Codeforces Problem Recommender dengan Metode Collaborative Filtering

Kelompok 34
Muhammad Zaky Khairuddin - 1406622805 - B
Sang Agung Raditya Prawara - 1406573766 - A
Shandy Darma - 1406622692 - B

Ringkasan

Codeforces adalah sebuah website *online judge* yang ditujukan untuk melatih orang-orang yang tertarik dalam *Competitive Programming*. Terdapat banyak sekali soal-soal pemrograman yang berguna mengasah kemampuan pemrograman dan *problem solving*. Untuk membantu pengguna memilih problem untuk berikutnya dipecahkan, kami mengusulkan sebuah sistem rekomendasi untuk problem Codeforces. Sistem rekomendasi yang kami usulkan akan mempertimbangkan kemampuan pengguna, serta jenis dan tingkat kesulitan permasalahan-permasalahan yang ada, sesuai dengan data yang diperoleh melalui Codeforces dan APInya.

Latar Belakang

Di bidang ilmu komputer, kemampuan *programming* merupakan salah satu kemampuan yang paling penting untuk dikuasai, karena pada dasarnya basis dari pekerjaan yang akan dilakukan adalah *programming*.

Salah satu cara untuk mengasah kemampuan *programming* adalah dengan *competitive programming*. Menurut Halim (2013), *competitive programming* pada dasarnya adalah “Diberikan masalah yang umum di bidang ilmu komputer, selesaikan secepatnya!”. *Competitive programming* mengasah banyak kemampuan yang berkaitan dengan *programming*, seperti mengidentifikasi masalah, menganalisis algoritma, dan memandang masalah dari berbagai sudut pandang. Situs yang menyediakan berbagai permasalahan *competitive programming* disebut *online judge*. Untuk saat ini, sudah banyak *online judge* yang ada, seperti Codeforces, UVA, dan HackerEarth.

Codeforces merupakan salah satu *online judge* yang terkenal. Selain karena banyaknya soal yang ada, codeforces juga mengadakan kontes mingguan, di mana soal yang diberikan merupakan soal yang baru. Selain itu, codeforces juga sering mengadakan kompetisi yang disponsori oleh pihak ketiga, seperti ACM. Walaupun begitu, codeforces tidak memiliki sistem rekomendasi soal, yang kami rasa akan sangat membantu para pengguna untuk memilih soal yang akan diselesaikan. Oleh karena itu, kami berencana untuk mengembangkan sistem rekomendasi soal untuk codeforces.

Kemampuan *coding* merupakan hal yang diperlukan bagi semua orang yang berkecimpung dalam dunia *computer science*. Pada cabang *computer science* manapun, kemampuan coding yang tinggi merupakan hal diperlukan.

Online judge merupakan sebuah jenis sarana yang sering dipakai untuk mengasah kemampuan *coding.* Terdapat banyak problem set dengan tingkat kesulitan berbeda-beda yang ada untuk diselesaikan user.

Mencari problem untuk diselesaikan pada online judge terkadang tidaklah sederhana. Seringkali problem yang kita pikir akan cocok untuk level kemampuan kita ternyata terlalu sulit atau terlalu mudah. Hal ini mengurangi efektivitas pembelajaran dan mungkin niat untuk melanjutkan menyelesaikan problem. Oleh karena itu, terdapat sebuah keperluan untuk mengidentifikasi problem yang cocok untuk skill level user.

Tujuan

Melalui recommender system ini, kami berharap dapat membantu user Codeforces dari Indonesia mendapatkan permasalahan-permasalahan yang tepat untuk membantu mereka mengasah kemampuan *competitive programming*. Dengan ini, diharapakan kualitas *programmer* yang ada semakin meningkat, khususnya di Indonesia.

Data Science Question

Pertanyaan utama :

- Diberikan informasi seorang user Codeforces, problem mana saja yang dapat direkomendasikan bagi user tersebut?

Pertanyaan sekunder :

- Seberapa besar pengaruh jumlah problem solved pada ranking seorang user?
- Untuk use case kami, metode apakah yang paling cocok untuk menghitung nilai kemiripan user?

Data yang Digunakan

Dalam pengembangan sistem rekomendasi ini, sumber utama data yang akan kami gunakan diperoleh menggunakan API yang disediakan oleh Codeforces (http://codeforces.com/api/help/). Dari API Ini, dapat diperoleh informasi terkait suatu problem seperti nama problem, *tag-tag* problem tersebut, dan berapa banyak user yang telah solve problem tersebut. Informasi peringkat dan rating dari seorang user juga dapat diperoleh. Selain itu, dapat diperoleh informasi submission-submission lalu oleh seorang user, termasuk informasi *verdict* dari submission tersebut (OK, Wrong Answer, dan sebagainya). Walaupun detail informasi tersebut dapat diperoleh, daftar user tidak dapat diperoleh melalui API. Informasi terkait daftar user akan diambil secara langsung dari website Codeforces (pada halaman http://codeforces.com/ratings/).

Ruang Lingkup dan Batasan

Kami membatasi ruang lingkup pengerjaan kami pada pengguna Codeforces yang berasal dari Indonesia, baik yang aktif dalam 6 bulan terakhir maupun tidak. Problem-problem yang memiliki tag "**special*" tidak kami gunakan sebagai problem yang akan direkomendasikan, karena problem-problem tersebut cenderung memiliki syarat penyelesaian khusus. Salah satu contoh syarat khusus yaitu wajib menggunakan bahasa pemrograman tertentu, yang biasanya bukan merupakan bahasa pemrograman yang umum digunakan dalam *competitive programming*.

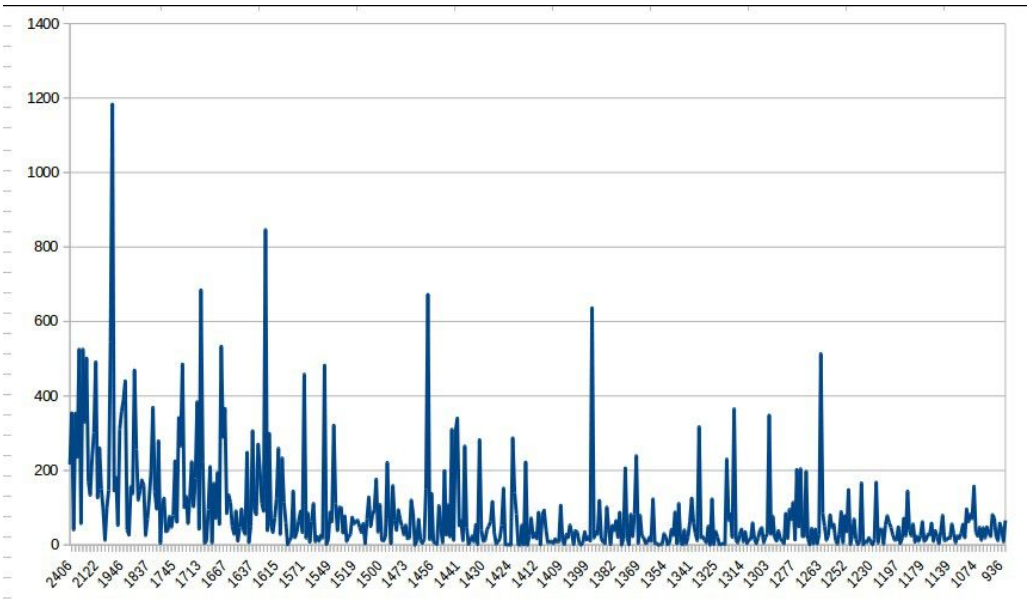
Studi Literatur

Recommender system untuk problem-problem *competitive programming* sudah pernah dibuat sebelumnya. Salah satu dari recommender systems tersebut adalah *Code-Drills* (https://code-drills.com/). Recommender system ini menggunakan tingkat kesulitan suatu problem berdasarkan banyak pengguna yang pernah memecahkan problem tersebut, dan riwayat pengguna yang akan direkomendasikan, untuk menentukan

rekomendasi problem yang akan diberikan. Pendekatan ini dapat dikatakan menggunakan metode *content-based*.

Recommender system yang kami buat memakai metode *collaborative filtering*. Salah satu tahap dari metode ini adalah mengukur kemiripan antara user. Pengukuran yang kami lakukan menggunakan Pearson correlation coefficient.

Cara untuk mengukur kemiripan fitur antara user tidak hanya melalui Pearson. Salah satu cara lain adalah menggunakan MANOVA. Kami tidak menggunakan MANOVA karena salah satu syarat penggunaan metode tersebut adalah jumlah outlier haruslah sedikit karena MANOVA rentan terhadap outlier. Sebuah visualisasi data menampakkan bahwa data yang kami olah memiliki beberapa outlier:



Kami juga mencari tahu mengenai *weighted correlation coefficient*. Metode ini menambah ‘bobot’ kepada sebuah variabel. Setelah kami telaah, ternyata metode ini jika diterapkan kepada use case kami akan mentransformasikan bobot nilai problem, sedangkan yang kami butuhkan adalah transformator bobot nilai user. Jadi metode *weighted correlation coefficient* kurang cocok untuk tujuan kami.

Metodologi

1. Data Retrieval

a) User List (username-list.txt)

Daftar pengguna situs *codeforces.com* yang berasal dari Indonesia. Data sejumlah 2745 user diambil dari *scraping web* pada laman list user menggunakan Chrome Web Scraper.

b) User Submission (API: getuserstatus.py + username-list.txt → userdata.txt)

Jawaban yang pernah diajukan pengguna untuk menyelesaikan masalah. Terdapat fungsi pada API codeforce yang mengeluarkan semua jawaban yang pernah diajukan oleh pengguna yang dijadikan parameter. Diambil semua jawaban yang pernah di-submit oleh pengguna Indonesia

c) Problem List & Statistics (API: apitest-problems.py → problems.txt)

Daftar semua problem yang ada pada codeforces. Diambil menggunakan API yang telah disediakan.

d) User Ratings (Scraping - BeautifulSoup: getuserrating.py + username-list.txt → user-rating.csv)

Codeforces memiliki sistem skor untuk menciptakan sebuah *leaderboard* dimana setiap rentang skor memiliki *title* unik. Diambil *rating* (skor) dari seluruh pengguna Indonesia menggunakan API yang telah disediakan, dan *title* kami tentukan menggunakan script sesuai dengan sistem skor yang ada.

2. Model Building

Dibuat tiga matriks data :

A) Rating Matrix

Matriks ini menggambarkan hubungan antara user dengan problem. Domain user adalah seluruh user Indonesia dan domain problem adalah seluruh problem yang pernah disolve oleh paling tidak satu user Indonesia.

Kolom merepresentasikan problem dan baris merepresentasikan user. Nilai preferensi memiliki nilai 0 - 1. Jika user telah memecahkan problem, nilai matriksnya 1. Jika user pernah mencoba menyelesaikan problem namun belum berhasil memecahkannya, nilai matriksnya 0.5. Jika user belum pernah mencoba menyelesaikan problem, nilai matriksnya 0. Penilaian ini dipilih karena matriks ini akan dikalikan dengan matriks lain sehingga nilai negatif akan mengubah representasi output.

	handle	1A	4A	71A	158A	118A	50A	231A	282A	96A	112A	339A	158B	116A	266A	131A
1	nathanajah	0.0	0.0	0	0.0	1	0	1	0.0	0	1	0	0.0	0.0	0	1
2	jonathanirvings	1.0	1.0	0	1.0	0	1	1	1.0	1	0	0	1.0	0.0	0	1
3	azariamuh	0.0	0.0	0	0.0	0	0	0	0.0	0	0	0	0.0	0.0	0	0
4	dolphinigle	0.0	0.0	0	1.0	0	1	0	0.0	0	0	0	1.0	1.0	0	0
5	ayaze	0.0	0.0	0	0.0	0	0	0	0.0	0	0	0	0.0	0.0	0	0
6	rais.fathin38	1.0	1.0	0	0.0	0	0	1	0.0	0	0	0	0.0	0.0	1	0
7	trivial	0.0	0.0	0	0.0	0	0	0	0.0	0	0	0	0.0	0.0	0	0
8	sokokaleb	0.5	1.0	1	1.0	1	1	1	1.0	0	1	0	1.0	0.0	0	1
9	Prabowo	0.0	0.0	0	0.0	0	0	0	0.0	0	0	0	0.0	0.0	0	1
10	wifi	0.0	0.0	1	0.0	0	0	0	0.0	0	0	0	0.0	0.0	0	0
11	Gyosh	1.0	0.0	0	1.0	0	0	0	0.0	0	0	0	1.0	1.0	0	0
12	athin	1.0	1.0	1	1.0	1	0	0	0.0	0	1	1	0.0	0.0	0	0
13	ptrrsn_1	0.0	0.0	0	1.0	1	0	0	0.0	0	0	0	1.0	0.0	0	1
14	fushar	0.0	0.0	0	1.0	0	1	0	0.0	0	0	0	0.5	1.0	0	1
15	zeulb	0.0	0.0	0	0.0	0	0	1	1.0	0	0	0	1.0	0.0	0	1
16	aguss787	1.0	0.5	0	0.0	0	0	0	0.0	0	0	0	0.0	0.0	0	0
17	azaky	0.0	0.0	0	0.0	1	0	1	1.0	0	0	0	0.0	0.0	0	0
18	chaotic_iak	1.0	1.0	1	1.0	0	0	0	0.0	0	0	0	0.0	0.0	0	0
19	handojo1	0.0	0.0	0	1.0	0	0	0	0.0	0	0	0	1.0	0.0	0	0
20	cfmode	0.0	0.0	0	0.0	0	0	0	0.0	0	0	0	0.0	0.0	0	0
21	tjandra	1.0	0.0	0	0.0	0	0	0	0.0	0	0	1	0.0	0.0	0	0
22	farisv	1.0	0.0	0	0.0	0	1	0	0.0	1	0	1	0.0	0.0	0	1

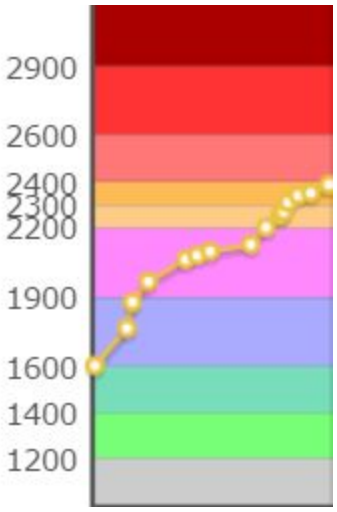
Showing 1 to 23 of 508 entries

Gambar x: Matriks antara user dan problem

B) Correlation Matrix

Dari matrix rating, kami hitung korelasi antara seorang user dengan user lainnya dengan menggunakan *Pearson correlation coefficient*. Hasilnya adalah matrix nilai kemiripan antara user.

C) User Weight Vector



Codeforces memiliki sistem title yang menggambarkan skor seorang user. Terdapat sepuluh *title* yang masing-masing diberikan kepada user dengan rentang skor tertentu.

Dibuat sebuah vector yang menggambarkan nilai kemiripan title satu user relatif dengan user lainnya. Semakin jauh ranking title satu user dengan user lain, semakin rendah nilai kemiripannya.

Nilai bobot yang diberikan kepada dua user yang memiliki title sama adalah 1. Setiap perbedaan satu tingkat title antara dua user akan mengurangi nilai bobot mereka sebanyak 0.1.

Ranking title sendiri dipengaruhi lumayan kuat oleh banyaknya problem yang telah di-solve oleh user tersebut. Sebuah aplikasi korelasi Pearson kepada kedua fitur tersebut menunjukkan keterkaitan yang lumayan kuat.

```
        Pearson's product-moment correlation

data:  user_solvedproblemcount$X2 and user_solvedproblemcount$X3
t = 11.409, df = 506, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3803177 0.5189236
sample estimates:
      cor
0.4523479
```

4. Recommendation System

Misalkan ada seorang user Zaky dengan username ZakyKh26 yang ingin menggunakan sistem rekomendasi ini. Langkah yang akan sistem lakukan adalah:

1. Mengambil baris nilai bobot user lain pada *user weight matrix* terhadap Zaky sebagai vektor w.

2. Mengambil baris nilai korelasi user Zaky dengan user lainnya pada correlation matrix sebagai matrix c.

$$w_{a,u} = \frac{\sum_{i \in I} \left(r_{a,i} - \overline{r_a}\right)\left(r_{u,i} - \overline{r_u}\right)}{\sqrt{\sum_{i \in I} \left(r_{a,i} - \overline{r_a}\right)^2} \sqrt{\sum_{i \in I} \left(r_{u,i} - \overline{r_u}\right)^2}}$$

3. Mengalikan matrix c dengan vektor w dan menghasilkan matrix m.
4. Dari matrix M yang berisi nilai hasil kemiripan user, dipilih 10 user yang paling mirip dengan Zaky.
5. Menghitung nilai prediksi preferensi untuk setiap soal menggunakan informasi yang didapatkan dari 10 user paling mirip tersebut.

$$p_{a,i} = \overline{r_a} + \frac{\sum_{u \in K} \left(r_{u,i} - \overline{r_u}\right) \cdot w_{a,u}}{\sum_{u \in K} w_{a,u}}$$

6. Dari semua soal tersebut, dipilih 5 soal yang memiliki nilai preferensi paling tinggi yang juga belum pernah diselesaikan ataupun dicoba oleh Zaky.
7. 5 soal tersebut diurutkan berdasarkan perkiraan tingkat kesulitan untuk direkomendasikan ke Zaky.

```
E:\Fasilkom\Kuliah\Mata Kuliah\Data Science & Analytics\2016-2017 Genap\Tugas Akhir\Workspace\recommender-code
forces\Codes>python recommender.py
Reading data... (../Data/user-problem.csv)
Reading data... (../Data/user-rating.csv)
Reading data... (../Data/problem-difficulty.csv)
Processing data...
Creating user list and user rating index list...
Creating problem list...
Welcome to Codeforces recommender system!
Input your username: ZakyKh26
Creating user-problem matrix...
Computing mean score of all users...
Computing pearson correlation coefficients between user ZakyKh26 and other users...
Calculating product preferrence scores...
Filtering recommendations...
Done! Recommended problems (sorted by difficulty, from easiest):
71A - Way Too Long Words (http://codeforces.com/problemset/problem/71/A)
118A - String Task (http://codeforces.com/problemset/problem/118/A)
158B - Taxi (http://codeforces.com/problemset/problem/158/B)
467A - George and Accommodation (http://codeforces.com/problemset/problem/467/A)
574A - Bear and Elections (http://codeforces.com/problemset/problem/574/A)
```

Soal yang direkomendasikan adalah soal 71A, 118A, 158B, 476A, dan 574B

Kekurangan

- Recommender ini tidak bersifat real-time. Hanya menggunakan data yang *up to date* sampai tanggal 24 Mei 2017.
- Update data hanya bisa dilakukan manual melalui scraping dan process melalui scripting.
- *Recommender* hanya bisa digunakan untuk *user* yang *handler*-nya telah disimpan.

Kesimpulan

Nilai preferensi yang dikeluarkan oleh recommender system memiliki value yang valid jika dilihat secara kasat mata. Anggota kami yang memiliki akun codeforces mencoba memakai sistem ini dan problem yang diberikan cocok dengan skill level.

Seberapa besar pengaruh jumlah problem solved pada rating seorang user?
Jika dilihat dari koefisien korelasi Pearson, korelasi antara jumlah problem solved dan rating user ~0.452. Kami merasa jumlah problem solved tidak terlalu berpengaruh terhadap rating seorang user.

Untuk use case kami, metode apakah yang paling cocok untuk menghitung nilai kemiripan user?

Kami telah mencoba tiga metode perhitungan nilai kemiripan user. MANOVA, weighted correlation, dan Pearson correlation. Dari ketiga metode tersebut hanya Pearson yang cocok untuk diaplikasikan ke profil data kami.

Daftar Pustaka

Halim, S., Halim, F., Skiena, S. S., & Revilla, M. A. (2013). *Competitive Programming 3*. Lulu Independent Publish.

"Code-Drills - Competitive Programming Resources And Problem Recommender". *Code-drills.com*. N.p., 2017. Web. 17 May 2017.

“Recommender Systems”. 2017. Retrieved from https://scele.cs.ui.ac.id/pluginfile.php/20272/mod_resource/content/1/5 Recommender systems.pdf

Google Chrome Web Scraper Extension : https://chrome.google.com/webstore/detail/web-scraper/jnhgnonknehpejjnehehllkliplmbmhn?hl=en

Almeida-de-Macedo, M. M., Ransom, N., Feng, Y., Hurst, J., & Wurtele, E. S. (2013). Comprehensive analysis of correlation coefficients estimated from pooling heterogeneous microarray data. *BMC bioinformatics*, *14*(1), 214.

Web Scraping - BeautifulSoup Tutorial: https://www.dataquest.io/blog/web-scraping-tutorial-python/

Web Scraping - BeautifulSoup Documentation: https://www.crummy.com/software/BeautifulSoup/bs4/doc/