# Matrix Factorization

**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

| Input | Model | Output |
|---|---|---|
| $x_1$, $x_2$, :, $x_D$ | $y(\mathbf{x},\mathbf{w})$ | *Klasifikasi, regresi, clustering, dll* |

- *Preprocessing*: ekstraksi fitur dan representasi data, misal dalam bentuk vektor $\mathbf{x}_i = (x_1, x_2, .., x_D)^T$
- *Learning*: pemilihan model dan penentuan parameter model, misal **w**, berdasarkan data pelatihan (*training data*)
- *Evaluation*: pengujian metode dengan data penguji (*testing data*) yang tidak sama dengan data pelatihan, sehingga didapat nilai estimasi untuk kapabilitas generalisasi dari model.

# Learning

Diberikan data pelatihan $\boldsymbol{x}_i$*, i = 1 sd N,* dan/atau $\boldsymbol{t}_i$*, i = 1 as N*

- Supervised Learning. Data pelatihan disertai target, yaitu {$\mathbf{x}_i$, $\mathbf{t}_i$}, i = 1 sd N. Tujuan pembelajaran adalah membangun model yang dapat menghasilkan output yang benar untuk suatu data input, misal untuk pengklasifikasian (*classification*) dan regresi (*regression*)

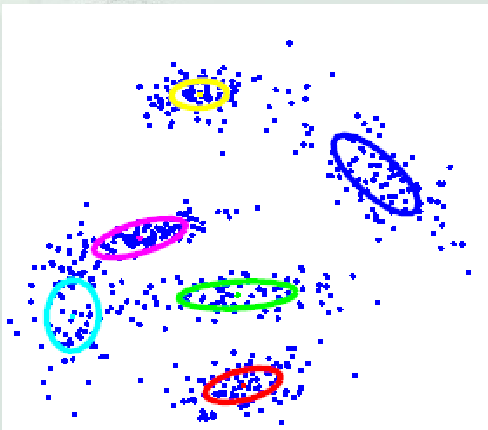
- Unsupervised Learning. Data pelatihan tidak disertai target, yaitu $\mathbf{x}_i$, i = 1 sd N. Tujuan pembelajaran adalah membagun model yang dapat menemukan variabel tersembunyi pada data pelatihan. Selanjutnya, variabel tersembunyi tersebut dapat digunakan untuk kebutuhan pemodelan (*latent variable models*)



# Unsupervised Learning

Bertujuan untuk mengekstraksi variabel tersembunyi pada data pembelajaran. Selanjutnya, variabel tersembunyi tersebut dapat digunakan, misal, untuk:

- *Clustering*
- *Features Extraction*
- *Dimensional Reduction*
- *Recommendation*
- *Topic Modeling*



## Unsupervised Learning

- Ada beberapa metode yang sering digunakan untuk mengekstrak variabel tersembunyi dari suatu koleksi data, antara lain:
  - *K-Means*
  - *Principal Component Analysis* (PCA)
  - *Independent Component Analysis* (ICA)
  - *Matrix Factorization*



## Matrix Factorization

- *Matrix Factorization* adalah penguraian suatu matriks menjadi beberapa buah matriks.
- Ada beberapa metode yang sering digunakan untuk memfaktorkan suatu matriks, antara lain:
  - *Singular Value Decomposition* (SVD)
  - *Nonnegative Matrix Factorization* (NMF)

# Matrix Factorization

## Singular Value Decomposition

- Diberikan suatu matriks $A_{mxn}$, SVD akan menguraikan matriks A menjadi:

$$A_{mxn} = U_{mxm}\ \Sigma_{mxn}\ V_{nxn}$$

dimana U dan V adalah matriks orthogonal, Σ adalah matriks diagonal dengan nilai-nilai pada diagonal lebih besar atau sama dengan nol

- Bentuk faktorisasi diatas dapat juga dimodifikasi menjadi:

$$\begin{aligned} A_{mxn} &\approx U_{mxk}\ \Sigma_{kxk}\ V_{kxn} \\ &= (U_{mxk}\ \Sigma_{kxk}^{1/2})\ (\Sigma_{kxk}^{1/2}\ V_{kxn}) \\ &= W_{mxk}\ H_{kxn} \end{aligned}$$

dimana k < rank(A)



# Matrix Factorization

## Nonnegative Matrix Factorization

- Diberikan suatu matriks $A_{mxn}$, NMF akan menguraikan matriks A menjadi:

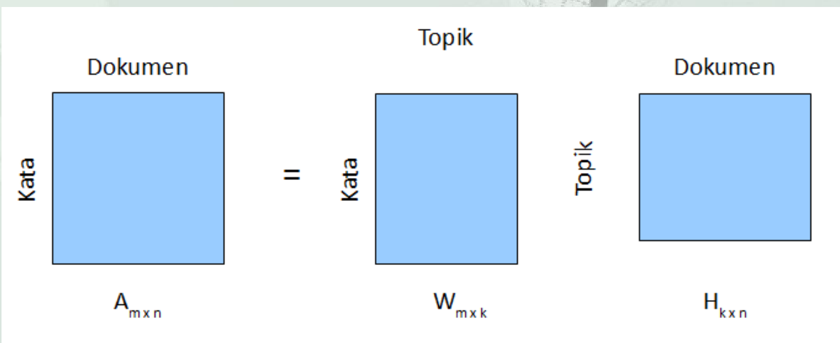
$$A_{mxn} \approx W_{mxk}\ H_{kxn}$$

dimana W dan H adalah matriks dengan entri-entri bernilai non negatif

# Matrix Factorization

## Latent Variable Models

- Misal matriks $A_{mxn}$ adalah matriks kata-dokumen, maka variabel tersembunyi pada data tersebut dapat berupa topik-topik:

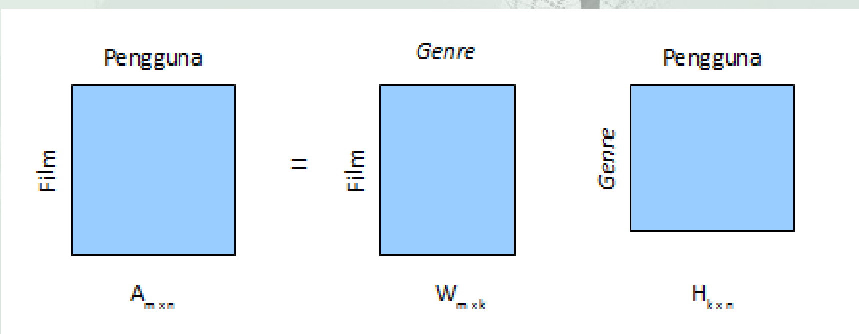





# Matrix Factorization

## Latent Variable Models

- Misal matriks $A_{mxn}$ adalah matriks film-pengguna, maka variabel tersembunyi pada data tersebut dapat berupa genre-genre film:



# Nonnegative Matrix Factorization

Formulasi Masalah

Diberikan suatu matriks non negatif V berukuran m x n dan suatu bilangan bulat positip k < min{m,n}, maka masalah NMF dapat diformulasikan sebagai suatu masalah optimasi berkendala sbb:

$$\min_{W,H} \quad f(W,H) = \frac{1}{2}\|V - WH\|^2$$
$$s.t. \quad W_{ij} \geqslant 0, H_{kl} \geqslant 0, \forall i,j,k,l$$

dimana W adalah matriks m x k, dan H adalah matrik k x n



# Nonnegative Matrix Factorization

Formulasi Masalah : Karakteristik Solusi

- *Definisi*: $S \in R^n$ adalah suatu *himpunan konveks* jika ruas garis lurus yang menghubungkan dua titik didalam *S* terletak semuanya didalam *S*. Secara formal, untuk sembarang dua titik $x \in S$ dan $y \in S$, maka $\alpha x + (1-\alpha)y \in S$

- *Definisi*: *f* adalah suatu *fungsi konveks* jika domainnya adalah suatu himpunan konveks dan jika untuk sembarang dua titik **x** dan **y** pada domain ini, maka grafik dari *f* terletak dibawah garis lurus yang menghubungkan $(\mathbf{x}, f(\mathbf{x}))$ ke $(\mathbf{y}, f(\mathbf{y}))$ di dalam ruang $R^{n+1}$, yaitu:

$$f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1-\alpha)f(\mathbf{y}), \text{ untuk semua } \alpha \in [0,1]$$

# Nonnegative Matrix Factorization

Formulasi Masalah : Karakteristik Solusi

- *Definisi:* Suatu masalah optimasi berkendala disebut konveks jika fungsi objektif *f* adalah suatu *fungsi konveks* dan kendalanya membentuk suatu *himpunan konveks*.

- *Teorema*: Jika suatu masalah optimasi berkendala adalah konveks maka sembarang titik minimum lokal $\boldsymbol{x}^*$ adalah suatu titik minimum global dari *f*. Jika ada tambahan bahwa *f* dapat diturunkan, maka sembarang titik stasioner $\boldsymbol{x}^*$ adalah suatu titik minimum global dari *f*.

  *Bukti*: lihat [1] hal 349-350 dan hal 17



# Nonnegative Matrix Factorization

Formulasi Masalah : Karakteristik Solusi

- Fungsi f(W,H) adalah konveks pada W atau H, akan tetapi tidak konveks pada W dan H.
- Sehingga, solusi optimal yang realistis untuk formulasi masalah diatas adalah bersifat minimum lokal, bukan minimum global

# Nonnegative Matrix Factorization

## Metode

- Secara umum, ada tiga metode yang sering digunakan untuk menyelesaikan masalah NMF, yaitu:
  - *Gradient Descent*
  - *Multiplicative Update*
  - *Alternating Least Squares*



# Nonnegative Matrix Factorization

## Gradient Descent

| Algoritma 1. Metode *Projected Gradient Descent* |
|---|
| 1. $\mathbf{W}$ = rand(m,k), dimana $w_{ij} \geq 0$ |
| 2. $\mathbf{H}$ = rand(k,n), dimana $h_{ij} \geq 0$ |
| 3. **for** i = 1 to maxiter **do** |
| 4. $\mathbf{H} = \mathbf{H} - \alpha_{\mathbf{H}}\, \partial f / \partial \mathbf{H}$ |
| 5. Proyeksikan semua elemen negatif **H** pada 0 |
| 6. $\mathbf{W} = \mathbf{W} - \alpha_{\mathbf{W}}\, \partial f / \partial \mathbf{W}$ |
| 7. Proyeksikan semua elemen negatif **W** pada 0 |
| 8. **end for** |

- Karena hasil dari metode *gradient descent* mungkin negatif, maka memproyeksikan nilai negatif tersebut ke nilai non negatif terdekat adalah cara yang umum dilakukan (Langkah 5, 7) [2]

# Nonnegative Matrix Factorization

## Gradient Descent : Konvergensi

- Tanpa pemilihan yang cermat terhadap parameter ukuran langkah pencarian $\alpha_W$ dan $\alpha_H$, sedikit sekali yang dapat dikatakan terkait dengan konvergensi dari metode *gradient descent*.
- Penambahan proyeksi non negatif membuat analisis kekonvergenan metode ini semakin sulit dilakukan.



# Nonnegative Matrix Factorization

## Multiplicative Update

| Algoritma 2. *Multiplicative Update* |
|---|
| 1. **W** = rand(m,k), dimana $w_{ij} \geq 0$ |
| 2. **H** = rand(k,n), dimana $h_{ij} \geq 0$ |
| 3. **for** i = 1 to maxiter **do** |
| 4. $\mathbf{H} = \mathbf{H} .* (\mathbf{V}^T\mathbf{V}) ./ (\mathbf{W}^T\mathbf{W}\mathbf{H})$ |
| 5. $\mathbf{W} = \mathbf{W} .* (\mathbf{V}\mathbf{H}^T) ./ (\mathbf{W}\mathbf{H}\mathbf{H}^T)$ |
| 6. **end for** |

- Berbeda dengan metode *gradient descent* yang menggunakan operasi penjumlahan, metode *multiplicative update* menggunakan operasi perkalian untuk pembaruan nilai W dan H (Langkah 4,5) [3]

# Nonnegative Matrix Factorization

## Multiplicative Update : Konvergensi

- Metode ini menggunakan *gradient* dan sifat *continual non-increasing* dari fungsi objektif *f(W,H)* untuk mengklaim bahwa algoritma akan konvergen ke titik minimum lokal [4].
- Akan tetapi klaim ini kemudian dibantah pada beberapa penelitian
- Secara ringkas, konvergensi dari metode *multiplicative update rule* dapat dinyatakan sebagai berikut: ketika algoritma konvergen ke suatu titik limit yang terletak di dalam (*interior*) dari daerah layak (*feasible*), maka titik ini adalah suatu titik stasioner . Titik stasioner ini bisa jadi minimum lokal atau bukan. Jika titik limit tersebut terletak pada batas (*boundary*) dari daerah layak, maka stasioneritas dari titik tersebut tidak dapat ditentukan [5]



# Nonnegative Matrix Factorization

## Alternating Nonnegative Least Squares

| Algoritma 3. *Alternating Non-negative Least Squares* |
|---|
| 1. W = rand(m,k) |
| 2. **for** i = 1 to maxiter **do** |
| 3. **H** = $\min_{H \geq 0}$ f(**W**,**H**) |
| 4. **W** = $\min_{W \geq 0}$ f(**W**,**H**) |
| 5. **end for** |

- W adalah suatu konstanta, sehingga Langkah 3 adalah suatu masalah *nonnegative least squares*
- Sebaliknya H menjadi konstanta pada Langkah 4, sehingga Langkah 4 juga adalah suatu masalah *nonnegative least squares*
- Dengan kata lain, W dan H dicari dengan menggunakan *nonnegative least squares* secara bergantian (*alternating*)

# Nonnegative Matrix Factorization

## Alternating Nonnegative Least Squares : Konvergensi

- Sembarang titik limit yang dihasilkan oleh metode *alternating least squares* adalah suatu titik stasioner [5]
- Dengan kata lain, metode ini memiliki sifat kekonvergenan yang lebih baik dari dua metode sebelumnya



# Nonnegative Matrix Factorization

## Alternating Nonnegative Least Squares

- Selanjutnya, kinerja metode ini sangat dipengaruhi oleh metode yang digunakan untuk menyelesaikan sub masalah *nonnegative least squares*
- Sub masalah ini dapat juga dilihat sebagai masalah *bound-constrained optimization*, yaitu:

$$\begin{aligned} \min_{H} \quad & f(W,H) = \frac{1}{2}\|V - WH\|^2 \\ s.t. \quad & H_{kl} \geqslant 0, \forall k,l \end{aligned}$$

$$\begin{aligned} \min_{W} \quad & f(W,H) = \frac{1}{2}\|V - WH\|^2 \\ s.t. \quad & W_{ij} \geqslant 0 \, \forall i,j \end{aligned}$$

# Nonnegative Matrix Factorization

## Alternating Nonnegative Least Squares

- Beberapa metode yang telah dikembangkan untuk menyelesaikan masalah *bound-contrained optimization* tersebut antara lain:
  - *Projected gradient descent* [6]
  - *Active-set* [7]
  - *Block principal pivoting* [8]



# Alternating Nonnegative Least Squares

## Projected Gradient Descent

| Algoritma 4a. Metode *Projected Gradient Descent* |
|---|
| 1. **W** = prosedur sebelumnya |
| 2. **for** i = 1 to maxiter **do** |
| 3.    $\mathbf{H} = \mathbf{H} - \alpha_{\mathbf{H}}\, \partial f/\partial \mathbf{H}$ |
| 4.    Proyeksikan semua elemen negatif **H** pada 0 |
| 5. **end for** |

| Algoritma 4b. Metode *Projected Gradient Descent* |
|---|
| 1. **H** = prosedur sebelumnya |
| 2. **for** i = 1 to maxiter **do** |
| 3.    $\mathbf{W} = \mathbf{W} - \alpha_{\mathbf{W}}\, \partial f/\partial \mathbf{W}$ |
| 4.    Proyeksikan semua elemen negatif **W** pada 0 |
| 5. **end for** |

# Referensi


1) Nocedal, J., Wright, S. J. *Numerical Optimization*. Springer-Verlag, 1999
2) P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.
3) D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999
4) D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing System*, 2001
5) M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52: 155-173, 2007.
6) C. J. Lin. Projected gradient methods for non-negative matrix factorization. *Neural Computation*, 19:2756-2779, 2007.




# Referensi


7) H. Kim, H. Park. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications,* 30(2): 713–730, 2008
8) J. Kim, H. Park. Fast nonnegative matrix factorization: an active-set-like method and comparisons. *SIAM Journal on Scientific Computing*, 33(6): 3261-3281, 2011
9) S. Arora, R. Ge, R. Kanna, A. Moitra. *Computing a nonnegative matrix factorization – provably*. Proceedings of the 44$^{th}$ Symposium on Theory of Computing, New York, 2012