Studi Kasus – Sistem Rekomendasi

Dr. rer. nat. Hendri Murfi

Intelligent Data Analysis (IDA) Group

Departemen Matematika, Universitas Indonesia – Depok 16424

Telp. +62-21-7862719/7863439, Fax. +62-21-7863439, Email. hendri@ui.ac.id

Machine Learning

Memahami keterhubungan (*relationships*) dan ketergantungan (*depedencies*) dalam suatu koleksi data adalah suatu aspek yang sangat penting dalam menganalisa koleksi data tersebut. Ketika tidak ada pendekatan pemodelan (*modelling approaches*) yang mudah untuk melakukan hal tersebut, maka pendekatan data (*data-driven approaches*) melalui metode-metode cerdas **machine learning** menjadi solusi alternatif.

Machine Learning

- *Preprocessing*: penentuan fitur-fitur yang relevan untuk masing-masing kasus
- *Learning*: penentuan parameter-parameter dari metode machine learning berdasarkan data training
- *Testing*: pengujian metode machine learning dengan data baru. Data testing tersebut harus dilakukan *preprocessing* yang sama dengan data training sebelum diolah oleh metode. Akurasi pada data testing, dikenal juga dengan istilah kapabilitas generalisasi, menjadi ukuran kinerja metode machine learning.

Learning

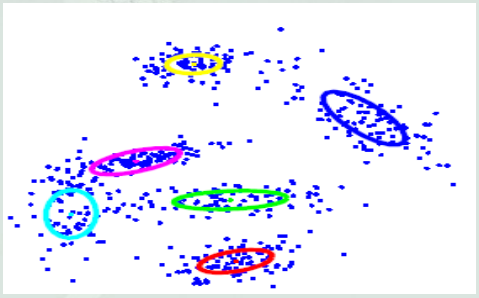
- Supervised Learning. Data training disertai target pembelajaran, yaitu $\{\boldsymbol{x}_n, t_n\}$, $n = 1 .. N$, dimana $\boldsymbol{x}_i$ adalah vektor input dan t adalah target. Tujuan pembelajaran adalah membangun model yang dapat memenuhi target pembelajaran, misal untuk *classification*, *regression*, *ordinal regression*, *ranking*, dll
- Unsupervised Learning. Data training tidak disertai target pembelajaran, yaitu $\{\boldsymbol{x}_n\}$, $n = 1 .. N$. Tujuan pembelajaran adalah membagun model yang dapat menemukan variabel/komponen tersembunyi pada data training, sehingga dapat digunakan untuk beberapa kebutuhan seperti: *density estimation*, *clustering*, *dimensionality reduction*, *topic/concept extraction*, *recommendation*, dll

Learning

- Semi-supervised Learning. Kombinasi dari supervised dan unsupervised learning, yaitu data training bisa memiliki atau tidak memiliki target untuk setiap vektor input.
- Reinforcement Learning. Tujuan pembelajaran adalah bagaimana aksi/tindakan yang harus dilakukan berdasarkan input/observasi dari lingkungan. Setiap aksi akan memiliki dampak terhadap lingkungan, dan lingkungan akan memberikan masukan terhadap setiap tindakan tersebut berupa *reward* yang akan menjadi acuan algoritma pembelajaran.
- Transfer Learning. Tujuan pembelajaran adalah bagaimana pengalaman belajar pada suatu masalah dapat digunakan untuk masalah yang lain.

Unsupervised Learning

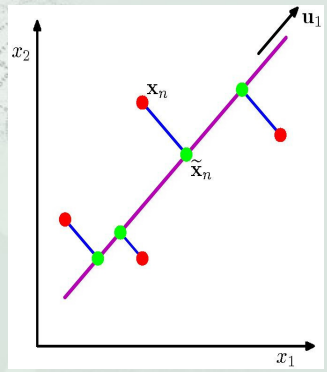
- Data training tidak disertai target pembelajaran, yaitu $\{\boldsymbol{x}_n\}$, $n = 1 .. N$.
- Bertujuan untuk membagun model yang dapat menemukan komponen/variabel tersembunyi pada data training. Selanjutnya dapat digunakan untuk beberapa kebutuhan seperti:
  - *Clustering*
  - *Feature Extraction*
  - *Dimension Reduction*
  - *Topic Modeling*
  - *Recommendations*



Recommendation

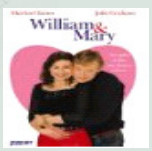
Diberikan data training $\{\boldsymbol{x}_n\}$, $n = 1$ sd N

- *Recommendation*
  - Permasalahan: memprediksi *rating* atau *preference* yang akan diberikan oleh seorang pengguna ke suatu *item*, misal musik, buku, film, dll. *Rating* ini akan menjadi dasar untuk merekomendasikan suatu *item* ke seorang pengguna



Recommendation

Contoh Sistem Rekomendasi



Recommendation

Strategi

- Secara umum, suatu sistem rekomendasi menggunakan salah satu dari dua strategi berikut ini, yaitu *content filtering* atau *collaboartive filtering*.
- *Content filtering* akan mendefiniskan profil untuk masing-masing pengguna atau *item*. Contoh profil film adalah genre, aktor, popularitas, dll; Profil pengguna adalah informasi geografis, isian kuisioner, dll.
  - Profil-profil ini memungkinkan sistem untuk mengasosiasikan pengguna dengan *items* yang sesuai.
  - Strategi berbasis konten ini membutuhkan informasi eksternal yang boleh jadi susah untuk diperoleh
- *Collaborative filtering* bergantung hanya pada aktifitas pengguna sebelumnya, misal transaksi sebelumnya atau pemberian *rating*, tanpa perlu mendefinisikan profil secara eksplisit.

Recommendation

Collaborative Filtering: nearest neighbors method

- Dua metode utama yang sering digunakan pada *collaborative filtering*, yaitu: *nearest neighbors method* dan *latent variabel models*.
- Ada dua cara untuk memprediksi rating pada metode *nearest neighbors,* yaitu dengan pendekatan berorientasi *item* (*item-oriented approach*) atau pendekatan berorientasi pengguna (*user-oriented approach*).
- Pada metode berorientasi *item*, prediksi *rating* yang akan diberikan oleh seorang pengguna kepada suatu *item* adalah berdasarkan rating yang diberikan oleh pengguna tersebut kepada *items* tetangga terdekat dari *item* tersebut.
- Contoh: untuk film *Saving Privat Ryan*, maka film-film tetangganya yang mungkin adalah film pertempuran, film Spielberg, film Tom Hanks, dll.

Recommendation

Collaborative Filtering: nearest neighbors method

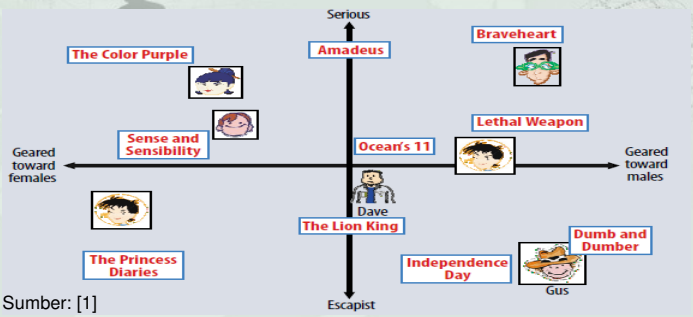
- Pada metode berorientasi pengguna, prediksi *rating* yang akan diberikan oleh seorang pengguna kepada suatu *item* adalah berdasarkan rating yang diberikan oleh pengguna tetangga terdekat kepada *item* tersebut.
- Contoh: Joe menyukai tiga film. Untuk membuat rekomendasi bagi Joe, sistem akan mencari pengguna yang mirip dengan Joe yang juga menyukai ketiga film tersebut, dan kemudian menentukan film lain yang mereka senangi. Pada contoh ini, ketiganya senang dengan film Saving Private Ryan, sehingga dijadikan rekomendasi pertama, dst.

Sumber: [1]

Recommendation

Collaborative Filtering: Latent Variable Models

- Latent variabel models akan memprediksi rating berdasarkan posisi relatif dari pengguna dan item pada beberapa variabel tersembunyi yang diekstrak dari pola rating. Untuk film, variabel tersembunyi tersebut boleh jadi genre, misal komedi, drama, aksi, anak-anak, dll.
- Contoh: Misal terdapat dua variabel tersembunyi sebagai sumbu x dan y, serta posisi relatif pengguna dan film pada koordinat tsb. Dari gambar, misal, kita dapat memprediksi bahwa Gus akan menyukai film *Dumb and Dumber*, tapi tidak suka film *The Color Purple*.



Sumber: [1]

Recommendation

Matrix Factorization

- Beberapa realisasi paling sukses untuk *latent variabel models* adalah berdasarkan *matrix factorization*. Metode ini menjadi populer karena memberikan skalabilitas yang baik dengan akurasi yang prediktif.
- Pada metode ini, data biasanya direpresentasi dalam bentuk matrik, dimana satu dimensi menunjukkan *items*, sementara dimensi lainnya menunjukan pengguna, yaitu matrik R_{mxn}, dimana r_{ij} menunjukkan *rating* dari *item i* oleh pengguna *j*.

	P-1	P-2	...	P-n
I-1	r_{11}	r_{12}	...	r_{1n}
I-2	r_{21}	r_{22}	...	r_{2n}
:	:	:		:
I-m	r_{m1}	r_{m2}	...	r_{mn}

Matrix Factorization

Representasi Sparse

- Biasanya matrik rating tersebut adalah jarang (*sparse*), karena seorang pengguna umumnya hanya memberikan rating kepada sebagian kecil *items* saja.

	p-1	p-2	...	p-n
o-1	r_{11}	-	...	-
o-2	-	r_{22}	...	r_{2n}
:	:	:		:
o-m	r_{m1}	r_{m2}	...	-

- Sehingga penggunaan *singular value decomposition* (SVD) untuk mengekstrak variabel tersembunyi seperti pada *latent semantic analysis* (LSA) hanya bisa dilakukan dengan sejumlah trik, misal mengganti entri yang tidak diketahui tersebut dengan nol (*imputation*) [2], akan tetapi menyebabkan peningkatan jumlah data yang jika tidak dilakukan secara benar akan merusak data.

Matrix Factorization

Formulasi Umum

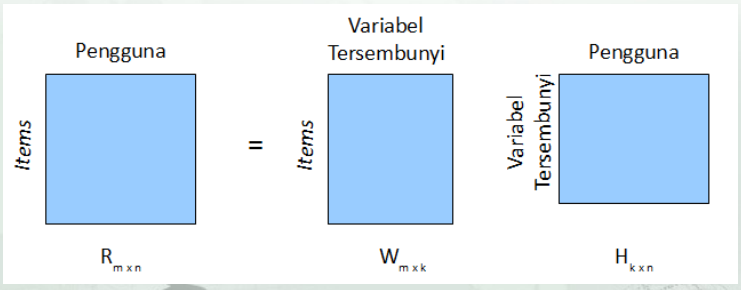
- Pendekatan alternatif adalah melakukan pemodelan secara langsung pada data *rating* yang diketahui saja (data training). Cara ini dapat diperoleh dengan melakukan formulasi masalah dengan bentuk umum sbb:

  Diberikan suatu matrik R_{mxn}, maka masalah faktorisasi matrik adalah mencari matrik W_{mxk} dan H_{kxn} sedemikian sehingga $R \approx WH$, atau:

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

Matrix Factorization

Formulasi Umum: Interpretasi



- Untuk film, variabel tersembunyi tersebut boleh jadi berupa *genre*, misal komedi, drama, aksi, anak-anak, dll.
- Vektor baris dari matrik W adalah representasi *items* relatif terhadap variabel tersembunyi yang terekstraksi
- Vektor kolom dari matrik H adalah representasi pengguna relatif terhadap variabel tersembunyi yang terekstraksi

Matrix Factorization

Formulasi Parsial

- Misal r_{ip} adalah rating yang berikan oleh pengguna *p* untuk item *i*, $\boldsymbol{w}_i$ adalah vektor baris dari *W* yang menunjukan vektor item *i*, $\boldsymbol{h}_p$ adalah vektor kolom dari *H* yang menunjukkan vektor pengguna *p*, maka:

$$\min_{\boldsymbol{h},\boldsymbol{w}} \; f(\boldsymbol{w},\boldsymbol{h}) = \frac{1}{2}\sum_{(i,p)\in T}(r_{ip}-\boldsymbol{w}_i^T\boldsymbol{h}_p)^2$$

dimana *T* adalah himpunan pasangan *(i,p)* dimana r_{ip} diketahui (data training)

- Untuk menghindari *overfitting*, bentuk teregularisasi sering juga digunakan, yaitu:

$$\min_{\boldsymbol{h},\boldsymbol{w}} \; f(\boldsymbol{w},\boldsymbol{h}) = \frac{1}{2}\sum_{(i,p)\in T}(r_{ip}-\boldsymbol{w}_i^T\boldsymbol{h}_p)^2 + \frac{\lambda}{2}(\|\boldsymbol{w}_i^T\|^2 + \|\boldsymbol{h}_p\|^2)$$

Matrix Factorization

Algoritma Gradient Descent

- Algoritma dasar yang digunakan untuk memecahkan masalah *matrix factorization* adalah metode *gradient descent* [3]. Algoritma ini memodifikasi parameter dengan suatu besaran α berlawanan arah gradien, yaitu:

Algoritma *Gradient Descent*
1. W = rand(m,k) 2. H = rand(k,n) 3. **while (not** *stoping criteria***) do** 4. H = H – α_H ∂f(W,H) / ∂H 5. W = W – α_W ∂f(W,H) / ∂W 6. **end while**

Matrix Factorization

Algoritma Gradient Descent

- Algoritma gradient descent tersebut memungkinkan kita untuk melakukan operasi hanya pada nilai rating yang diketahui saja, yaitu:

$$\boldsymbol{w}_i \leftarrow \boldsymbol{w}_i + \alpha_w (r_{ip} - \boldsymbol{w}_i^T \boldsymbol{h}_p) \boldsymbol{h}_p$$
$$\boldsymbol{h}_p \leftarrow \boldsymbol{h}_p + \alpha_h (r_{ip} - \boldsymbol{w}_i^T \boldsymbol{h}_p) \boldsymbol{w}_p$$

- Untuk bentuk teregularisasi akan menjadi:

$$\boldsymbol{w}_i \leftarrow \boldsymbol{w}_i + \alpha_w [(r_{ip} - \boldsymbol{w}_i^T \boldsymbol{h}_p) \boldsymbol{h}_p - \lambda \boldsymbol{w}_i]$$
$$\boldsymbol{h}_p \leftarrow \boldsymbol{h}_p + \alpha_h [(r_{ip} - \boldsymbol{w}_i^T \boldsymbol{h}_p) \boldsymbol{w}_p - \lambda \boldsymbol{h}_p]$$

dimana r_{ip} diketahui (data training)

Matrix Factorization

Algoritma Lain

- *Alternating least squares* [4]. Karena baik W maupun H tidak diketahui maka formulasi *matrix factorization* tidak *convex*. Akan tetapi, jika kita membuat tetap (konstan/tidak bebas) salah satu parameter tersebut, maka bentuknya menjadi *convex* dan dapat diselesaikan dengan metode *least squares* secara bergantian (*alternating*) sbb:

Algoritma *Alternating Least Squares*
1. W = rand(m,k)
2. **while (not** *stoping criteria***) do**
3. Pecahkan SPL $W^T WH = W^T R$ untuk mendapatkan H baru
4. Pecahkan SPL $HH^T W = HR^T$ untuk mendapatkan W baru
5. **end while**

- *Probabilistic matrix factorization* [5]
- *Maximum-margin matrix factorization* [6]

Matrix Factorization

Contoh

- Salah satu contoh aplikasi adalah sistem rekomendasi film pada *movielens.com* dimana salah satu data eksperiment yang dibangun dari sistem tersebut dan dibuat terbuka untuk publik memiliki karakteristik sbb:

1	Jumlah pengguna	943
2	Jumlah *items*	1682
3	Jumlah *rating*	100.000
4	*Sparsity*	93.7%
5	Jumlah data training	80.000
6	Jumlah data testing	20.000
7	*cross-validation*	5-fold

Referensi

1) Y. Koren, R. Bell, C. Volinsky. *Matrix Factorization Techniques for Recommender Systems*, The IEEE Computer Society, 2009
2) B. M. Sarwar et al., *Application of Dimensionality Reduction in Recommender System – A Case Study*. Proceeding of KDD Workshop on Web Mining for e-Commerce: Challenges and Oppurtunities, 2000
3) S. Funk. *Netflix Update: Try This at Home*. Dec. 2006 (http://sifter.org/~simon/journal/20061211.html)
4) R. Bell, Y. Koren. *Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights*. Proceeding IEEE International Conference on Data Mining, 2007
5) R. Salakhutdinov, A. Minih. *Probabilistic Matrix Factorization*. Advances in Neural Information Processing System 20, 2008
6) N. Srebreo, J. D. M. Rennie, T. S. Jaakkola. *Maximum-Margin Matrix Factorization*. Advances in Neural Information Processing System 17, 2005