

# Sistem Rekomendasi

Ali Akbar Septiandri

Universitas Al-Azhar Indonesia

*aliakbars@live.com*

December 31, 2019

# Selayang Pandang

- 1 Model Sistem Rekomendasi
- 2 Rekomendasi Berbasis Konten
  - Pembuatan Profil
  - Merekomendasikan Barang
- 3 Collaborative Filtering
  - Mengukur Kemiripan
  - Dualitas Kemiripan
  - Clustering
- 4 Dimensionality Reduction
- 5 The NetFlix Challenge

## Bahan Bacaan

- ① Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). *Mining of massive datasets*. Cambridge University Press. (Chapter 9)
- ② Wibisono, O. (10 July 2016). "The Many-Faces of Recommender System".  
<https://tentangdata.wordpress.com/2016/07/10/the-many-faces-of-recommender-system/>

# Model Sistem Rekomendasi

# Contoh Sistem Rekomendasi

- Menawarkan artikel untuk dibaca secara daring berdasarkan prediksi topik yang diminati
- Menawarkan saran untuk barang yang akan dibeli melalui situs e-commerce berdasarkan riwayat belanja atau pencarian

# Teknologi Sistem Rekomendasi

Secara umum, dibagi dua kategori besar:

- Sistem berbasis konten
- *Collaborative filtering*

# Utility Matrix

- Terdapat dua entitas utama: pengguna (*users*) dan barang (*items*)
- Matriks yang dibentuk merupakan preferensi pengguna terhadap barang yang ada
- Hasilnya kemungkinan besar adalah *sparse matrix*

## Contoh Utility Matrix

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

**Tabel:** *Utility matrix* yang merepresentasikan nilai film dalam skala 1-5  
[Leskovec et al., 2014, p. 308]



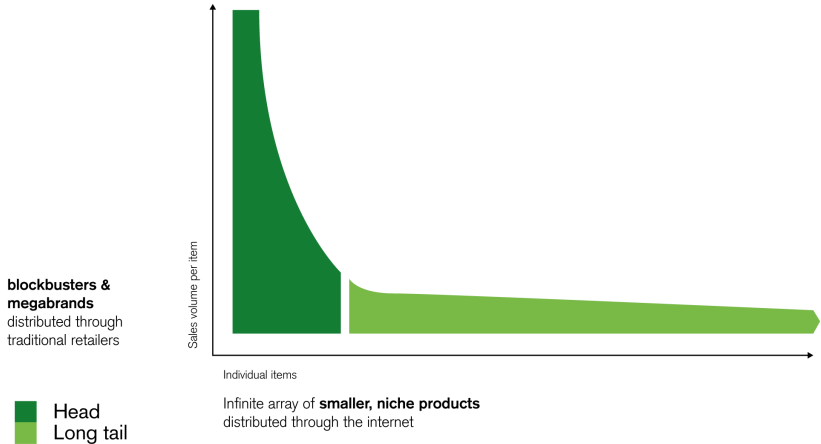
Tugas utama kita bukanlah untuk  
mengisi semua bagian yang masih kosong!

# The Long Tail Phenomenon

Toko konvensional terbatas karena:

- ① Sumber daya yang terbatas dari ruang, e.g. toko buku punya rak terbatas
- ② Tidak bisa menyimpan preferensi setiap pembeli
- ③ Sangat tergantung pada popularitas!

# The Long Tail Phenomenon



**Gambar:** Peluang penjualan di ceruk dalam "long tail phenomenon" —  
Sumber: CreditSuisse

## Mengisi Utility Matrix

- Pembuatan sistem rekomendasi dengan *utility matrix* bukan tanpa masalah

## Mengisi Utility Matrix

- Pembuatan sistem rekomendasi dengan *utility matrix* bukan tanpa masalah
- Tidak semua orang mau mengisi *rating*

## Mengisi Utility Matrix

- Pembuatan sistem rekomendasi dengan *utility matrix* bukan tanpa masalah
- Tidak semua orang mau mengisi *rating*
- Solusi: Anggap pembelian, konsumsi, atau bahkan pencarian sebagai bentuk “suka” terhadap produk tersebut

# Rekomendasi Berbasis Konten

Dalam rekomendasi berbasis konten, yang harus kita lakukan adalah membentuk *profil* untuk setiap barang atau pengguna.



## Contoh Profil Barang

Untuk kasus **film**, beberapa fitur yang bisa digunakan antara lain:

- aktor
- sutradara
- tahun pembuatan
- *genre*

## Pencarian Fitur

- **Tidak semua fitur** sudah langsung **tersedia** seperti kasus film atau buku

## Pencarian Fitur

- **Tidak semua fitur** sudah langsung **tersedia** seperti kasus film atau buku
- Bagaimana dengan kasus **dokumen**?

## Pencarian Fitur

- **Tidak semua fitur** sudah langsung **tersedia** seperti kasus film atau buku
- Bagaimana dengan kasus **dokumen**?
- Bagaimana dengan **gambar**, e.g. rekomendasi dalam Instagram?

## Pencarian Fitur

- **Tidak semua fitur** sudah langsung **tersedia** seperti kasus film atau buku
- Bagaimana dengan kasus **dokumen**?
- Bagaimana dengan **gambar**, e.g. rekomendasi dalam Instagram?
- Penting untuk merepresentasikan **fitur non-boolean** dengan benar!

## Profil Pengguna

Selain bisa membentuk profil barang, kita juga dapat membentuk profil pengguna berdasarkan barang yang ada.

## Profil Pengguna

Selain bisa membentuk profil barang, kita juga dapat membentuk profil pengguna berdasarkan barang yang ada.

### Example

Misalnya, jika dari seluruh film yang ditonton pengguna  $U$  terdapat 20% yang aktrisnya adalah Gal Gadot, maka profil pengguna  $U$  akan memiliki nilai 0.2 untuk komponen Gal Gadot.

## Contoh: Rekomendasi Film

Dua pendekatan yang bisa digunakan:

- Film apa yang mirip dengan salah satu film yang disukai pengguna  $U$ ?
- Berdasarkan preferensi pengguna  $U$ , apakah film baru yang akan direkomendasikan ini cocok?

Gunakan *cosine similarity* dan LSH!



# Penggunaan Algoritma Klasifikasi

- Buat model untuk **setiap pengguna**

# Penggunaan Algoritma Klasifikasi

- Buat model untuk **setiap pengguna**
- **Prediksi** *rating* untuk barang baru yang akan direkomendasikan

# Penggunaan Algoritma Klasifikasi

- Buat model untuk **setiap pengguna**
- **Prediksi** *rating* untuk barang baru yang akan direkomendasikan
- *Metrics* yang akan digunakan mungkin **bukan akurasi**

# Collaborative Filtering

Tidak perlu membuat profil,  
langung saja gunakan *utility matrix*!

## Jaccard Similarity

Berapa nilai *Jaccard similarity* untuk A & B? A & C?

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

**Tabel:** *Utility matrix* yang merepresentasikan nilai film dalam skala 1-5  
[Leskovec et al., 2014, p. 308]

Bagaimana jika kita menggunakan *cosine similarity*?

# Prapemrosesan

Dua pendekatan agar kemiripan yang dihasilkan lebih mengikuti intuisi:

- Pembulatan nilai
- Normalisasi nilai



## Pembulatan nilai

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	1			1			
B	1	1	1				
C					1	1	
D		1					1

**Tabel:** *Utility matrix* dengan pembulatan nilai [Leskovec et al., 2014, p. 323]

## Normalisasi nilai

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	2/3			5/3	-7/3		
B	1/3	1/3	-2/3				
C				-5/3	1/3	4/3	
D		0					0

**Tabel:** *Utility matrix* yang merepresentasikan nilai film setelah dinormalisasi [Leskovec et al., 2014, p. 324]

A dan C jadi sangat jauh =  $-0.559$ , sedangkan A dan B tidak terlalu dekat =  $0.092$

## Menggunakan Utility Matrix

- Nilai kemiripan yang kita hitung sebelumnya bisa dilakukan untuk pengguna maupun barang
- Masing-masing punya kelebihan dan kekurangan
- Kasus: Dalam rekomendasi musik, mungkin ada orang yang suka berbagai *genre*

# Perbandingan Pendekatan

## Pengguna

- Hanya perlu diproses sekali untuk tiap pengguna
- Ada kemungkinan ketertarikan dari *genre* yang berbeda

## Barang

- Informasi kemiripan antarbarang lebih *reliable*
- Perhitungannya bisa sangat lama

Yang jelas, hitung kemiripan terlebih dahulu!

## Masalah Lain

- Meski dua barang (e.g. musik atau film) ada dalam *genre* yang sama, hanya sedikit yang membeli keduanya
- Meski dua pengguna menyukai *genre* yang sama, mungkin sedikit barang yang sama yang dibeli
- Solusi: *clustering*!

# Clustering

- Alih-alih pengguna vs barang  $\rightarrow$  pengguna vs klaster
- Nilai yang kosong diganti dengan nilai rata-rata untuk klaster tersebut

## Contoh Clustering

- 1 Ganti nilai 3, 4, dan 5 menjadi 1; dan 1, 2, dan kosong menjadi 0
- 2 Hitung nilai Jaccard distance, buat klaster secara hierarki
- 3 Buat kembali matrix awal, lalu isi bagian kosong dengan rata-rata dari elemen yang tidak kosong dalam satu klaster



# Dimensionality Reduction

## UV-Decomposition

- Kita bisa melihat *utility matrix*  $M$  sebagai produk dari dua matriks  $U$  dan  $V$
- Maka, matriks  $5 \times 5$  direpresentasikan sebagai produk dari matriks  $U$  dan  $V$  dengan dimensi  $5 \times 2$  dan  $2 \times 5$
- Matriks terbaik didapatkan saat RMSE  $UV$  dengan  $M$  sekecil mungkin
- Matriks dengan dimensi kecil tersebut digunakan untuk aproksimasi nilai

## UV-Decomposition

$$\begin{array}{c} \text{U} \end{array}
 \begin{array}{c} \text{d1} \quad \text{d2} \\ \begin{array}{c} \text{A} \\ \text{B} \\ \text{C} \\ \text{D} \\ \text{E} \end{array} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \end{array}
 \times
 \begin{array}{c} \text{V} \\ \begin{array}{c} \text{d1} \\ \text{d2} \end{array} \begin{array}{c} \text{HP1} \quad \text{HP2} \quad \text{TW} \quad \text{SW1} \quad \text{SW2} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array} \end{array}
 =
 \begin{array}{c} \begin{array}{c} \text{HP1} \quad \text{HP2} \quad \text{TW} \quad \text{SW1} \quad \text{SW2} \\ \begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix} \end{array} \begin{array}{c} \text{A} \\ \text{B} \\ \text{C} \\ \text{D} \\ \text{E} \end{array} \end{array}$$

**Gambar:** Membentuk utility matrix dari perkalian dua matriks

# Optimasi

- Proses optimasi untuk pencarian  $UV$ -decomposition menggunakan *gradient descent*
- Mungkin terjebak optimal lokal
- Mungkin terjadi *overfitting*

Lihat [Leskovec et al., 2014, pp. 330-336]

# The NetFlix Challenge

# NETFLIX

Hadiah \$1,000,000 untuk yang bisa mengalahkan  
algoritma CineMatch sebesar 10% (RMSE)

# The NetFlix Challenge

Beberapa pengetahuan dari tantangan yang dimenangkan bulan September 2009 ini:

- Pemenangnya menggunakan gabungan beberapa algoritma
- Pendekatan *machine learning* tidak membutuhkan *genre*
- Waktu pemberian nilai berguna, karena jika seseorang sangat menyukai suatu film, akan segera diberi nilai



# Referensi



Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014)

Mining of Massive Datasets

*Cambridge University Press*

Terima kasih