Penggunaan *Correlation-Based Similarity* untuk Sistem Rekomendasi Tanpa *Rating*

Hans Yulian[#1], Inge Martina[#2]

[#]Departemen Teknik Informatika, *Institut Teknologi Harapan Bangsa,*
Jl. Dipatiukur no. 80-84 Bandung, Indonesia
[1]hansyulian@windowslive.com
[2]inge@ithb.ac.id

***Abstrak*— Sistem rekomendasi adalah sebuah sistem yang menjadi sebuah kebutuhan banyak perusahaan saat ini, terutama perusahaan yang menjual produk dan melakukan aktivitasnya melalui media web. *Rating* yang telah diberikan oleh seorang pengguna akan digunakan sebagai referensi untuk menentukan rekomendasi untuk pengguna tersebut serta orang lain yang memiliki karakteristik yang mirip. Terkadang sistem rekomendasi tidak menggunakan *rating*, tetapi berdasarkan sejarah pembelian/pemakaian. *Correlation-based similarity* adalah sebuah algoritma yang dapat digunakan untuk mendapatkan nilai kemiripan antar dua obyek yang berbeda. Kemiripan dihitung berdasar *rating* yang diberikan oleh pengguna. Seorang pengguna dikatakan mirip dengan pengguna lain berdasarkan nilai *threshold* yang ditentukan. Kadang sulit untuk mendapatkan *rating* maka diperlukan sistem rekomendasi tanpa *rating*. Rumus *correlation-based similarity* perlu dioptimasi untuk sistem rekomendasi tanpa *rating*.**

Kata Kunci*— Sistem rekomendasi, *rating*, *correlation-based similarity*, *threshold

***Abstract*— *Recommender system is needed by many companies, mainly for companies that selling products and services in web. User ratings are used as recommendation references for user himself or for another users having similar characteristics. Sometimes recommender system doesn't use rating, but based on buying/using history. Correlation-based similarity is an algorithm that can be used for getting similarity value between two different objects. Similarity is calculated based on user ratings. Sometimes it is difficult to get rating, so that it is needed a recommender system without user input. Correlation-based similarity formula must be optimized for recommender system without rating*.**

Keywords*— *Recommender system, rating, correlation-based similarity, threshold

I. PENDAHULUAN

Pertambahan pesat dari produk-produk yang ditawarkan oleh sebuah perusahaan mempersulit untuk menemukan produk-produk yang disenangi pelanggannya Manusia memiliki keterbatasan kecepatan dalam memproses informasi jika dibandingkan dengan komputer. Oleh karena itu, lahirlah sebuah ide untuk bisa menggunakan kemampuan komputer yang mampu memproses informasi dengan cepat untuk bisa membantu menemukan produk dan informasi yang kemungkinan sesuai dengan selera seseorang. Program komputer seperti itu disebut sistem rekomendasi. Pengertian sistem rekomendasi menurut [1] adalah "*Recommender Systems are software tools and techniques providing suggestions for items to be of use to a user*". Sistem rekomendasi sudah banyak digunakan di *website* yang menjual produk atau layanan dalam jumlah banyak namun setiap produk atau layanan tersebut memiliki kemiripan-kemiripan dengan produk lain, contohnya adalah film, berita, dan musik. Kemiripan tersebut dapat dijadikan pembanding untuk menilai kemungkinan seseorang menyukai produk atau informasi yang baru.

Sistem rekomendasi *correlation-based similarity* dapat digunakan bersama-sama dengan metoda *thresholding* untuk menyaring rekomendasi. Nilai *thresholding* yang diambil adalah 0,75 [2].

II. *CORRELATION-BASED SIMILARITY*

Correlation-based similarity adalah sebuah teori yang dapat digunakan untuk menentukan bobot kemiripan antara 2 obyek yang sedang dibandingkan dengan menghitung jarak kosinus dari vektor x_u dan x_v. Obyek yang dimaksud dapat berupa pengguna atau *item*. Dalam kasus ini, kemiripan yang dicari dengan *correlation-based similarity* adalah pengguna. Pengguna akan memberikan *rating* pada *item-item*. *Rating* dapat berupa bagus, sedang, buruk atau suka, biasa saja, tidak suka, dan sejenisnya. Kemiripan pengguna-pengguna dapat dihitung dengan menggunakan rumus berikut:

$$CV(u,v) = \cos(x_w, x_v) = \frac{\sum i \in I_{uv} r_{ui} r_{vi}}{\sqrt{\sum i \in I_u r^2_{ui} \sum j \in I_v r^2_{uj}}} \quad (1)$$

Keterangan:
x : vektor dari obyek yang di-*rating* oleh seorang pengguna
I : obyek yang sudah di-*rating*
r : *rating*

Vektor x_u adalah obyek-obyek yang sudah di-*rating* oleh pengguna u dan vektor x_v adalah obyek-obyek yang sudah di*rating* oleh pengguna v. I_u adalah obyek yang sudah di-*rating* oleh pengguna u dan I_v adalah obyek yang sudah di-*rating* oleh pengguna v. I_{uv} adalah obyek yang sudah di-*rating* oleh pengguna u dan pengguna v.

III. Optimasi *Correlation-based* Tanpa *Rating*

Sistem rekomendasi mungkin tidak menggunakan *rating* karena sistem *rating* memerlukan peran serta pengguna untuk mengisinya yang kadang-kadang tidak disukai oleh pengguna. Tanpa masukan dari pengguna, sistem rekomendasi dapat tetap berjalan dengan cara hanya mengambil data berupa ada atau tidak. Contoh kasus adalah apakah *item* tertentu pernah dibeli seseorang atau tidak. Rumus *correlation-based similarity* pada persamaan (1) perlu diubah untuk memenuhi kebutuhan ini. Karena tidak adanya sistem *rating*, maka nilai *rating* dalam konteks ini akan diganti dengan status "sudah" sebagai nilai 1 dan "belum" sebagai nilai 0. Dalam hal ini, "sudah" dapat diinterpretasikan sebagai "sudah membeli", "sudah membaca", "sudah mengunjungi", dan lain-lain. Dengan demikian, beberapa komponen pada persamaan (1) dapat diubah sebagai berikut:

Komponen awal:	Komponen akhir:
$\sum_{i \in I_{uv}} r_{ui} r_{vi}$	nI_{uv}
$\sqrt{\sum_{i \in I_u} r^2_{ui} \sum_{i \in I_v} r^2_{uj}}$	$\sqrt{nI_u \, nI_v}$

Dengan *n* berupa banyaknya elemen. Perubahan ini dihasilkan karena *rating* yang akan dihasilkan hanyalah antara 1 dan 0, sehingga penggunaan kuadrat r^2_{ui} dan r^2_{vj} dapat diganti. $r_{ui}r_{vj}$ tidak diperlukan lagi, sehingga hanya akan bernilai 1 jika keduanya bernilai 1, sehingga dapat diartikan sebagai "jumlah elemen yang sama pada *u* dan *v*". Rumus (1) akan menjadi:

$$CV(u,v) = \cos(x_u, x_v) = \frac{nI_{uv}}{\sqrt{nI_u \, nI_v}} \qquad (2)$$

Keterangan:
n : banyaknya elemen
I : obyek yang sudah di-*rating*

Tentunya dalam upaya pemrograman, hal tersebut membuat program berjalan lebih cepat dan efisien.

IV. Studi Kasus Pemilihan Rekomendasi

Penelitian terhadap sifat dari penyebut rumus (2) mendapatkan hasil sebagai berikut:

i. Nilai hanya akan berkisar antara 0 sampai 1
ii. Nilai 1 hanya akan didapat jika $I_u = I_v$
iii. Semakin banyak obyek yang berstatus "sudah" oleh salah satu pihak akan menurunkan hasil perhitungan secara drastis.

Pernyataan pertama memang akan dapat selalu diterima dan pasti terjadi. Namun untuk pernyataan kedua, kurang dapat disetujui bahwa nilai 1 hanya akan didapat jika $I_u = I_v$ karena berarti nilai tersebut hanya bisa didapat jika pengguna *u* dan pengguna *v* benar-benar memiliki kesukaan yang sama tanpa beda sedikitpun. Padahal setiap manusia berbeda dan tidak ada satupun orang yang sama. Keadaan tersebut akan menurunkan hasil dari perhitungan rekomendasi karena bobot yang diberikan akan berkurang.

Tabel I adalah contoh kasus pembacaan berita oleh beberapa pengguna dengan status "sudah dibaca" dan "belum dibaca".

Jika dilakukan penghitungan dengan menggunakan rumus di atas mengenai bobot kemiripan Alpha dan Beta, maka akan didapatkan hasil $3/15^{1/2} = 0.774567$. Padahal jika diperhatikan, Alpha sudah membaca 3 berita yang sudah dibaca Beta, sedangkan Beta sudah membaca 5 berita dengan keadaan 3 berita sama dengan yang sudah dibaca oleh Alpha. Namun 2 berita belum ia baca dan belum dibaca oleh Alpha. Seharusnya kepada Alpha akan direkomendasikan berita 3 dan berita 5 karena ia memiliki bobot kemiripan dengan Beta sebesar 0.774567.

Hal ini masih belum menjadi masalah karena nilai tersebut masih berada di atas nilai *threshold*, yaitu 0.75 dan membuat Beta menjadi tetangga dari Alpha. Berikut contoh kasus serupa. Tabel II memberikan kasus yang sedikit berbeda dengan kasus pada Tabel I, yaitu dengan menambahkan sebuah berita.

Untuk kasus pada Tabel II di atas, maka hasil penghitungannya adalah $3/18^{1/2} = 0.7071$, dimana nilai tersebut akan membuat Beta tereliminasi dari tetangga Alpha (dengan *threshold* 0.75). Padahal jika dilihat, seharusnya Beta memang tetangga Alpha karena semua berita yang dibaca Alpha dibaca oleh Beta. Berita yang sudah dibaca oleh Beta seharusnya direkomendasikan untuk Alpha. Jadi, berdasarkan contoh kasus pada Tabel II ada yang perlu diubah dari rumus yang sudah diturunkan tersebut. Kasus seperti contoh di atas sangat mungkin terjadi dan seharusnya perhitungan yang dilakukan tetap dapat merekomendasikan apa yang sudah dibaca oleh Beta namun belum dibaca oleh Alpha.

Dengan demikian, diambil tindakan pengubahan terhadap rumus dengan mengganti komponen $\sqrt{nI_u \, nI_v}$ menjadi $\min(nI_u, nI_v)$ sehingga rumus (2) akan menjadi:

$$CV(u,v) = \cos(x_u, x_v) = \frac{nI_{uv}}{\min(nI_u, nI_v)} \qquad (3)$$

Alasan-alasan diambilnya keputusan tersebut adalah:

i. Keputusan tersebut dapat menangani kasus yang disebutkan di atas agar rekomendasi menjadi lebih akurat.

Tabel 1
Contoh Kasus 1

Pengguna	Berita 1	Berita 2	Berita 3	Berita 4	Berita 5	Berita 6	Berita 7
Alpha	✓	✓		✓			
Beta	✓	✓	✓	✓	✓		
Charlie						✓	✓

Tabel 2
Contoh Kasus 2

Pengguna	Berita 1	Berita 2	Berita 3	Berita 4	Berita 5	Berita 6	Berita 7	Berita 8
Alpha	✓	✓		✓				
Beta	✓	✓	✓	✓	✓			✓
Charlie						✓	✓	

ii. Perubahan dari sistem *rating* ke sistem histori (sudah atau belum) memerlukan penyesuaian ulang untuk algoritma penghitungannya.

Dengan demikian, perhitungan pada 2 contoh di atas memberikan nilai bobot ketetanggaan Alpha dan Beta sebesar 1.0. Tabel III memberikan contoh kasus berbeda.

Contoh pada Tabel III memberikan nilai bobot ketetanggaan Alpha dan Beta sebesar 3/min(4,6) = 3/4 = 0,75. Seiring dengan keputusan untuk menggunakan rumus tersebut, maka ditemukan contoh kasus lain yang mungkin terjadi.

Untuk kasus pada Tabel IV, jika dihitung menggunakan cara sebelumnya, hasilnya adalah 1/81/2 = 0,3535 yang menyebabkan Alpha dan Beta tidak terseleksi untuk ketetanggaan, sedangkan untuk perhitungan yang telah diganti menggunakan analisis penelitian ini menghasilkan nilai 1/min(1,8) = 1/1 = 1,0 yang menyebabkan Alpha dan Beta memiliki ketetanggaan yang sempurna dan semua berita dari Beta yang belum dibaca, Alpha direkomendasikan kepada Alpha. Untuk kasus ini, dapat dilihat ada 2 keadaan yang sama-sama tidak baik, yaitu:

i. Alpha tidak mendapatkan rekomendasi sama sekali.
ii. Alpha mendapatkan terlalu banyak rekomendasi yang tidak akurat.

Untuk kasus ini, jika dilihat secara seksama, penyebab masalahnya adalah Alpha baru membaca 1 berita. Bagaimanapun caranya, jika dilakukan pembandingan dengan pengguna lain untuk didapatkan rekomendasinya, hasil yang diberikan pasti tidaklah akurat apalagi jika misalnya seorang pengguna hanya salah pencet di halaman web. Tetapi, untuk idealnya, saat seseorang memiliki sebuah fitur rekomendasi, lebih baik salah memberikan rekomendasi (dan pada dasarnya tidak akan mungkin 100% benar) daripada tidak memberikan rekomendasi sama sekali. Oleh karena itu, untuk kasus ini masih berpihak kepada rumus hasil analisis penelitian ini.

V. Pengujian

Ketepatan hasil sistem rekomendasi sulit diuji. Misal seorang pengguna menyukai rekomendasi yang diberikan untuk pembelian sebuah barang, tetapi pengguna tersebut tidak membeli barang tersebut karena ada faktor penghambat misalnya tidak memiliki uang untuk membeli barang tersebut. Dalam hal ini, rekomendasi sudah tepat tetapi sistem sulit mengetahuinya. Untuk menguji ketepatan, hal yang dapat dilakukan hanyalah menanyakan kepuasan pelanggan terhadap rekomendasi yang dibuat.

Ketepatan rekomendasi berita juga sulit diukur. Misal seorang pengguna menerima rekomendasi yang diberikan tetapi pengguna tersebut ternyata sudah tahu berita tersebut dari media lain maka berita tersebut tidak akan dibaca.

Tujuan utama penelitian ini adalah memodifikasi rumus untuk kasus tanpa *rating*. Oleh sebab itu akan dibandingkan kecepatan pemakaian rumus asli dengan rumus hasil modifikasi. Tabel V adalah hasil pengujian perbandingan kecepatan antara algoritma *correlation-based similarity* biasa dengan *correlation-based similarity* khusus tanpa *rating*. Data yang dipakai adalah data fiktif dengan pola "Pengguna A sudah membaca berita X". Kolom pertama Tabel V menunjukkan banyaknya data berupa kombinasi pengguna dan berita.

Tabel III
Contoh Kasus 3

Pengguna	Berita 1	Berita 2	Berita 3	Berita 4	Berita 5	Berita 6	Berita 7	Berita 8
Alpha	✓	✓		✓		✓		
Beta	✓	✓	✓	✓	✓			✓
Charlie						✓	✓	

Tabel IV
Contoh Kasus 4

Pengguna	Berita 1	Berita 2	Berita 3	Berita 4	Berita 5	Berita 6	Berita 7	Berita 8
Alpha	✓							
Beta	✓	✓	✓	✓	✓	✓	✓	✓

Tabel V
Pengujian Kecepatan

Jumlah Data	***Correlation-based similarity* Biasa**	***Corellation-Based Similarity* Khusus**
1000000	0,016	0,015
10000000	0,078	0,063
10000000	0,078	0,078
100000000	0,764	0,735
100000000	0,75	0,719
100000000	0,797	0,734
200000000	1,703	1,632
200000000	1,547	1,501

VI. Kesimpulan

Correlation-based similarity dapat dimodifikasi untuk sistem rekomendasi tanpa *rating* sehingga dapat dihasilkan kecepatan yang lebih baik.

Daftar Referensi

[1] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender System Handbook*, New York, USA: Springer Science+Business Media, 2010.

[2] T. Kim and S. Yang, "An Effective Threshold-based Neighbor Selection", *Advances in Information Retrieval*, 2007.

Hans Yulian, mahasiswa Teknik Informatika Institut Teknologi Harapan Bangsa yang lulus pada tahun 2015 dengan lama studi 2 tahun 9 bulan dengan predikat Cum Laude, menulis buku "iPhone 5 dan iOS 6 plus Jailbreak" yang diterbitkan oleh Penerbit Andi.

Inge Martina, menyelesaikan S1 di Jurusan Teknik Informatika ITB pada Maret 1990 dan melanjutkan S2 di Jurusan Teknik Informatika ITB hingga selesai pada bulan Agustus 2014. Minat penelitian pada efisisiensi algoritma. Saat ini menjabat sebagai Kepala Departemen Teknik Informatika ITHB.

Halaman kosong