mathrm{R}^{mxk}$ dan matriks $\mathbf{H} \in \mathrm{R}^{kxn}$ sedemikian sehingga $\|\mathbf{R} - \mathbf{WH}\|$ adalah minimum.

Salah satu metode optimasi yang dapat digunakan untuk menyelesaikan masalah faktorisasi matriks berdasarkan formulasi di atas adalah metode *gradient descent* [5,6]. Dalam metode ini, pencarian nilai minimum lokal dari fungsi objektif dilakukan dengan arah berlawanan atau negatif dari *gradient*. Pada makalah ini, kami akan melakukan studi secara eksperimen terhadap metode faktorisasi matriks berbasis *gradient descent* dalam mengekstrak variabel tersembunyi pada sistem rekomendasi.

Studi ini meliputi penentuan jumlah variabel tersembunyi (k) yang optimal dan simulasi akurasi dari metode dengan menggunakan data riil dari suatu sistem rekomendasi.

2. FAKTORISASI MATRIKS DENGAN METODE *GRADIENT DESCENT*

Sebagaimana telah disampaikan di atas, masalah dalam faktorisasi matriks dapat diformulasikan sebagai suatu masalah optimasi, yaitu diberikan suatu matriks *rating* $\mathbf{R} \in \mathrm{R}^{mxn}$, maka bagaimana mendapatkan matriks $\mathbf{W} \in \mathrm{R}^{mxk}$ dan matriks $\mathbf{H} \in \mathrm{R}^{kxn}$ dengan $k = (m,n)$ sedemikian sehingga hasil perkalian kedua matriks tersebut memiliki selisih yang sekecil mungkin bila dibandingkan dengan matriks $\mathbf{R} \in \mathrm{R}^{mxn}$ atau $\mathbf{WH} \approx \mathbf{R}$. Secara matematis, pernyataan ini dapat ditulis sebagai berikut:

$$\min_{W,H} f(\mathrm{W},\mathrm{H}) = \frac{1}{2}\|\mathrm{R} - \mathrm{W}\,\mathrm{H}\|^2 \qquad (1)$$

Misal vektor $\mathbf{w_i}$ adalah vektor yang entri-entrinya diambil dari baris ke-*i* matrik **W**, vektor $\mathbf{h_j}$ adalah vektor yang entri-entrinya diambil dari kolom ke-*j* matrik **H**, r_{ij} adalah entri matriks *rating* **R** pada baris ke-*i* dan kolom ke-*j*, dan R adalah himpunan pasangan (*i, j*), maka persamaan (1) dapat ditulis:

$$\min_{\mathbf{w_i},\mathbf{h_j}} f(\mathbf{w_i},\mathbf{h_j}) = \frac{1}{2}\sum_{i,j\in R}\left(r_{ij} - \mathbf{w_i}^{\mathrm{T}}\mathbf{h_j}\right)^2 \qquad (2)$$

Berdasarkan formulasi pada persamaan (2), maka memungkinkan kita melakukan faktorisasi matriks ketika data tidak lengkap atau tidak semua elemen dari matriks *rating* **R** diketahui. Dalam hal ini, kita melakukan faktorisasi matriks hanya berdasarkan data yang diketahui saja yang mana informasinya disimpan pada himpunan R.

Permasalah pada persamaan (2) merupakan permasalah optimasi tanpa kendala. Salah satu metode optimasi yang dapat digunakan untuk menyelesaikan masalah optimasi tanpa kendala tersebut adalah metode *gradient descent* [5,6].

Perhatikan fungsi objektif dari persamaan (2):

$$f(\mathbf{w_i},\mathbf{h_j}) = \frac{1}{2}\sum_{i,j\in R}\left(r_{ij} - \mathbf{w_i}^{\mathrm{T}}\mathbf{h_j}\right)^2$$

Selanjutnya, turunkan fungsi $f(\mathbf{w_i},\mathbf{h_j})$ terhadap vektor $\mathbf{w_i}$ sebagai berikut:

$$f(\mathbf{w_i},\mathbf{h_j}) = \frac{1}{2}\sum_{i,j\in R}\left(r_{ij} - \mathbf{w_i}^{\mathrm{T}}\mathbf{h_j}\right)^2$$

$$\frac{\partial f}{\partial \mathbf{w_i}} = \frac{1}{2}\sum_{i,j\in R}\frac{\partial}{\partial \mathbf{w_i}}\left(r_{ij} - \mathbf{w_i}^{\mathrm{T}}\mathbf{h_j}\right)^2$$

$$\frac{\partial f}{\partial \mathbf{w_i}} = -\sum_{i,j\in R}\left(r_{ij} - \mathbf{w_i}^{\mathrm{T}}\mathbf{h_j}\right)\mathbf{h_j}$$

$$\frac{\partial f}{\partial \mathbf{w_i}} = -\sum_{i,j\in R} e_{ij}\,\mathbf{h_j} \qquad (3)$$

Dengan cara yang sama turunan fungsi $f(\mathbf{w_i},\mathbf{h_j})$ terhadap vektor $\mathbf{h_j}$ adalah:

$$f(\mathbf{w_i},\mathbf{h_j}) = \frac{1}{2}\sum_{i,j\in R}\left(r_{ij} - \mathbf{w_i}^{\mathrm{T}}\mathbf{h_j}\right)^2$$

$$\frac{\partial f}{\partial \mathbf{h_j}} = \frac{1}{2}\sum_{i,j\in R}\frac{\partial}{\partial \mathbf{w_i}}\left(r_{ij} - \mathbf{w_i}^{\mathrm{T}}\mathbf{h_j}\right)^2$$

$$\frac{\partial f}{\partial \mathbf{h_j}} = -\sum_{i,j\in R}\left(r_{ij} - \mathbf{w_i}^{\mathrm{T}}\mathbf{h_j}\right)\mathbf{w_i}$$

$$\frac{\partial f}{\partial \mathbf{h_j}} = -\sum_{i,j\in R} e_{ij}\,\mathbf{w_i} \qquad (4)$$

Sebagaimana telah dijelaskan sebelumnya, g*radient descent* adalah metode optimasi untuk menemukan minimum lokal dari fungsi dengan cara melakukan langkah secara berulang-ulang dengan arah pencarian berlawanan atau negatif dari *gradient*. Sehingga, aturan pembaruan (*update rule*) dari $\mathbf{w_i}$ dan $\mathbf{h_j}$ pada metode *gradient descent* adalah sebagai berikut:

$$\mathbf{w_i}^{t+1} = \mathbf{w_i}^{t} + \eta\sum_{i,j\in R} e_{ij}\,\mathbf{h_j} \qquad (5)$$

dengan t = 0, 1, 2, 3, …, maxIterasi

$$\mathbf{h}_j^{t+1} = \mathbf{h}_j^t + \eta \sum_{i,j \in R} e_{ij} \mathbf{w}_i \qquad (6)$$

dengan t = 0, 1, 2, 3, …, MaxIterasi
dengan η adalah *learning rate*.

Setiap kali dilakukan proses *update* pada vektor $\mathbf{w_i}$ dan vektor $\mathbf{h_j}$ maka nilai *root mean square error* (RMSE) terus menurun dan akan berhenti jika telah mencapai nilai yang cukup terkecil.

Algoritma faktorisasi matriks dengan *gradient descent* adalah sebagai berikut:

(A). INPUT
R : data set,
k : jumlah fitur,
η : *learning rate*

(B). PROSES
1. Lakukan inisialisasi **W** dan **H** dengan perintah:
W = rand(M,*k*)
H = rand(*k*, N)

Lakukan langkah no. 2 sampai langkah no. 3 sampai maksimum iterasi.
---
2. Tentukan nilai *error*
$e_{ij} = r_{ij} - \mathbf{w_i}^T \mathbf{h_j}$

3. Lakukan *update* pada setiap $\mathbf{w_i}$ dan $\mathbf{h_j}$
$$\mathbf{w}_i^{t+1} = \mathbf{w}_i^t + \eta \sum_{i,j \in R} e_{ij} \mathbf{h_j}$$
---
4. Tentukan nilai *root mean square error* (*RMSE*)

$$RMSE = \sqrt{\frac{1}{|R|} \sum_{(i,j,r) \in R}^{n} (\mathbf{w_i}^T \mathbf{h_j} - r_{ij})^2} \qquad [7]$$

3. HASIL DAN PEMBAHASAN

Pada simulasi ini, kami menggunakan data dari situs *www.moviellens.org* yang dikelola oleh *GroupLens Research, University of Minnesota*. Deskripsi data tersebut dapat dilihat pada Tabel 1.

Tabel 1. Deskripsi data penelitian

Sumber Data	:	www.movielens.org
Jumlah Pengguna (N)	:	943 orang
Jumlah Produk (M)	:	1.682 judul film
Jumlah Data Rating	:	100.000
Jumlah Data Training	:	80.000 (80% data)
Jumlah Data Validasi	:	20.000 (20% data)
Ukuran Matriks		943 x 1682
Jumlah entri matriks	:	1.586.126
Prosentasi data rating	:	6.3%
Sparsity	:	93.7%

Desain eksperiment yang digunakan pada studi ini adalah *5-fold cross validation*, data dibagi dalam lima partisi yaitu S1, S2, S3, S4 dan S5. Masing-masing partisi ini saling lepas (*disjoint*). Selanjutnya, 80% data digunakan data *training* dan 20% data sebagai data *testing*

Secara umum, tahapan eksperimen yang dilakukan dijelaskan sebagai berikut:

Data awal : nama file u.data
data dibagi dalam lima partisi yaitu S1, S2, S3, S4 dan S5

Dikelompokkan dalam lima *fold*, yaitu:

Fold 1
Data *training* (Nama File: u1.base, Partisi : S1, S2, S3 dan S4)
Data *testing* (Nama File: u1.test, Partisi : S5)

Fold 2
Data *training* (Nama File: u2.base, Partisi : S1, S2, S3 dan S5)
Data *testing* (Nama File: u2.test, Partisi : S4)

Fold 3
Data *training* (Nama File: u3.base, Partisi : S1, S2, S4 dan S5)
Data *testing* (Nama File: u3.test, Partisi : S3)

Fold 4
Data *training* (Nama File: u2.base, Partisi : S1, S3, S4 dan S5)
Data *testing* (Nama File: u2.test, Partisi : S2)

Fold 5
Data *training* (Nama File: u2.base, Partisi : S2, S3, S4 dan S5)
Data *testing* (Nama File: u2.test, Partisi : S1)

Langkah eksperimen adalah sebagai berikut:

Langkah-1
Menentukan nilai *learning rate* η optimal untuk suatu nilai k. Nilai *learning rate* η yang didapat selanjutnya akan digunakan dalam semua simulasi menentukan nilai k yang optimum pada *fold-1* sampai dengan *fold-5*

Langkah-2
Melakukan simulasi menentukan nilai nilai RMSE (*root mean square error*) optimum pada k =1 sampai dengan k = 20 dalam *fold*-1 dengan nilai l *learning rate* η tetap.

Langkah-3
Melakukan simulasi menentukan nilai RMSE optimum pada k =1 sampai dengan k = 20 dalam *fold*-2 dengan nilai l *learning rate* η tetap

Langkah-4
Melakukan simulasi menentukan nilai RMSE optimum pada k =1 sampai dengan k = 20 dalam *fold*-3 dengan nilai l *learning rate* η tetap.

Langkah-5
Melakukan simulasi menentukan nilai RMSE optimum pada k =1 sampai dengan k = 20 dalam *fold*-4 dengan nilai l *learning rate* η tetap
Langkah-6
Melakukan simulasi menentukan nilai RMSE optimum pada k =1 sampai dengan k = 20 dalam *fold*-5 dengan nilai l *learning rate* η tetap.

Langkah-7
Dari lima *fold* tersebut cari k model yaitu nilai k yang memberikan rata-rata RMSE terkecil.

3.1 Menentukan Nilai *Learning Rate η*

Tahap awal, dilakukan simulasi untuk mencari nilai *learning rate* η untuk suatu nilai k. Nilai *learning rate* yang dipilih adalah nilai *learning rate* η yang memberikan RMSE yang minimum dan iterasi yang paling kecil. Hasil dari simulasi penentuan nilai *learning rate* η dapat dilihat pada Tabel 2.

Tabel 2. Nilai RMSE pada beberapa learning rate η

Learning rate η	Nilai RMSE pada data *testing*				
	Fold-1	*Fold*-2	*Fold*-3	*Fold*-4	*Fold*-5
3×10^{-5}	0,9676	0,9501	0,9391	0,9324	0,9390
4×10^{-5}	0,9676	0,9501	0,9391	0,9324	0,9390
5×10^{-5}	0,9676	0,9501	0,9391	0,9324	0,9390
6×10^{-5}	0,9676	0,9501	0,9391	0,9324	0,9390
7×10^{-5}	0,9676	0,9501	0,9391	0,9324	0,9390
8×10^{-5}	0,9676	0,9501	0,9391	0,9324	0,9390
9×10^{-5}	0,9676	0,9502	0,9391	0,9324	0,9391
10×10^{-5}	0,9676	0,9502	0,9391	0,9324	0,9391
11×10^{-5}	0,9677	0,9502	0,9391	0,9325	0,9391
12×10^{-5}	0,9677	0,9502	0,9392	0,9325	0,9391

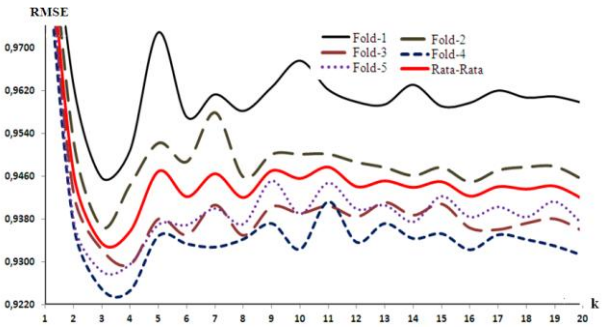
Dari hasil simulasi tersebut di atas, terlihat bahwa nilai *learning rate η* kurang dari dari 8×10^{-5} sudah tidak lagi mempengaruhi nilai RMSE. Oleh sebab itu nilai *learning rate η* yang digunakan dapat sembarang di bawah nilai 8×10^{-5}. Untuk simulasi ini nilai yang digunakan adalah 8×10^{-5}.

3.2 Menentukan Nilai k Optimum

Nilai *learning rate η* yang telah didapatkan, yaitu 8×10^{-10} selanjutnya digunakan untuk melakukan simulasi dalam menentukan nilai k optimum, yaitu nilai k yang memberikan nilai *RMSE* terkecil. Nilai RMSE untuk data *testing* pada masing-masing *fold* untuk beberapa nilai k dapat dilihat pada Tabel 5 dan Gambar 1. Dari Tabel 5 dan Gambar 1 ini dapat dilihat bahwa rata-rata RMSE terkecil terjadi pada nilai k = 3 dengan nilai rata-rata RMSE 0,9335. Dengan demikian parameter nilai k optimal untuk model 3. Dengan kata lain, model akan menggunakan 3 varibel tersembunyi tersebut untuk merepresentasikan pengguna dan produk. Selanjutnya, kecenderungan seorang pengguna pada suatu produk dihitung berdasarkan kemiripan atau jarak antara pengguna dan produk tersebut pada representasi baru ini, yaitu pada ruang tiga dimensi dengan dimensi berupa masing-masing variable tersembunyi yang berhasil diekstrak tersebut.

Tabel 3. Nilai RMSE data testing untuk berbagi nilai k pada masing-masing fold

Nilai k	RMSE pada data testing					Rata2 RMSE
	Fold-1	*Fold*-2	*Fold*-3	*Fold*-4	*Fold*-5	
1	1,0098	1,0049	0,9936	0,9904	0,9969	0,9991
2	0,9632	0,9524	0,9427	0,9367	0,9376	0,9465
3	0,9457	0,9364	0,9322	0,9249	0,9282	0,9335
4	0,9508	0,9444	0,9295	0,9247	0,9296	0,9358
5	0,9729	0,9522	0,9380	0,9349	0,9370	0,9470
6	0,9571	0,9487	0,9350	0,9334	0,9369	0,9422
7	0,9613	0,9579	0,9406	0,9328	0,9400	0,9465
8	0,9582	0,9458	0,9349	0,9343	0,9371	0,9421
9	0,9626	0,9501	0,9403	0,9372	0,9452	0,9471
10	0,9676	0,9501	0,9391	0,9324	0,9390	0,9456
11	0,9622	0,9501	0,9404	0,9413	0,9448	0,9478
12	0,9599	0,9486	0,9384	0,9337	0,9399	0,9441
13	0,9594	0,9476	0,9411	0,9372	0,9406	0,9452
14	0,9631	0,9461	0,9387	0,9344	0,9375	0,9440
15	0,9591	0,9476	0,9408	0,9353	0,9423	0,9450
16	0,9597	0,9449	0,9363	0,9323	0,9384	0,9423
17	0,9620	0,9471	0,9360	0,9351	0,9403	0,9441
18	0,9607	0,9477	0,9372	0,9342	0,9384	0,9436
19	0,9609	0,9478	0,9380	0,9330	0,9413	0,9442
20	0,9597	0,9453	0,9357	0,9312	0,9369	0,9418



Gambar 1. Nilai RMSE untuk berbagi nilai k pada masing-masing fold

Nilai akhir dari semua parameter optimal tersebut disimpan dalam *database*. Matriks **W** dan matriks **H** optimal yang telah didapat, selanjutnya dapat digunakan untuk memprediksi nilai dari matriks **R**, misal kita sebut $\hat{\mathbf{R}}$. Matriks prediksi $\hat{\mathbf{R}}$ ini memprediksi semua entri ke (*i*,*j*) dari matriks **R** termasuk entri-entri yang sudah memiliki *rating*. Karena model hanya fokus mencari entri-entri pada **R** yang belum mendapat *rating* saja maka dilakukan penyelesaian dengan membuat matrik indikator $\mathbf{B} \in R^{mxn}$ dari matriks *rating* R dengan definisi:

$$b_{ij} = \begin{cases} 1, & jika\, r_{ij} = 0 \\ 0 & jika\, r_{ij} \neq 0 \end{cases} \qquad (7)$$

Dengan adanya matrik indikator $\mathbf{B} \in R^{mxn}$ maka matriks *rating* **R** dapat diperbaharui menjadi $\mathbf{R}^{update}$ yaitu matriks dengan elemen-elemen dari entri-entri (i, j) anggota **R** yang sudah mendapat *rating* dan entri-entri (i, j) anggota matriks prediksi $\hat{\mathbf{R}}$ jika entri-entri (i, j) anggota **R** belum mendapat *rating*. Untuk mendapatkan $\mathbf{R}^{update}$ dapat digunakan persamaan berikut:

$$\mathbf{R}^{update} = \mathbf{R} + \mathbf{B}e\ \hat{\mathbf{R}} \quad [8] \qquad (8)$$

4. KESIMPULAN

Berdasarkan hasil studi eksperimental metode faktorisasi matriks berbasis *gradient descent* pada data MovieLens (www.movielens.org) didapat hasil sebagai berikut: nilai *learning rate* η optimal adalah 8 x 10^{-5}. Selanjutnya, dengan menggunakan nilai *learning rate* ini, jumlah variable tersembunyi k optimal yang diperoleh adalah 3. Dengan menggunakan nilai parameter ini, maka nilai RMSE yang dihasilkan adalah 0,9335. Hasil ini menunjukan bahwa metode faktorisasi berbasis *gradient descent* dapat berkerja dengan sangat baik pada data MovieLens.

DAFTAR ACUAN

G. Karypis. *Evaluation of Item Based Top-N Recommendation Algorithms*. Proceedings of the tenth international conference on Information and knowledge management, 2001.

M. A. Hao. *Learning to Recommend*, PhD Thesis, The Chinese University of Hongkong, 2009

B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. *Item-based Collaborative Filtering Recommender System Algorithms.* Proceeding of 10th International Conference on World Wide Web, 2001

I. Pil´aszy. *Factorization-Based large Scale Recommendation Algorithms*, PhD Thesis, Budapest University of Technology and Economics, 2009.

J. Nocedal dan S. J. Wright, *Numerical Optimization*, Springer, 1999.

S. Funk, *Netflix Update: Try This at Home*, 2006 (http://sifter.org/~simon/journal/20061211.html)

S. Xiaoyuan dan Khoshgoftaar. *A survey of Collaborative Filtering Techniques.* Advanced in Artificial Intelligence, Hindawi Publishing Corporation, 2009.

J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Economics*. John Wiley & Sons Ltd, 1999.