Recommendations

Alfan F. Wicaksono
Information Retrieval & NLP Lab.
Fasilkom, Universitas Indonesia

Anda mengunjungi sebuah toko buku online...

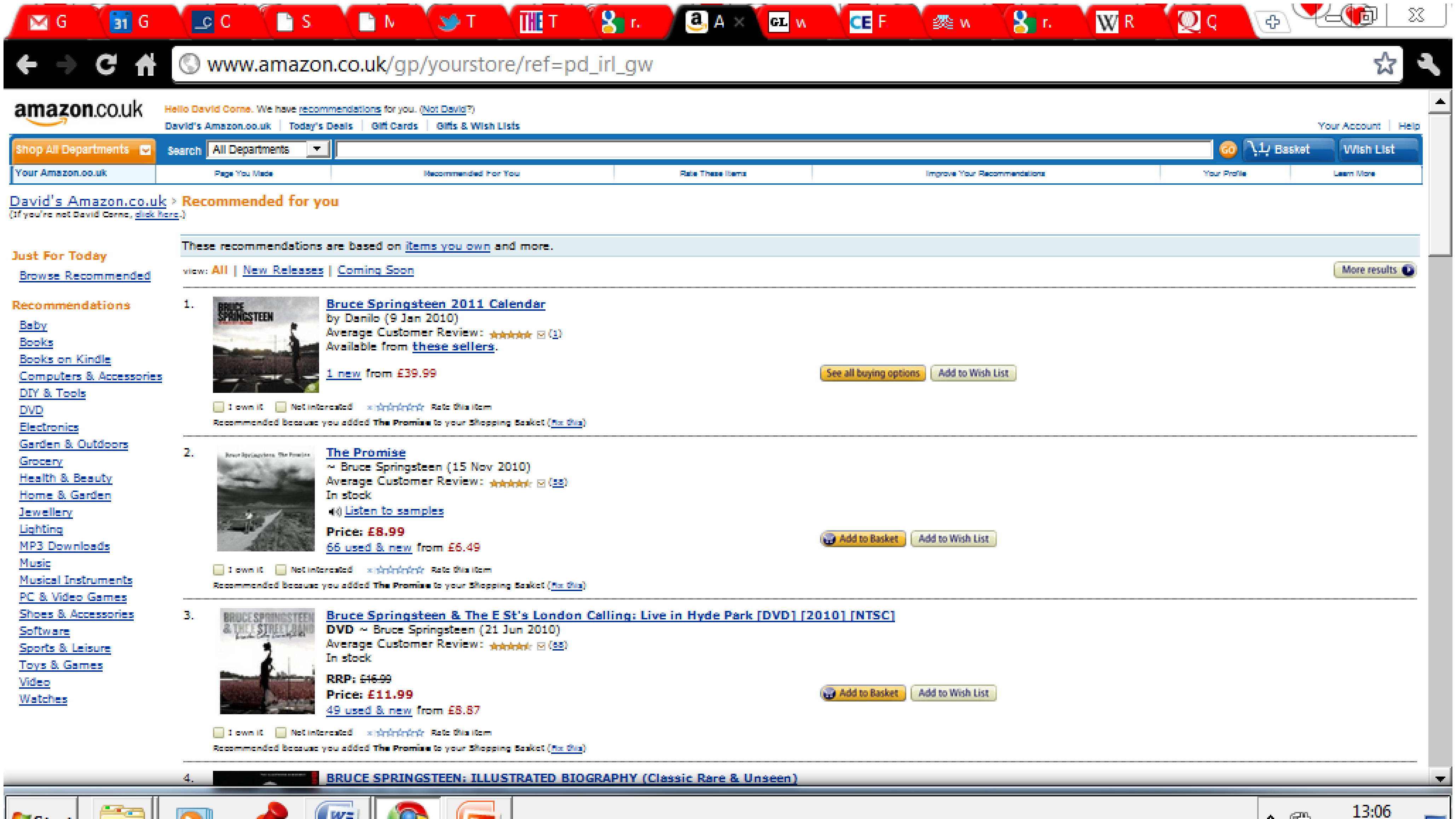
Toko buku online tersebut mempunyai **lebih dari 200.000** buku...

Pada halaman web utama, mereka akan menampilkan 6 buku, **khusus untuk Anda**.

Kira-kira, buku apa saja yang akan ditampilkan?

Amazon Product Search: Customers Who Bought This Item Also Bought...

Amazon Product Search: when logged in



Help people make decisions …

Difficulties in Decision Making:

- Which digital camera should I buy?
- Where should I spend my holiday?
- Which movie should I rent?
- Whom should I follow?
- Where should I find interesting news article?
- Which movie is the best for **our family**?

Why?

Big financial uplift if stores get recommendations “right”.

Problems:

- There are many choices
- We do not have enough resources to check all options (**information overload**)
- We do not have enough knowledge and experience to choose.
  - *“I’m lazy, but don’t want to miss out on good stuff.”*

Common Solutions:

- Consulting friends
- Search the Internet
- Following the crowd
  - Pick the item from top-lists
  - Best sellers

Can we automate all the above?

This is **Recommender System**

GOAL: To come up with a short list of items that fits user’s interests

Recommendations

Here we assume that users always express their explicit feedback. If they don't like it, they will give a rating of 1. No rating means no interaction.

Suppose we have a **user-item interaction** matrix:

Missing value tidak dipertimbangkan dalam pengembangan model

	book1	book2	book3	book4	book5
Alice	5	3	4	4	?
Rudi	3	?	2	3	3
Ahmad	4	3	4	3	5

Buatlah model yang bisa prediksi otomatis, kira-kira Alice akan suka tidak dengan book5?

Rating (1 - 5)

"Item" can be anything: book, product, movie, …

Recommendations

In practice, a user **<u>may choose to NOT react</u>** to a disliked item, leaving the entry for that item missing.

In most cases, we only have **implicit feedback**

	book1	book2	book3	book4	book5
Budi	1	?	1	1	?
Anto	1	?	?	1	?
Doni	?	1	?	1	1

Bagaimana memanfaatkan matriks ini (termasuk missing values) untuk mengembangkan model rekomendasi?

1: user clicks the link, and visits the book detail page
?: user don't click the detail link (no behavior)

No click → user bisa jadi suka, bisa jadi tidak suka

Recommendations - Approach

- Content-based Recommendations
- Collaborative Filtering
  - **Model-based Approach**
    → **Neural Networks, Embedding Models**
  - Memory-based Approach
- Hybrid Approach

Content-Based Approach

Content-Based Recommender Systems

- **Find me things that I liked in the past.**
- Machine *learns* preferences through user feedback and builds a user *profile*

- **Explicit feedback** – user rates items
- **Implicit feedback** – system records user activity
  - Clicksteam data classified according to page category and activity, e.g. browsing a product page
  - Time spent on an activity such as browsing a page

http://socialmediamining.info/

Content-Based Recommender Systems

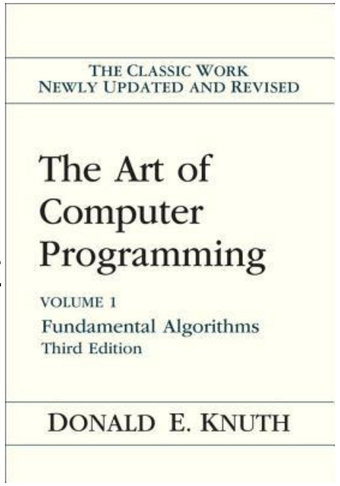
Selain dari history, *profile* juga bisa dibentuk dengan masukan langsung dari user ...

User Profile

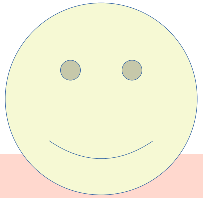
Items Recommended

Content-Based Recommender Systems

Item 1 – The Art of Computer Programming



Description:
The bible of all fundamental **algorithms** and the work that taught many of today's **software** developers most of what they know about **computer** **programming**. - Byte , September 1995 I can't begin to tell you how many pleasurable hours of study and recreation they have afforded me have pored over them in cars, restaurants, at work, at home ...

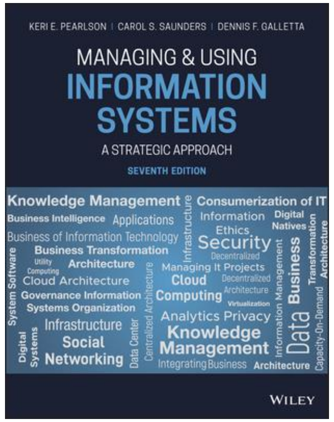


User Profile:

Programming information technology computer internet algorithm software

Item 2 – Managing and Using Information Systems: A Strategic Approach

Description:
Managing & Using **Information** Systems: A Strategic Approach provides a solid knowledgebase of basic concepts to help readers become informed, competent participants in **Information** Systems (IS) decisions.



Item mana yang seharusnya lebih besar similarity-nya dengan user profile?

Content-Based Recommender Systems

- We represent user profiles and item descriptions by vectorizing them using a set of keywords
- We can vectorize (e.g., using **TF-IDF**) both users and items and compute their similarity

$$I_j = (i_{j,1}, i_{j,2}, \ldots, i_{j,k}) \quad U_i = (u_{i,1}, u_{i,2}, \ldots, u_{i,k})$$

$$sim(U_i, I_j) = cos(U_i, I_j) = \frac{\sum_{l=1}^{k} u_{i,l} i_{j,l}}{\sqrt{\sum_{l=1}^{k} {u_{i,l}}^2}\sqrt{\sum_{l=1}^{k} {i_{j,l}}^2}}$$

We can recommend the top most similar items to the user

Collaborative Filtering

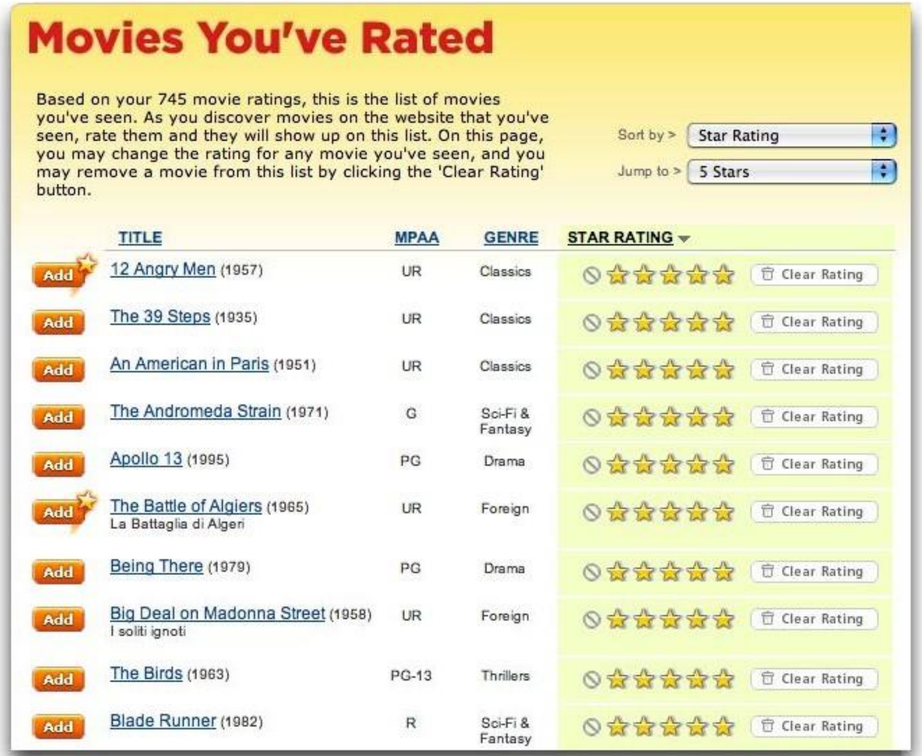
Collaborative Filtering

- Match people with similar interests as a basis for recommendation.

- **Approach**
  - use the **"wisdom of the crowd"** to recommend items

- **Basic assumption and idea**
  - Users give ratings to catalog items (implicitly or explicitly)
  - Customers who had similar tastes in the past, will have similar tastes in the future

Collaborative Filtering (CF)



Value	Graphic representation	Textual representation
5	☆ ☆ ☆ ☆ ☆	Excellent
4	☆ ☆ ☆ ☆	Very good
3	☆ ☆ ☆	Good
2	☆ ☆	Fair
1	☆	Poor

Table 9.1: User-Item Matrix

	Lion King	Aladdin	Mulan	Anastasia
John	3	0	3	3
Joe	5	4	0	2
Jill	1	2	4	2
Jane	3	?	1	0
Jorge	2	2	0	1

Input: Rating Matrix

Memory-Based Collaborative Filtering

Two memory-based methods:

	I1	I2	I3	I4
U1	1	2	4	4
U2	1	2	4	?
U3	2	5	2	2
U4	5	2	3	3

User-Based Collaborative Filtering

Users with similar **previous** ratings for items are likely to rate future items similarly

	I1	I2	I3	I4
U1	1	2	4	4
U2	1	2	4	?
U3	2	5	2	2
U4	5	2	3	3

Item-Based Collaborative Filtering

Items that have received similar ratings **previously** from users are likely to receive similar ratings from future users

"Naive" Collaborative Filtering

UNIVERSITAS INDONESIA
FAKULTAS ILMU KOMPUTER

Memanfaatkan kesamaan antar pengguna (*similarity between users*) untuk memprediksi kesukaan pengguna tersebut.

	buku1	buku2	buku3	buku4	buku5
Reni	4	2	3	1	?
Anto	3	2	3	1	5
Doni	1	1	5	5	1
Dewi	2	4	3	2	2

Salah satu cara prediksi, memanfaatkan rating dari Anto (pengguna lain yang paling mirip dengan Reni):

$$= \bar{r}_{reni} + (r_{anto,buku5} - \bar{r}_{anto})$$

$$= 2.5 + (5 - 2.8)$$

$$= 4.7$$

Kemiripan antara "Reni" dan "Anto" dapat dihitung dengan beberapa metrik seperti ***Pearson's Correlation Coefficient*** antara vektor Reni [4, 2, 3, 1] dan vektor Anto [3, 2, 3, 1].

"Naive" Collaborative Filtering

Cara sebelumnya tidak bersifat ***robust***! Perhatikan contoh:

$sim(Rivan, Doni) < sim(Rivan, Farhan)$

Oleh karena itu, basis untuk memprediksi seberapa suka Rivan dengan buku "Best Practices in C++" adalah Farhan. **Apakah ini OK? Apa yang terjadi?**

Problem: Prediksi rating yang diberikan Rivan.

	Menanam Pisang	Beternak Unggas	Programming with Python	Programming with Java	Best Practices in C++
Rivan	5	4	5	?	?
Doni	1	1	?	5	5
Farhan	5	3	1	1	1

Gagal menangkap keterkaitan antara "Programming with Python", "Programming with Java", dan "Best Practices in C++"

Collaborative Filtering

Approach Based On Embedding Models

Embedding Models

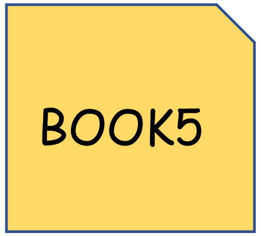
Embedding → Pemetaan setiap user & setiap item ke **(low) latent vector space.**

User Embedding

Budi

$$\begin{bmatrix} -0.40 \\ 0.18 \end{bmatrix}$$

Masing-masing element pada vector merepresentasikan suatu **"latent aspect"**



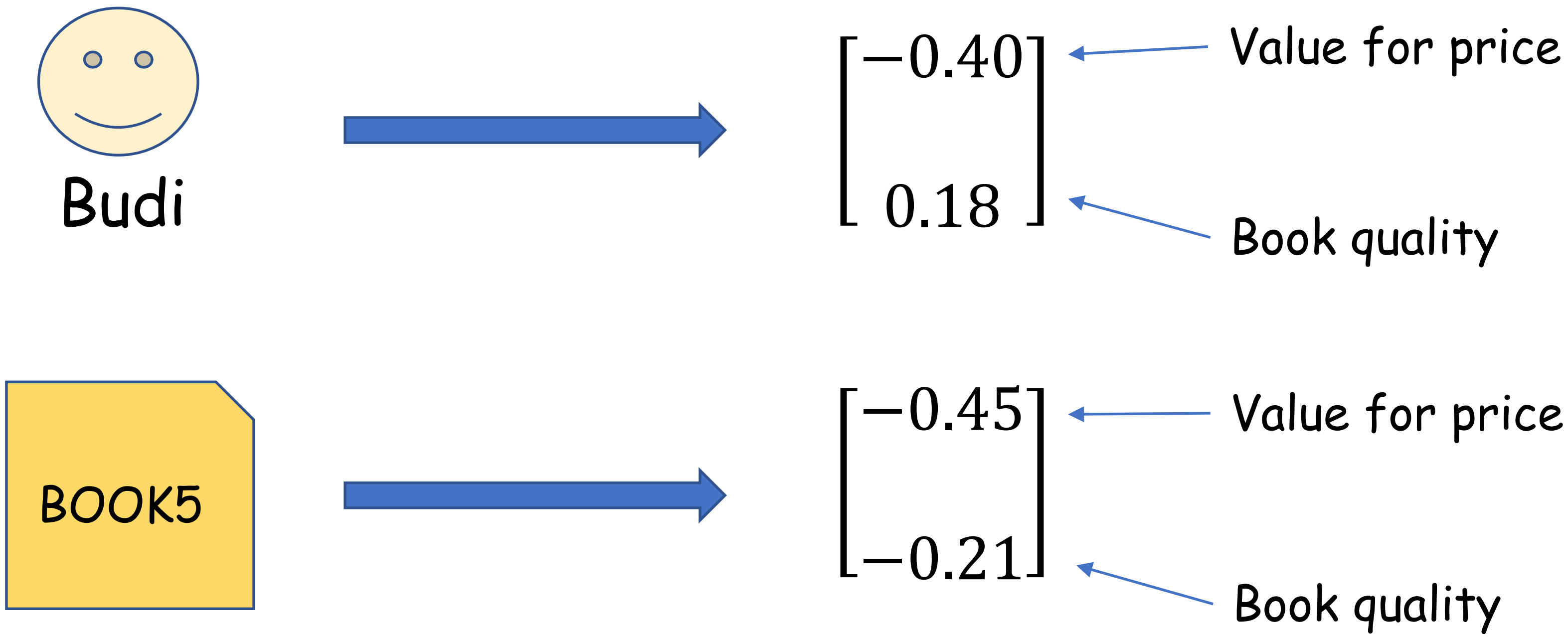
$$\begin{bmatrix} -0.45 \\ -0.21 \end{bmatrix}$$

Item Embedding

Embedding Models

Embedding → Pemetaan setiap user & setiap item ke **(low) latent vector space.**

Budi suka buku murah dan berkualitas

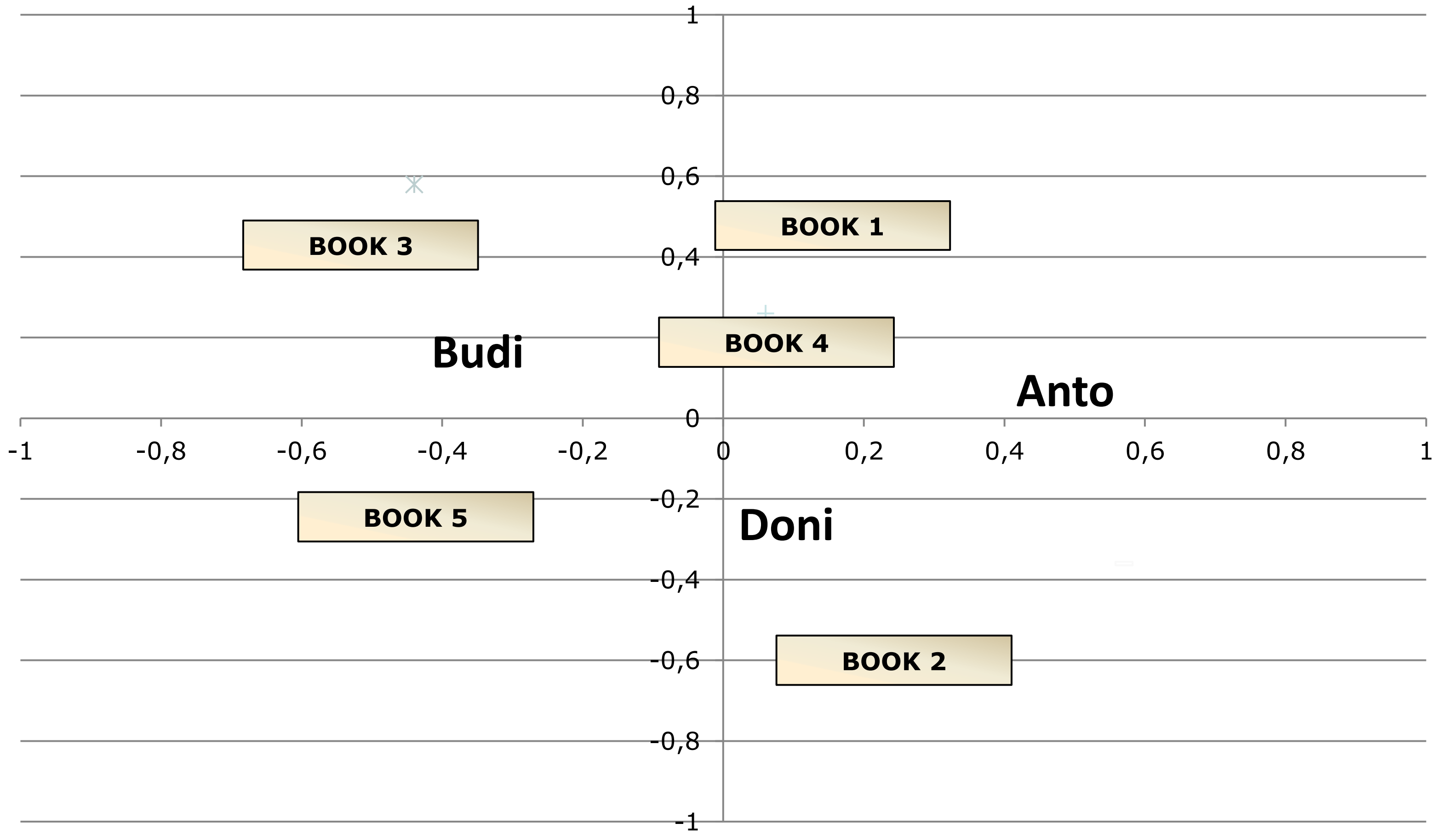


Misal…

Book5 adalah buku yang murah, namun kurang berkualitas

User & Item should share the same vector space

Contoh jika Panjang embedding vector dari user dan item adalah 2



How to predict the “preference”?

We simply compute dot product between user & item vectors

$$
\begin{aligned}
r_{budi,book5} &= p_{budi} \times q_{book5}^{T} \\
&= \begin{bmatrix} -0.40 & 0.18 \end{bmatrix} \times \begin{bmatrix} -0.45 \\ -0.21 \end{bmatrix} \\
&= (-0.40 \times -0.45) + (0.18 \times -0.21)
\end{aligned}
$$

Intuition: if there is hight correlation between user’s profile and items’ characteristics, the rating should be higher.

General Problem: Matrix Factorization

Problem: how to decompose an interaction matrix (real/binary) into a dot product of user & item matrices.

Di contoh ini, k = 2

	book1	book2	book3	book4	book5
Budi	1	?	1	1	?
Anto	1	?	?	1	?
Doni	?	1	?	1	1

≈

	F1	F2
Budi	-0.40	0.18
Anto	0.45	0.07
Doni	0.10	-0.23

•

	book1	book2	book3	book4	book5
F1	0.18	0.21	-0.50	0.10	-0.42
F2	0.43	-0.60	0.41	0.20	-0.21

Ukuran vector embedding

$$\hat{R}_{m \times n} \approx P_{m \times k} \cdot Q^T_{n \times k}$$

General Problem: Matrix Factorization

Problem: how to decompose an interaction matrix (real/binary) into a dot product of user & item matrices.

	book1	book2	book3	book4	book5
Budi	1	?	1	1	?
Anto	1	?	?	1	?
Doni	?	1	?	1	1

≈

	F1	F2
Budi	?	?
Anto	?	?
Doni	?	?

•

	book1	book2	book3	book4	book5
F1	?	?	?	?	?
F2	?	?	?	?	?

How to learn user & item embeddings from interaction matrix?

$$\hat{R}_{m \times n} \approx P_{m \times k} \cdot Q^{T}_{n \times k}$$

General Optimization Problem

	Item 1	Item 2	Item 3	...	Item n
User 1	3	?	5	1	?
...	1	?	?	2	?
User m	?	1	?	4	3

≈

	F1	F2
User 1	?	?
...	?	?
User m	?	?

•

	Item 1	Item 2	Item 3	...	Item n
F1	?	?	?	?	?
F2	?	?	?	?	?

$$\hat{R}_{m\times n} \approx P_{m\times k} \cdot Q^T_{n\times k}$$

Cari P dan Q sehingga:

$$\min_{P,Q} ||R - PQ^T||^2$$

Frobenius Norm: $||A|| = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j}^2}$

General Optimization Problem

	Item 1	Item 2	Item 3	...	Item n
User 1	3	?	5	1	?
...	1	?	?	2	?
User m	?	1	?	4	3

≈

	F1	F2
User 1	?	?
...	?	?
User m	?	?

•

	Item 1	Item 2	Item 3	...	Item n
F1	?	?	?	?	?
F2	?	?	?	?	?

$$\hat{R}_{m\times n} \approx P_{m\times k} \cdot Q^T_{n\times k}$$

Cari *user* & *item embedding* sehingga:

$$\min_{P,Q} \sum_{(u,i)\in Z} \left[\left(r_{ui} - p_u q_i^T \right)^2 \right]$$

Untuk setiap pasangan *user* u dan *item* i yang ada rating-nya di R

General Optimization Problem

	Item 1	Item 2	Item 3	...	Item n
User 1	3	?	5	1	?
...	1	?	?	2	?
User m	?	1	?	4	3

$\approx$

	F1	F2
User 1	?	?
...	?	?
User m	?	?

$\bullet$

	Item 1	Item 2	Item 3	...	Item n
F1	?	?	?	?	?
F2	?	?	?	?	?

$$\hat{R}_{m\times n} \approx P_{m\times k} \cdot Q^T_{n\times k}$$

Cari user & item embedding sehingga:

Predicted rating oleh model

$$\min_{P,Q} \sum_{(u,i)\in Z} \left[\left(r_{ui} - p_u q_i^T \right)^2 \right]$$

Rating asli yang diberikan *user* u terhadap *item* i

General Optimization Problem

	Item 1	Item 2	Item 3	...	Item n
User 1	3	?	5	1	?
...	1	?	?	2	?
User m	?	1	?	4	3

≈

	F1	F2
User 1	?	?
...	?	?
User m	?	?

•

	Item 1	Item 2	Item 3	...	Item n
F1	?	?	?	?	?
F2	?	?	?	?	?

$$\hat{R}_{m\times n} \approx P_{m\times k} \cdot Q^T_{n\times k}$$

Cari user & item embedding sehingga:

Secara praktis, perlu **regularization** agar mencegah overfitting dan menjadikan solusi unik, terutama jika **Matriks R sangat sparse**!

$$\min_{P,Q} \sum_{(u,i)\in Z} \left[\left(r_{ui} - p_u q_i^T\right)^2 \boxed{+ \gamma_p \|p_u\|^2 + \gamma_q \|q_i\|^2} \right]$$

Tunggu sebentar …

Kita sudah belajar Singular Value Decomposition (SVD) yang dapat digunakan untuk Matrix Factorization.

Mengapa kita tidak gunakan SVD saja di sini?

SVD tidak bisa digunakan karena Matriks Rating R tidak komplit, alias banyak yang tidak ada observed rating-nya (“banyak yang bolong-bolong”).

Loss function L?

Kasus: **explicit feedback**, dan interaction matrix berisi nilai real seperti **rating**.

L biasanya berjenis ***square-error***:

$$L(p_u, q_i, r_{ui}) = \left(r_{ui} - p_u q_i^T\right)^2 + \gamma_p \|p_u\|^2 + \gamma_q \|q_i\|^2$$

Predicted rating

True rating

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia
FAKULTAS ILMU KOMPUTER

Loss function L?

Kasus: **explicit feedback**, dan interaction matrix berisi nilai real seperti **rating**.

$$L(p_u, q_i, r_{ui}) = \left(r_{ui} - p_u q_i^T\right)^2 + \gamma_p \|p_u\|^2 + \gamma_q \|q_i\|^2$$

Gradient w.r.t model weights p_{uk} and q_{ik} for a user-item pair (u, i) :

$$\frac{\partial L}{\partial p_{uk}} = -2 \cdot \left(r_{ui} - p_u q_i^T\right) \cdot \frac{\partial p_u q_i^T}{\partial p_{uk}} + 2\gamma_p \cdot p_{uk} = -2 \cdot \left(r_{ui} - p_u q_i^T\right) \cdot q_{ik} + 2\gamma_p \cdot p_{uk}$$

$$\frac{\partial L}{\partial q_{ik}} = -2 \cdot \left(r_{ui} - p_u q_i^T\right) \cdot \frac{\partial p_u q_i^T}{\partial q_{ik}} + 2\gamma_q \cdot q_{ik} = -2 \cdot \left(r_{ui} - p_u q_i^T\right) \cdot p_{uk} + 2\gamma_q \cdot q_{ik}$$

```
import torch
import torch.nn.functional as fun
import numpy as np

# dummy rating matrix
# 5-star rating, dan 0 berarti tidak ada rating
# baris adalah user
# kolom adalah item
rating = np.array([[3, 0, 1, 4, 5],
                   [2, 1, 0, 4, 4],
                   [0, 1, 0, 3, 4],
                   [1, 5, 3, 0, 2],
                   [0, 5, 0, 1, 2]])


number_of_users, number_of_items = rating.shape
```

https://colab.research.google.com/drive/1pbXyUE9AueKV9ARzSrG1lgUrVM3FLsPu?usp=sharing

```
def get_user_tensor(id_user):
    return fun.one_hot(torch.tensor(id_user), \
                                    num_classes = number_of_users).float()

def get_item_tensor(id_item):
    return fun.one_hot(torch.tensor(id_item), \
                                    num_classes = number_of_items).float()

EMBEDDING_DIMS = 3

P = torch.rand(number_of_users, EMBEDDING_DIMS, requires_grad = False)
Q = torch.rand(number_of_items, EMBEDDING_DIMS, requires_grad = False)

# set nilai random awal agar uniform pada kisaran -
initrange hingga +initrange
initrange = 0.5 / EMBEDDING_DIMS
P = -2 * initrange * P + initrange
Q = -2 * initrange * Q + initrange
```

https://colab.research.google.com/drive/1pbXyUE9AueKV9ARzSrG1lgUrVM3FLsPu?usp=sharing

```
def loss(rating, P, Q, l2 = 0.02):
  total_loss = torch.tensor(0.0)
  for u, i in zip(*rating.nonzero()):
    r = torch.tensor(rating[u][i])

    p = torch.matmul(get_user_tensor(u), P)
    q = torch.matmul(get_item_tensor(i), Q)

    total_loss += (r - torch.dot(p, q)) \
                  + l2 * p.pow(2.0).sum() \
                  + l2 * q.pow(2.0).sum()
  return total_loss
```

https://colab.research.google.com/drive/1pbXyUE9AueKV9ARzSrG1lgUrVM3FLsPu?usp=sharing

```
def dloss_dp(u, i, l2 = 0.02):
  r = torch.tensor(rating[u][i])
  p = torch.matmul(get_user_tensor(u), P)
  q = torch.matmul(get_item_tensor(i), Q)
  return -2 * (r - torch.dot(p, q)) * q + 2 * l2 * p

def dloss_dq(u, i, l2 = 0.02):
  r = torch.tensor(rating[u][i])
  p = torch.matmul(get_user_tensor(u), P)
  q = torch.matmul(get_item_tensor(i), Q)
  return -2 * (r - torch.dot(p, q)) * p + 2 * l2 * q
```

https://colab.research.google.com/drive/1pbXyUE9AueKV9ARzSrG1lgUrVM3FLsPu?usp=sharing

```
EPOCHS = 100
LEARNING_RATE = 0.005

for i in range(EPOCHS):
  for u, i in zip(*rating.nonzero()):
    P[u] -= LEARNING_RATE * dloss_dp(u, i)
    Q[i] -= LEARNING_RATE * dloss_dq(u, i)
  l = loss(rating, P, Q)
  print("loss value", l.item())
```

https://colab.research.google.com/drive/1pbXyUE9AueKV9ARzSrG1lgUrVM3FLsPu?usp=sharing

```
print("true rating: ")
print(rating)

print("predicted rating: ")
print(torch.matmul(P, Q.T))
```

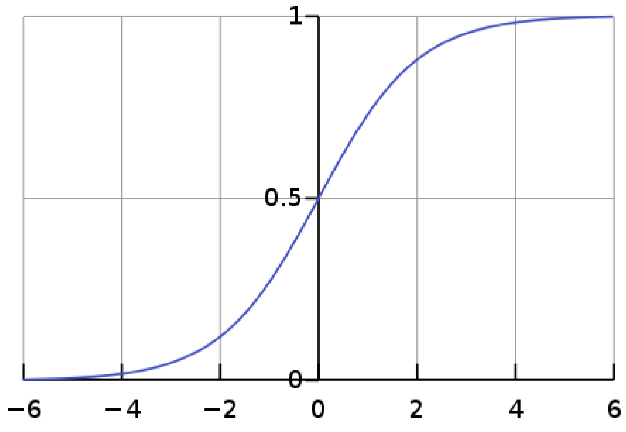
```
true rating:
[[3 0 1 4 5]
 [2 1 0 4 4]
 [0 1 0 3 4]
 [1 5 3 0 2]
 [0 5 0 1 2]]

predicted rating:
tensor([[2.7312, 1.3818, 1.0695, 4.1495, 4.9284],
        [2.2673, 0.9571, 0.7366, 3.5600, 4.1668],
        [2.0560, 1.0518, 0.7758, 3.1932, 3.7735],
        [0.9839, 5.0349, 2.8291, 1.0622, 2.0344],
        [0.8900, 4.9070, 2.6910, 1.0422, 1.9625]])
```

https://colab.research.google.com/drive/1pbXyUE9AueKV9ARzSrG1lgUrVM3FLsPu?usp=sharing

Loss function L?

Plot fungsi sigmoid (tanpa malu copy dari Wikipedia)



Kasus: **explicit feedback**, dan interaction matrix berisi nilai **binary**.

L biasanya menggunakan ***cross entropy***:

$$L(p_u, q_i, r_{ui}) = -r_{ui}\log\left(\sigma\left[p_u q_i^T\right]\right) - (1 - r_{ui})\log\left(1 - \sigma\left[p_u q_i^T\right]\right)$$

Fungsi sigmoid/logistic

Note: Regularization Term di-skip karena space slide

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia
FAKULTAS
ILMU KOMPUTER

Loss function L?

Kasus: **implicit feedback**

Di kasus ini, no missing data.

$$L = \sum_{u,i \in R} c_{ui}\left(\varphi_{ui} - p_u q_i^T\right)^2$$

Hu, Koren, and Volinsky (2008)

Loss function L?

Kasus: **implicit feedback**

Di kasus ini, no missing data.

Preference indicator untuk binarisasi interaction matrix:

$$\varphi_{ui} = \begin{cases} 1 & if\ r_{ui} > 0 \\ 0 & if\ r_{ui} = 0 \end{cases}$$

$$L = \sum_{u,i \in R} c_{ui}\left(\varphi_{ui} - p_u q_i^T\right)^2$$

Confidence level:

$$c_{ui} = 1 + \alpha \cdot r_{ui}$$

Hu, Koren, and Volinsky (2008)

Hu, Koren, and Volinsky (ICDM 2008)

Loss function L?

Kasus: **implicit feedback**

Di kasus ini, no missing data.

$$L = \sum_{u,i \in R} c_{ui}\left(\varphi_{ui} - p_u q_i^T\right)^2$$

Alternating Least Squares (ALS)

Optimasi dengan loss function ini lebih efisien jika dilakukan secara bergantian seperti berikut:

Loop hingga convergence:

- Freeze matriks P, lalu update weights di matriks Q
- Freeze matriks Q, lalu update weights di matriks P

Komputasi ALS dapat dilakukan secara parallel!

Loss function L?

Square-loss, Cross Entropy, ALS

- Model dioptimasi agar bisa menentukan apakah sebuah item akan disukai oleh seorang user atau tidak.
- Model **tidak dioptimasi** untuk permasalahan ranking, yaitu preferensi apakah user lebih pilih item A atau item B.

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia
FAKULTAS ILMU KOMPUTER

Loss function L?

Bayesian Personalized Ranking (Rendle et al., 2009)

	Item 1	Item 2	Item 3	Item 4
User 1	?	5	3	?
User 2	1	?	3	2

Extracting preference structures

$$R_1 = \begin{bmatrix} I_2 > I_1 \\ I_2 > I_3 \\ I_2 > I_4 \\ I_3 > I_1 \\ I_3 > I_4 \end{bmatrix}$$

User 1 lebih prefer Item 2 dibandingkan Item 1

$$R_2 = \begin{bmatrix} I_1 > I_2 \\ I_3 > I_1 \\ I_3 > I_2 \\ I_3 > I_4 \\ I_4 > I_1 \\ I_4 > I_2 \end{bmatrix}$$

Loss function L?

Bayesian Personalized Ranking (Rendle et al., 2009)

Given a user u with preference structure $\boldsymbol{R_u}$, bayes' rule shows that the model weights $\theta = (P, Q)$ can be computed as:

$$P(\theta|R_u) = \frac{P(R_u|\theta)P(\theta)}{P(R_u)} \propto P(R_u|\theta)P(\theta)$$

Loss function L?

Bayesian Personalized Ranking (Rendle et al., 2009)

The likelihood of preference structure for user **u**

$$P(R_u|\theta) = \prod_{u,i,j} P(I_i > I_j | u, \theta)$$

$$P(\theta|R_u) = \frac{P(R_u|\theta)P(\theta)}{P(R_u)} \propto P(R_u|\theta)P(\theta)$$

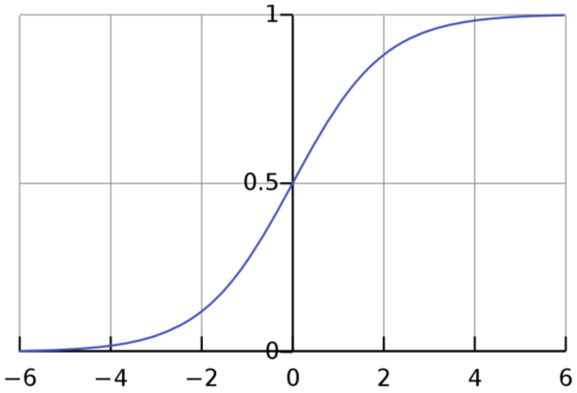
Loss function L?

Bayesian Personalized Ranking (Rendle et al., 2009)

The likelihood of preference structure for user **u**

$$P(R_u|\theta) = \prod_{u,i,j} \boxed{P(I_i > I_j|u,\theta)}$$

Plot fungsi sigmoid (tanpa malu copy dari Wikipedia)



Salah satu pilihan untuk memodelkan probability ini adalah dengan fungsi sigmoid terhadap **selisih** dari score kesukaan user **u** terhadap item I_i dan item I_j.

$$P\left(I_i > I_j \middle| u, \theta\right) = \sigma(p_u q_i^T - p_u q_j^T)$$

Loss function L?

Bayesian Personalized Ranking (**BPR**) (Rendle et al., 2009)

Tujuan kita adalah ingin cari $\theta = (P, Q)$ sehingga:

$$\max_{\theta} P(R_u|\theta)P(\theta)$$

Ini namanya **MAP (Maximum A Posteori)** estimation

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia
FAKULTAS ILMU KOMPUTER

Loss function L?

Bayesian Personalized Ranking (**BPR**) (Rendle et al., 2009)

Tujuan kita adalah ingin cari $\theta = (P, Q)$ sehingga:

Inilah yang namanya **BPR loss**

$$\max_{\theta} P(R_u|\theta)P(\theta) \Leftrightarrow \min_{\theta} -\sum_{u} \sum_{(i,j)\in R_u} \log\left(\sigma\left[p_u q_i^T - p_u q_j^T\right]\right) + \lambda_\theta \|\theta\|^2$$

Yang paling mudah adalah kita menggunakan **Independent Normal Distributions** untuk memodelkan prior dari setiap parameter/weight

Loss function L?

Bayesian Personalized Ranking (**BPR**) (Rendle et al., 2009)

Tujuan kita adalah ingin cari $\theta = (P, Q)$ sehingga:

Inilah yang namanya **BPR loss**

$$\max_{\theta} P(R_u|\theta)P(\theta) \Leftrightarrow \min_{\theta} -\sum_{u} \sum_{(i,j)\in R_u} \log\left(\sigma\left[p_u q_i^T - p_u q_j^T\right]\right) + \lambda_\theta ||\theta||^2$$

Jika setiap parameter mengikuti **distribusi normal**, ini sama saja dengan menambahkan **L2-regularizer** pada loss function dengan lambda untuk masing-masing parameter menandakan nilai variance-nya.

Loss function L?

Bayesian Personalized Ranking (**BPR**) (Rendle et al., 2009)

Ingat Kembali bahwa:

$$\frac{d\sigma(x)}{dx} = \sigma(x)(1 - \sigma(x))$$

$$\frac{d\log(f(x))}{dx} = \frac{f'(x)}{f(x)}$$

$$\Rightarrow \quad \frac{d\log(\sigma(x))}{dx} = 1 - \sigma(x)$$

Loss function L?

Untuk mempersingkat notasi, misal:

$$s_{uij} = s_{ui} - s_{uj} = p_u q_i^T - p_u q_j^T$$

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia
FAKULTAS ILMU KOMPUTER

Bayesian Personalized Ranking (**BPR**) (Rendle et al., 2009)

$$L_{BPR} = -\sum_{u} \sum_{(i,j)\in R_u} \log(\sigma[p_u q_i^T - p_u q_j^T]) + \lambda_\theta \|\theta\|^2$$

$$L_{BPR} = -\sum_{u} \sum_{(i,j)\in R_u} \log(\sigma[p_u q_i^T - p_u q_j^T]) + \lambda_p \|p_u\|^2 + \lambda_q \|q_i\|^2 + \lambda_q \|q_j\|^2$$

$$L_{BPR} = -\sum_{u} \sum_{(i,j)\in R_u} \log(\sigma[s_{uij}]) + \lambda_p \|p_u\|^2 + \lambda_q \|q_i\|^2 + \lambda_q \|q_j\|^2$$

Loss function L?

Untuk mempersingkat notasi, misal:

$$s_{uij} = s_{ui} - s_{uj} = p_u q_i^T - p_u q_j^T$$

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia
FAKULTAS ILMU KOMPUTER

Bayesian Personalized Ranking (**BPR**) (Rendle et al., 2009)

$$L_{BPR} = -\sum_{u} \sum_{(i,j)\in R_u} \log(\sigma[s_{uij}]) + \lambda_p \|p_u\|^2 + \lambda_q \|q_i\|^2 + \lambda_q \|q_j\|^2$$

$$\frac{\partial L_{BPR}}{\partial p_{uk}} = -\sum_u \sum_{(i,j)\in R_u} \left[\left(1 - \sigma(s_{uij})\right) \cdot \left(q_{ik} - q_{jk}\right) + 2\lambda_p \cdot p_{uk}\right]$$

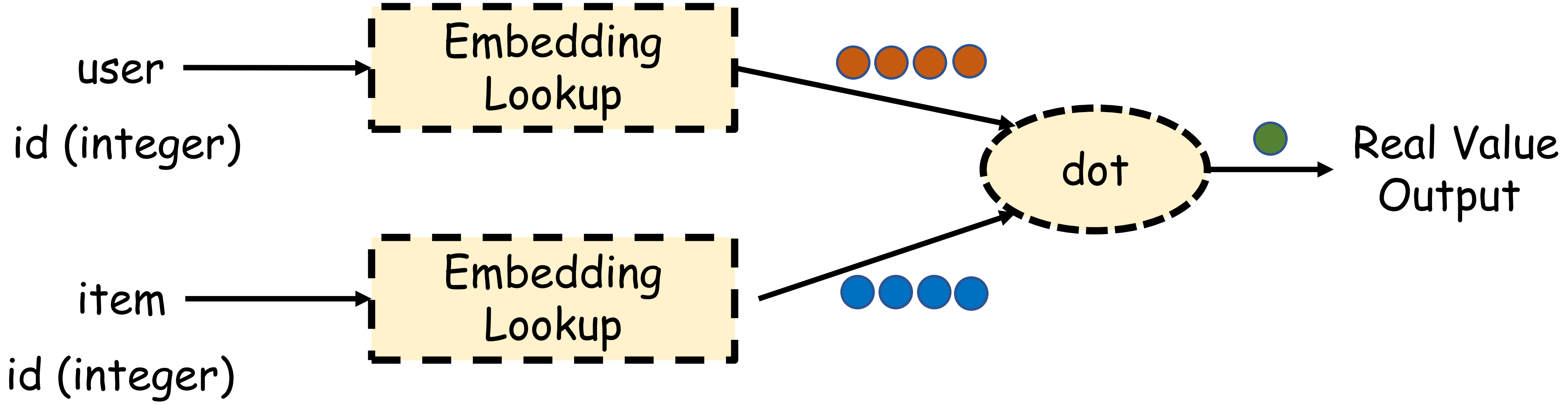
$$\frac{\partial L_{BPR}}{\partial q_{ik}} = -\sum_u \sum_{(i,j)\in R_u} \left[\left(1 - \sigma(s_{uij})\right) \cdot p_{uk} + 2\lambda_q \cdot q_{ik}\right]$$

$$\frac{\partial L_{BPR}}{\partial q_{jk}} = -\sum_u \sum_{(i,j)\in R_u} \left[\left(1 - \sigma(s_{uij})\right) \cdot (-p_{uk}) + 2\lambda_q \cdot q_{jk}\right]$$

Neural Networks

Factorization model can be framed as a Neural Network model.

Bagi yang familiar dengan Tensorflow/PyTorch:

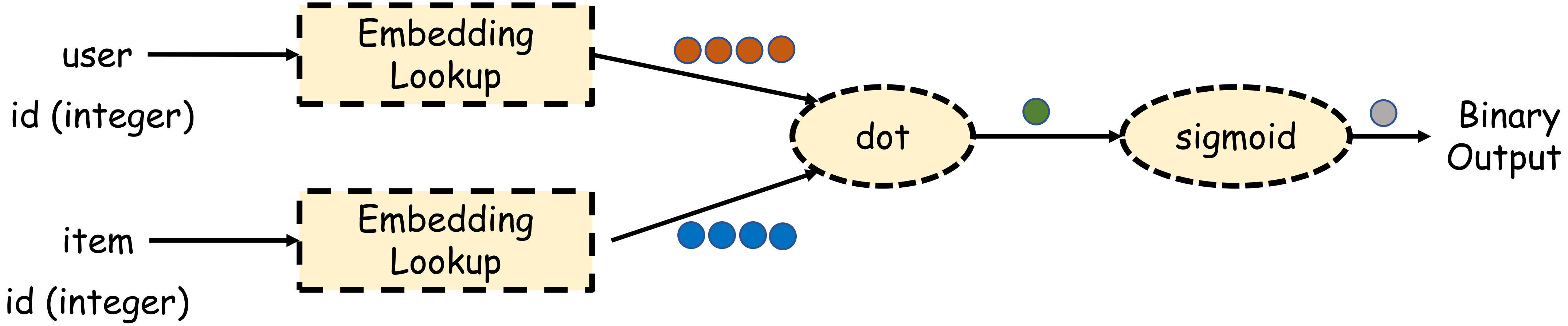


Jika output adalah real value

Neural Networks

Factorization model can be framed as a Neural Network model.

Bagi yang familiar dengan Tensorflow/PyTorch:

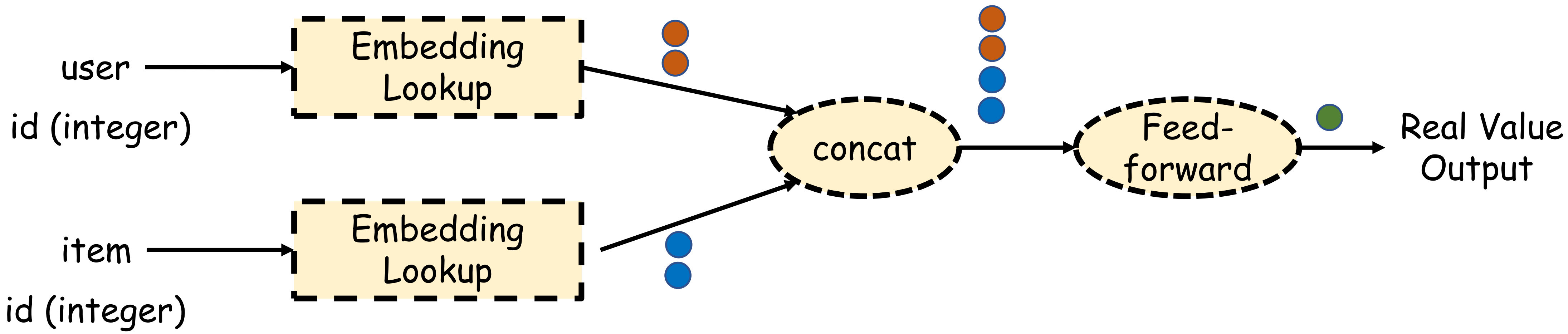


Jika output adalah binary

Neural Networks

Lebih umum lagi, kita juga bisa usulkan arsitektur berikut:

Bagi yang familiar dengan Tensorflow/PyTorch:



Reflection ☺

- Apa itu user & item **embeddings**? Dan apa kaitannya dengan **Matrix Factorization**?

- Apa perbedaan BPR loss-function dengan common loss-function lain seperti ALS, cross-entropy, dan squared-loss?

Materi berikutnya bersifat *Optional*, dan di luar kajian kuliah Information Retrieval saat ini.

Problems

- **Cold-start problem**
  - New user: ada user yang belum pernah ada history interaction
  - New item: ada item yang belum pernah di-rating sebelumnya
- Sparsity
- Privacy
- Serendipity
  - Recommend to me something that I don't know already

Cold-Start Problem

Ada seorang user yang belum ada interaction history!

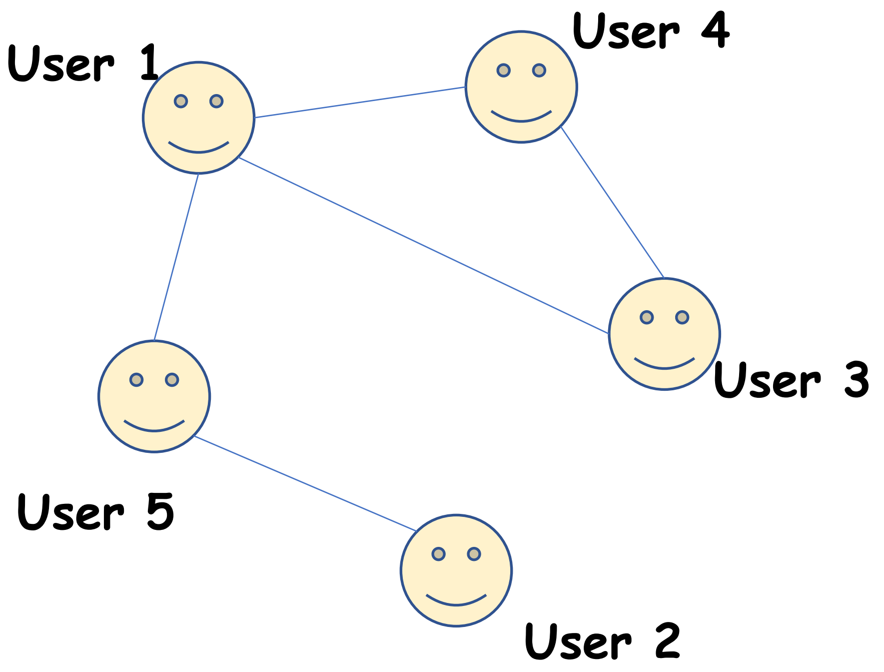
Sulit sekali membuat personalized recommendations untuk User 1

	Item 1	Item 2	Item 3	Item 4
User 1	?	?	?	?
User 2	1	?	3	2
User 3	1	3	1	5
User 4	2	1	?	4
User 5	2	3	2	3

Suppose social networks are available

Hmm...sepertinya kita bisa manfaatkan Social Networks

Pertemanan di Facebook



	Item 1	Item 2	Item 3	Item 4
User 1	?	?	?	?
User 2	1	?	3	2
User 3	1	3	1	5
User 4	2	1	?	4
User 5	2	3	2	3

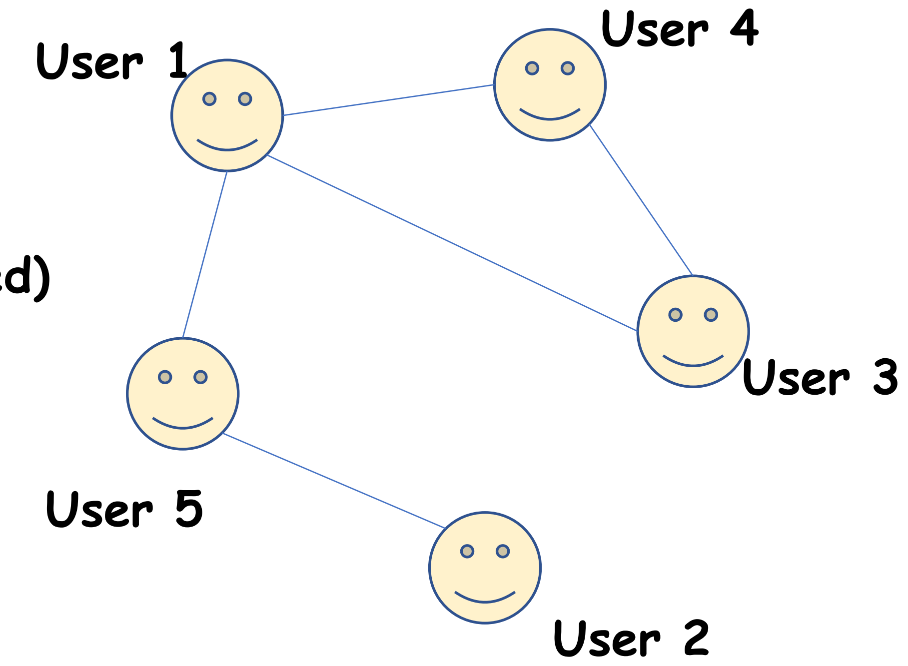
Observation (Guo et al., AAAI '15)

"A user's ratings have a weakly positive correlation with the average of her social neighbors under the concept of trust-alike relationships, and a strongly positive correlation under the concept of trust relationships."

Observation (Guo et al., AAAI '15)

*"A user's ratings have a weakly positive correlation with the average of her social neighbors under the concept of **trust-alike relationships**, and a strongly positive correlation under the concept of trust relationships."*

Trust-alike relationship = Friendship (undirected)

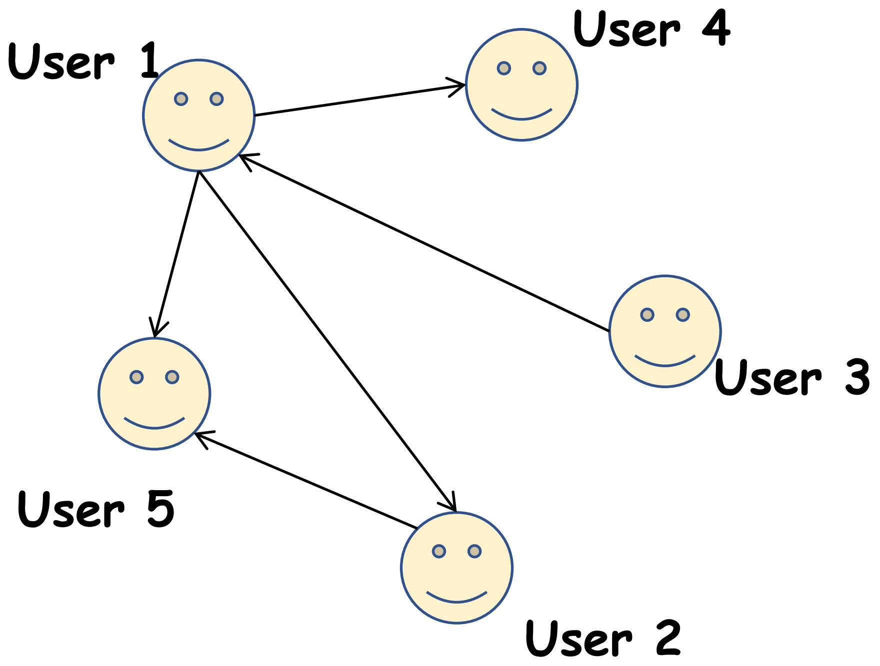


Observation (Guo et al., AAAI '15)

"A user's ratings have a weakly positive correlation with the average of her social neighbors under the concept of trust-alike relationships, and a strongly positive correlation under the concept of ***trust relationships****."*

Explicit trust relationship (directed)

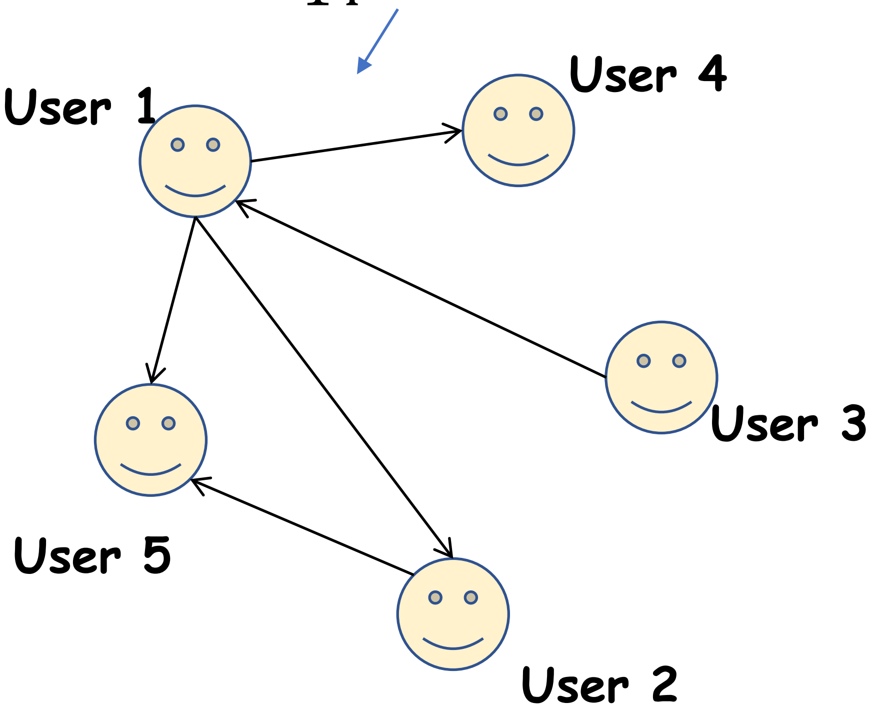
A → B : A trusts B



Jadi, bagaimana memanfaatkan data social networks?

Sabar ☺ kita pahami dulu beberapa notasi

$A_{14} = 1$



$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Jadi, bagaimana memanfaatkan data social networks?

Two common approaches:

1) Introducing a new regularization term
2) Setting social networks (adjacency matrix) as input to enhance the original user embeddings

Jadi, bagaimana memanfaatkan data social networks?

Regularization term?

L2-Regularizer biasa

$$\min_{P,Q} \sum_{u,i \in R} \left[L(p_u, q_i, r_{ui}) + \gamma_p ||p_u||^2 + \gamma_q ||q_i||^2 \right]$$

BPR, ALS, …

Jadi, bagaimana memanfaatkan data social networks?

Regularization term?

L2-Regularizer biasa

$$\min_{P,Q} \sum_{u,i \in R} \left[L(p_u, q_i, r_{ui}) + \gamma_p ||p_u||^2 + \gamma_q ||q_i||^2 \right]$$

$$+ \sum_{j,k} A_{j,k} \left\| u_j - u_k \right\|^2$$

BPR, ALS, …

"Social Regularization Term" (Ma et al, 2011)

Jadi, bagaimana memanfaatkan data social networks?

Guo et al., (2015) incorporates social influence into common scoring function.

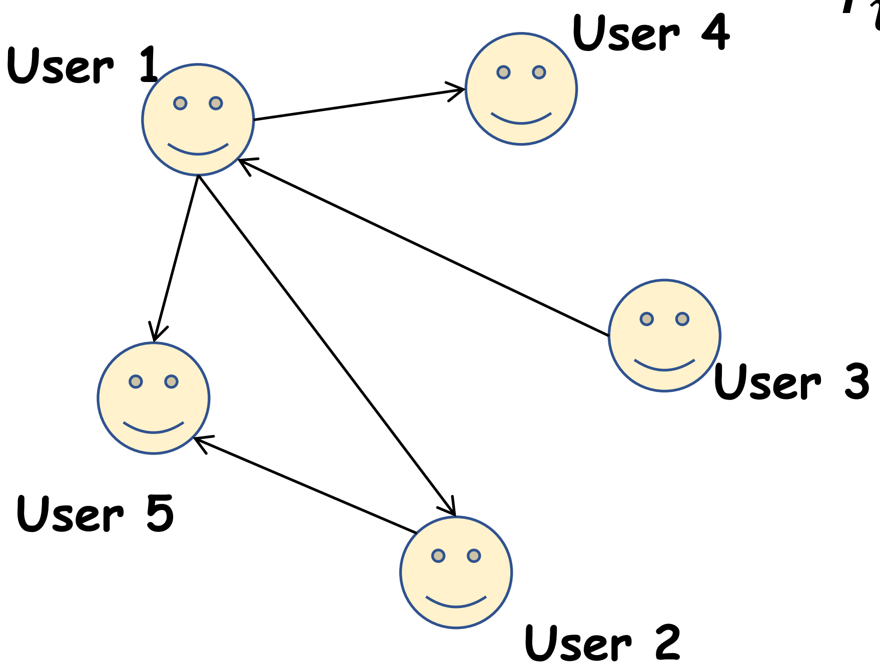
Recall our scoring function: $\hat{r}_{ui} = p_u q_i^T$

Sebelum kita gunakan social influence

Jadi, bagaimana memanfaatkan data social networks?

Guo et al., (2015) incorporates social influence into common scoring function.

Now:



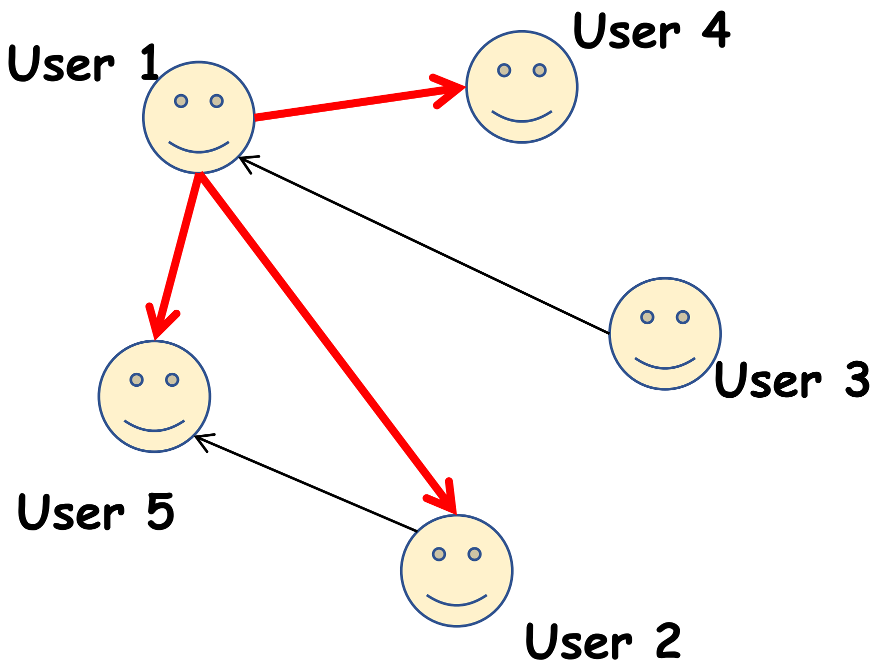
$$\hat{r}_{ui} = \left(p_u + \frac{1}{|N(u)|} \sum_{b \in N(u)} p_b \right) q_i^T$$

$$N(u) = \{p_j | A_{uj} = 1\}$$

Semua orang yang dipercaya (di-follow oleh user u)

Jadi, bagaimana memanfaatkan data social networks?

Guo et al., (2015) incorporates social influence into common scoring function.



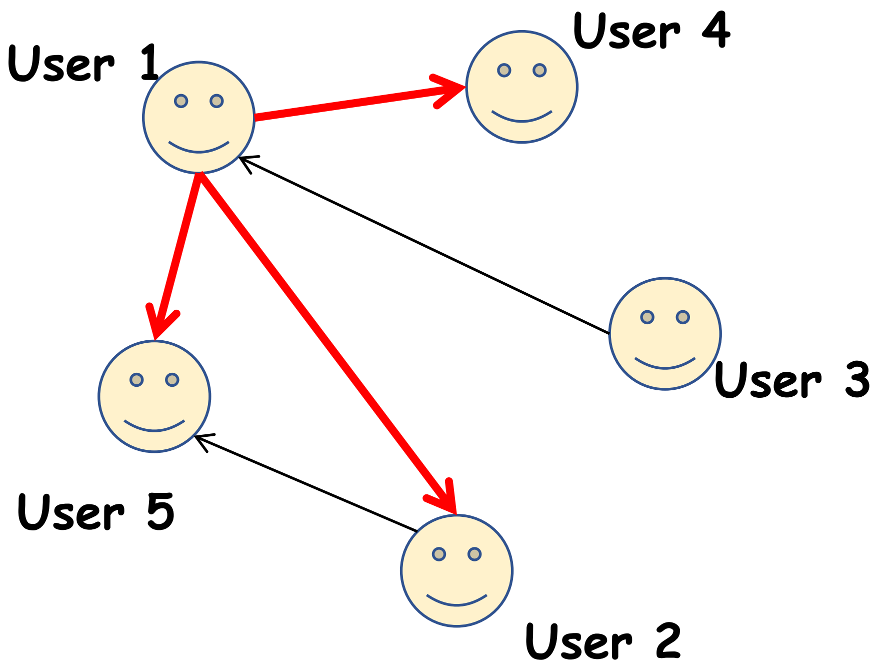
Contoh:

$$\hat{r}_{1,i} = \left(p_1 + \frac{p_2 + p_4 + p_5}{3}\right) q_i^T$$

Embedding untuk user 1 merupakan kombinasi antara embedding original user 1 dan embedding dari user lain yang dipercaya (yang di-follow) oleh user 1.

Problem?

This method is just for modeling the **first-order neighbors** of each user. In reality, a user can be influenced by their **friend of friend of friend of friend of …**



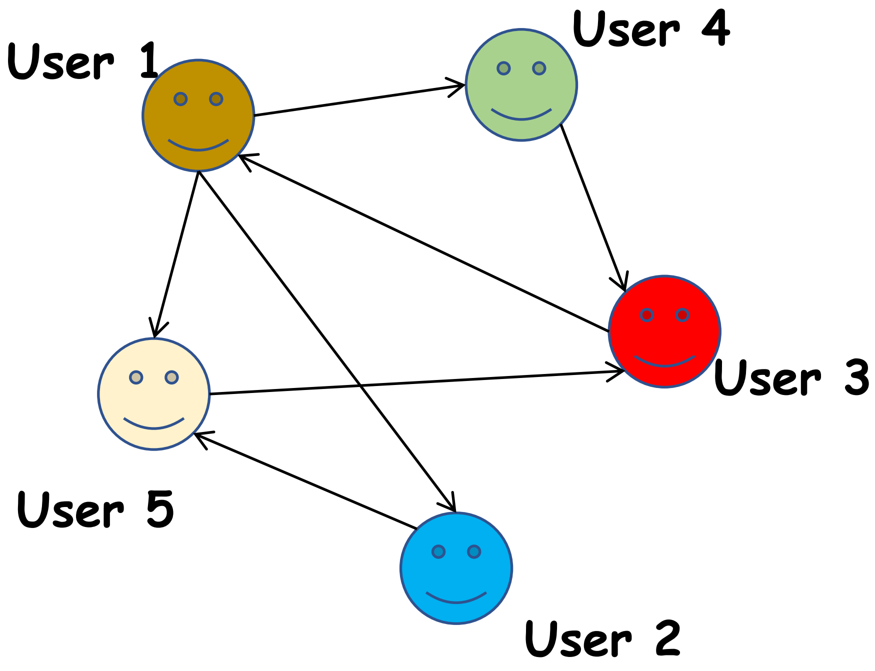
Contoh:

$$\hat{r}_{1,i} = \left(p_1 + \frac{p_2 + p_4 + p_5}{3}\right) q_i^T$$

Embedding untuk user 1 merupakan kombinasi antara embedding original user 1 dan embedding dari user lain yang dipercaya (yang di-follow) oleh user 1.

More General Approach: Graph Neural Networks

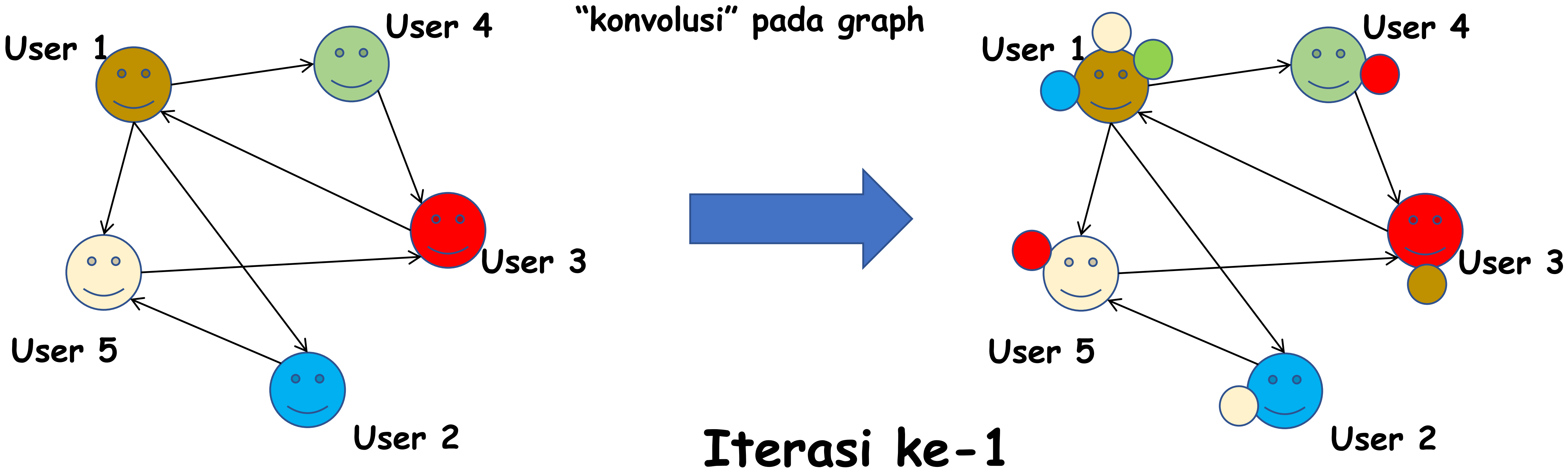
Kita dapat menggunakan beberapa **Graph Neural Networks** untuk menghasilkan **user embeddings** dari sebuah social networks.



Iteratively aggregate feature information from neighbors and integrate the aggregated information with the current central node representation (Wu et al., 2021).

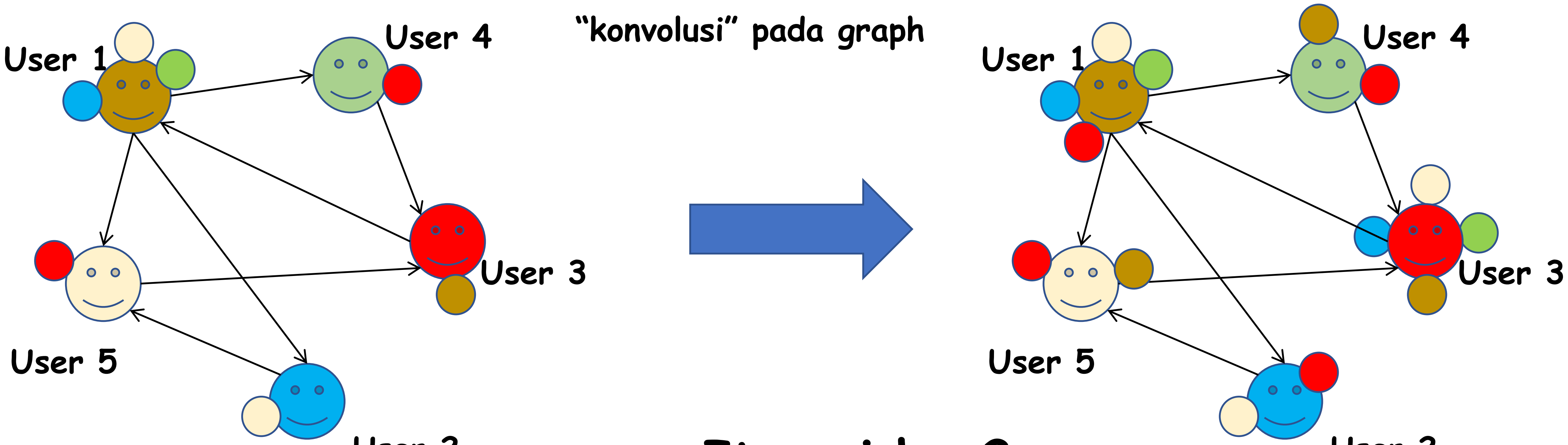
More General Approach: Graph Neural Networks

Kita dapat menggunakan beberapa **Graph Neural Networks** untuk menghasilkan **user embeddings** dari sebuah social networks.



More General Approach: Graph Neural Networks

Kita dapat menggunakan beberapa **Graph Neural Networks** untuk menghasilkan **user embeddings** dari sebuah social networks.



Iterasi ke-2

Graph Neural Networks

Untuk semua tetangga dari **u**

Secara umum, ada dua operasi:

1. **Aggregation**: bagaimana menggabungkan informasi dari tetangga?
2. **Update**: update representasi vector dari **central node** dengan menggabungkan vector saat ini dengan vector hasil gabungan para tetangganya.

$$n_u^{(l)} = Aggregator_l\left(\left\{h_k^{(l)} | \forall k \in N(u)\right\}\right)$$

$$h_u^{(l+1)} = Updater_l\left(h_u^{(l)}, n_u^{(l)}\right)$$

Vector representation of node u at (l+1)-th layer

Vector representation of node u at l-th layer

Graph Convolutional Networks (GCN) (Kipf & Welling, 2017)

UNIVERSITAS INDONESIA
FAKULTAS ILMU KOMPUTER

Simple Version

$$n_u^{(l)} = Aggregator_l\left(\left\{h_k^{(l)} | \forall k \in N(u)\right\}\right) = \sum_{k \in N(u)} \tilde{A}_{u,k}\, h_k^{(l)}$$

Adjacency matrix + Identity matrix supaya ada self-loop

$$\tilde{A} = A + I$$

$$\tilde{A}_{u,u} = 1$$
$$\tilde{A}_{u,k} = A_{u,k} \quad , u \neq k$$

$$h_u^{(l+1)} = Updater_l\left(h_u^{(l)}, n_u^{(l)}\right) = ReLU\left(W^{(l)} n_u^{(l)}\right)$$

Trainable transformation matrix

Graph Convolutional Networks (GCN) (Kipf & Welling, 2017)

Hang On …

Representasi vektor dari semua node akan terus di-update setiap kali masuk ke layer "konvolusi" berikutnya. Namun, untuk yang pertama kali, isinya apa ya?

$$h^{(0)} = ?$$

Graph Convolutional Networks (GCN) (Kipf & Welling, 2017)

Hang On …

Representasi vektor dari semua node akan terus di-update setiap kali masuk ke layer "konvolusi" berikutnya. Namun, untuk yang pertama kali, isinya apa ya?

$$h^{(0)} = X$$

Isinya adalah **feature vector** dari node tersebut. Bisa memanfaatkan beberapa node properies seperti **centrality**, **clustering coefficient**, dsb.

Graph Convolutional Networks (GCN) (Kipf & Welling, 2017)

Hang On …

Representasi vektor dari semua node akan terus di-update setiap kali masuk ke layer "konvolusi" berikutnya. Namun, untuk yang pertama kali, isinya apa ya?

$$h^{(0)} = I$$

Namun, jika sama sekali tidak ada informasi tentang fitur-nya, yang paling sederhana bisa kita set awalnya dengan **identity matrix**.

Animasi

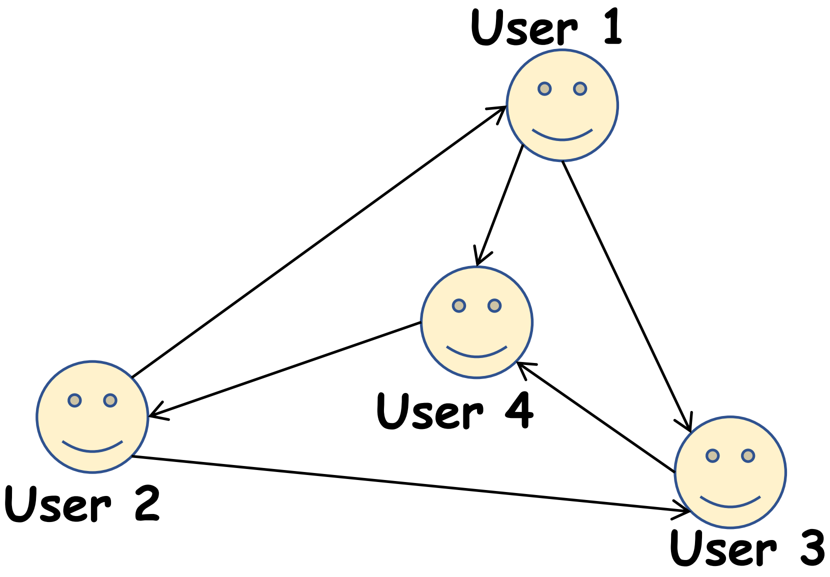
Alfan F. Wicaksono, Fasilkom UI @ 2022

Graph Convolutional Networks (GCN) (Kipf & Welling, 2017)

Simple Version

$$n_u^{(l)} = Aggregator_l\left(\left\{h_k^{(l)} | \forall k \in N(u)\right\}\right) = \sum_{k \in N(u)} \tilde{A}_{u,k}\, h_k^{(l)}$$

Consider the following illustration:



$$A = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

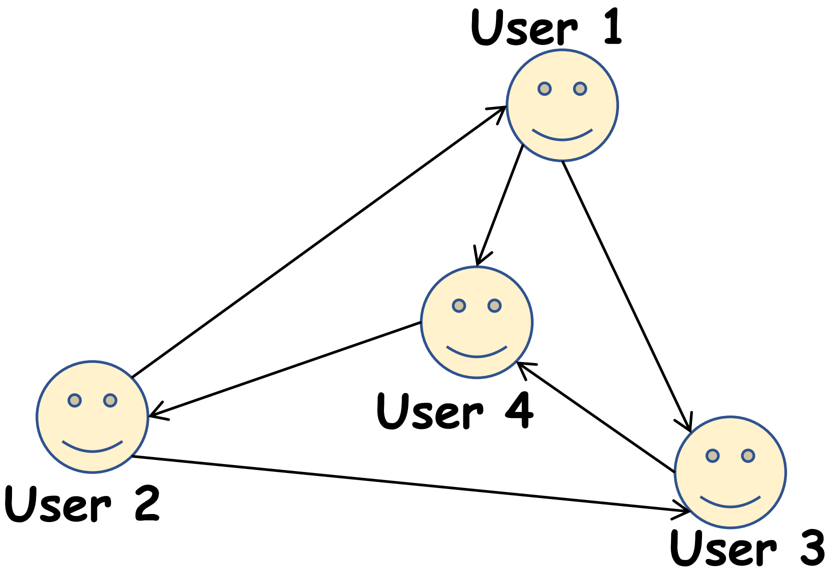
Alfan F. Wicaksono, Fasilkom UI @ 2022

Graph Convolutional Networks (GCN) (Kipf & Welling, 2017)

Simple Version

$$n_u^{(l)} = Aggregator_l\left(\left\{h_k^{(l)} | \forall k \in N(u)\right\}\right) = \sum_{k \in N(u)} \tilde{A}_{u,k}\, h_k^{(l)}$$

Consider the following illustration:



$$A = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \tilde{A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Animasi

Graph Convolutional Networks (GCN) (Kipf & Welling, 2017)

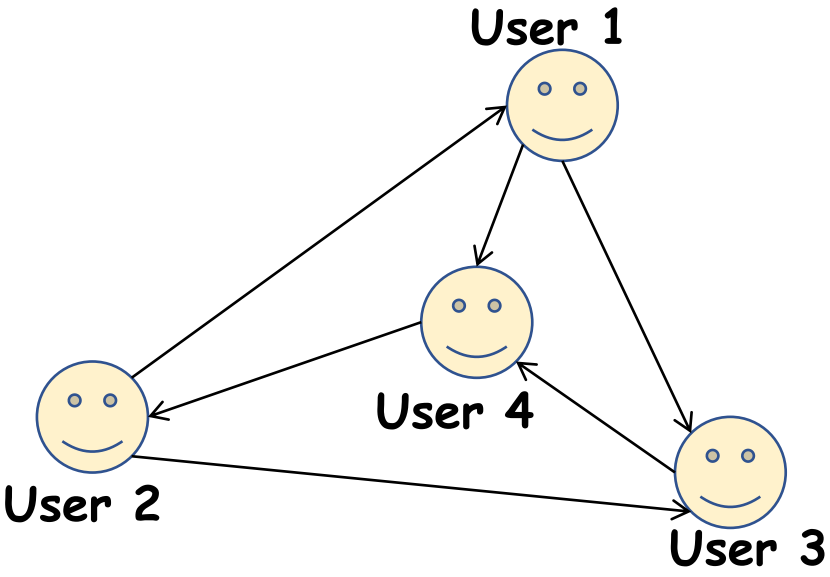
Misal, kita pilih ukuran embedding vector untuk masing-masing user adalah 4.

Simple Version

$$n_u^{(l)} = Aggregator_l\left(\left\{h_k^{(l)} | \forall k \in N(u)\right\}\right) = \sum_{k \in N(u)} \tilde{A}_{u,k}\, h_k^{(l)}$$

Layer 1 – fase aggr.

Consider the following illustration:



$$\underbrace{\begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}}_{\tilde{A}} \times \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{h^{(0)}} = \underbrace{\begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}}_{n^{(0)}}$$

Kita pilih I

Animasi

Kita asumsikan W^0 sudah dilatih.

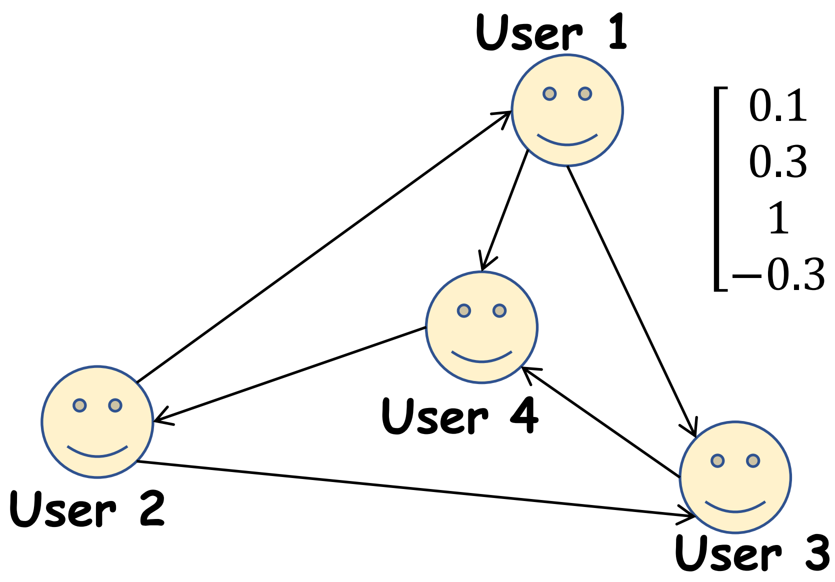
Graph Convolutional Networks (GCN) (Kipf & Welling, 2017)

Simple Version

Layer 1 – fase updat.

$$h_u^{(l+1)} = Updater_l\left(h_u^{(l)}, n_u^{(l)}\right) = ReLU\left(W^{(l)} n_u^{(l)}\right)$$

Consider the following illustration:



$$\underbrace{\begin{bmatrix} 0.1 & 0 & 0.1 & 1.4 \\ 0.3 & -0.1 & 1.3 & 0 \\ 1 & 0.2 & 0.03 & 0.76 \\ -0.3 & 1 & -0.6 & 1 \end{bmatrix}}_{W^{(0)}} \times \underbrace{\begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}}_{n^{(0)}} = \begin{bmatrix} 0.2 & 1.4 & 0.2 & 1.6 \\ 1.5 & -0.1 & 1.5 & 1.6 \\ 1.23 & 0.96 & 1.23 & 1.79 \\ 0.1 & 2 & 0.1 & 0.1 \end{bmatrix}$$

$$\xrightarrow{\text{ReLU}} \underbrace{\begin{bmatrix} 0.2 & 1.4 & 0.2 & 1.6 \\ 1.5 & 0 & 1.5 & 1.6 \\ 1.23 & 0.96 & 1.23 & 1.79 \\ 0.1 & 2 & 0.1 & 0.1 \end{bmatrix}}_{h^{(1)}}$$

Animasi

Alfan F. Wicaksono, Fasilkom UI @ 2022

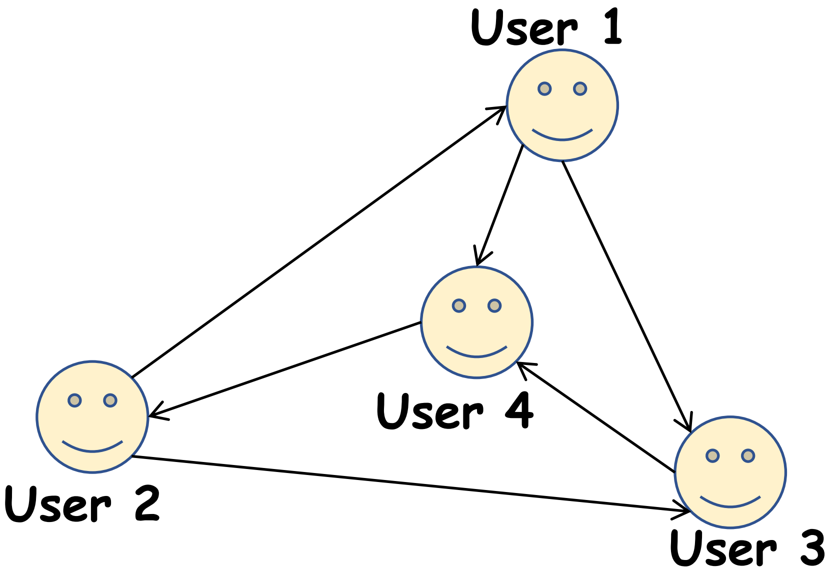
Graph Convolutional Networks (GCN) (Kipf & Welling, 2017)

Simple Version

$$n_u^{(l)} = Aggregator_l\left(\left\{h_k^{(l)} | \forall k \in N(u)\right\}\right) = \sum_{k \in N(u)} \tilde{A}_{u,k}\, h_k^{(l)}$$

Layer 2 – fase aggr.

Consider the following illustration:



$$\underbrace{\begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}}_{\tilde{A}} \times \underbrace{\begin{bmatrix} 0.2 & 1.4 & 0.2 & 1.6 \\ 1.5 & 0 & 1.5 & 1.6 \\ 1.23 & 0.96 & 1.23 & 1.79 \\ 0.1 & 2 & 0.1 & 0.1 \end{bmatrix}}_{h^{(1)}} = \underbrace{\begin{bmatrix} 1.53 & 4.36 & 1.53 & 3.49 \\ 2.93 & 2.36 & 2.93 & 4.99 \\ 1.53 & 4.36 & 1.53 & 3.49 \\ 1.6 & 2 & 1.6 & 1.7 \end{bmatrix}}_{n^{(1)}}$$

Animasi

Kita asumsikan W^1 sudah dilatih.

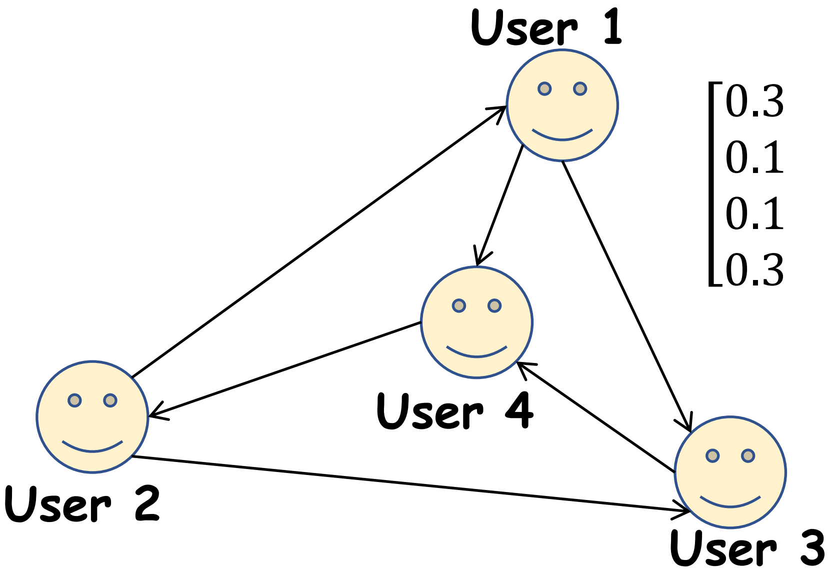
Graph Convolutional Networks (GCN) (Kipf & Welling, 2017)

Simple Version

Layer 2 – fase updat.

$$h_u^{(l+1)} = Updater_l\left(h_u^{(l)}, n_u^{(l)}\right) = ReLU\left(W^{(l)} n_u^{(l)}\right)$$

Consider the following illustration:



$$\underbrace{\begin{bmatrix} 0.3 & -0.4 & 0.12 & 0.5 \\ 0.1 & -0.1 & 1.2 & 0 \\ 0.1 & 0.2 & 0.01 & 0 \\ 0.3 & 1.2 & -0.7 & 0 \end{bmatrix}}_{W^{(1)}} \times \underbrace{\begin{bmatrix} 1.53 & 4.36 & 1.53 & 3.49 \\ 2.93 & 2.36 & 2.93 & 4.99 \\ 1.53 & 4.36 & 1.53 & 3.49 \\ 1.6 & 2 & 1.6 & 1.7 \end{bmatrix}}_{n^{(1)}} = \begin{bmatrix} 0.27 & 1.89 & 0.27 & 0.32 \\ 1.69 & 5.43 & 1.70 & 4.04 \\ 0.75 & 0.95 & 0.75 & 1.38 \\ 2.90 & 1.08 & 2.90 & 4.59 \end{bmatrix}$$

$$\downarrow \text{ReLU}$$

$$\underbrace{\begin{bmatrix} 0.27 & 1.89 & 0.27 & 0.32 \\ 1.69 & 5.43 & 1.70 & 4.04 \\ 0.75 & 0.95 & 0.75 & 1.38 \\ 2.90 & 1.08 & 2.90 & 4.59 \end{bmatrix}}_{h^{(2)}}$$

Dan seterusnya …

Graph Convolutional Networks (GCN) (Kipf & Welling, 2017)

Simple Version

$$n_u^{(l)} = Aggregator_l\left(\left\{h_k^{(l)} | \forall k \in N(u)\right\}\right) = \sum_{k \in N(u)} \tilde{A}_{u,k}\, h_k^{(l)}$$

$$h_u^{(l+1)} = Updater_l\left(h_u^{(l)}, n_u^{(l)}\right) = ReLU\left(W^{(l)} n_u^{(l)}\right)$$

Trainable transformation matrix

Adjacency matrix tidak dinormalisasi. Ini bisa berbahaya: **vanishing/exploding gradient problem.**

Graph Convolutional Networks (GCN) (Kipf & Welling, 2017)

Original Version

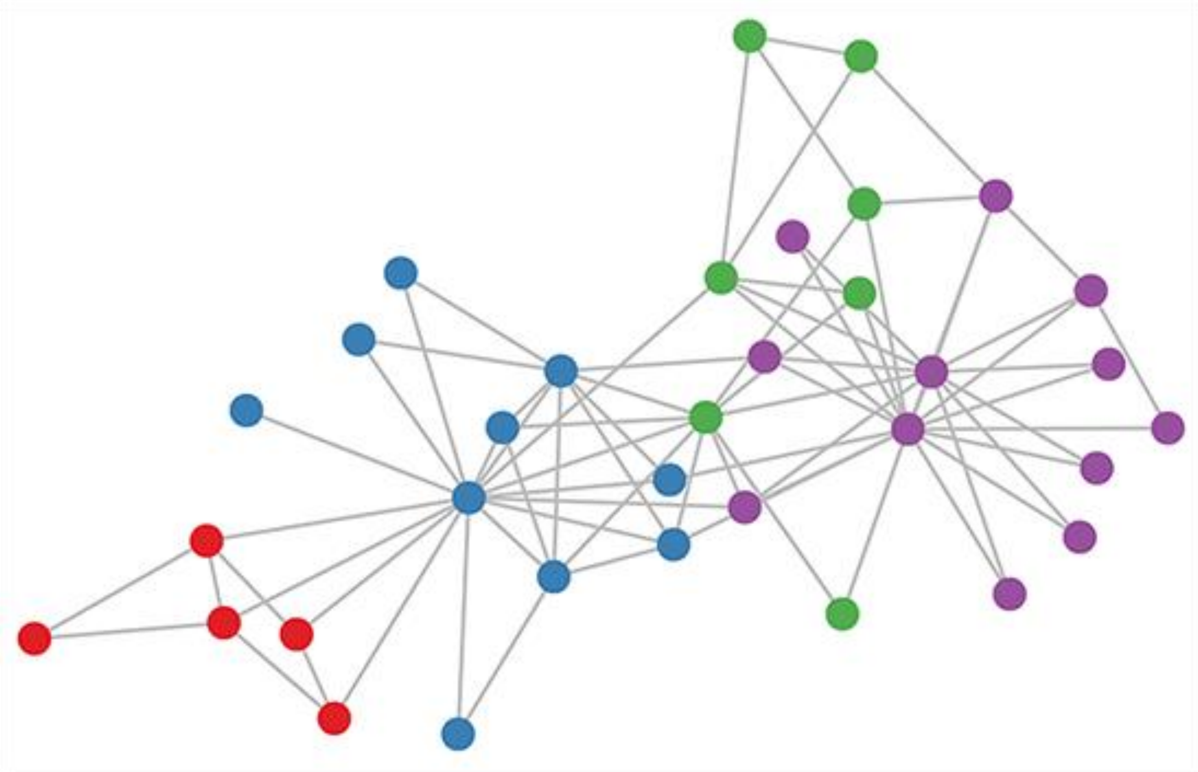
$$n_u^{(l)} = Aggregator_l\left(\left\{h_k^{(l)} \middle| \forall k \in N(u)\right\}\right) = \sum_{k \in N(u)} \underbrace{\tilde{d}^{-\frac{1}{2}}{}_{u,u}\ \tilde{A}_{u,k}\ \tilde{d}^{-\frac{1}{2}}{}_{k,k}}_{\text{Untuk normalisasi adjacency matrix } A} h_k^{(l)}$$

$$h_u^{(l+1)} = Updater_l\left(h_u^{(l)}, n_u^{(l)}\right) = ReLU\left(W^{(l)} n_u^{(l)}\right)$$

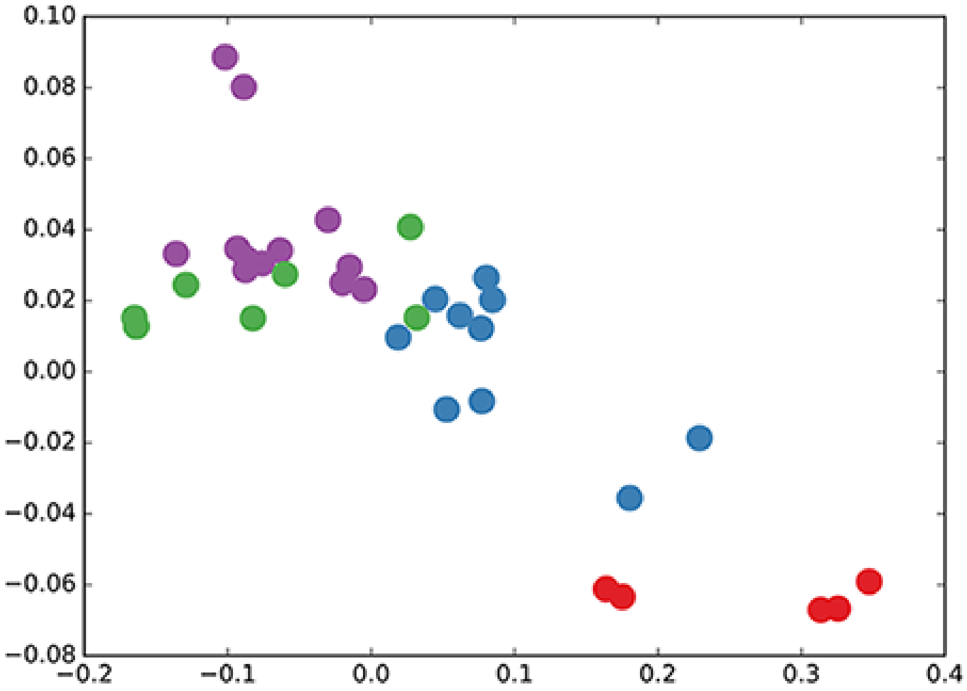
Degree dari user j $d_{j,j} = \sum_k \tilde{A}_{j,k}$

Graph Convolutional Networks (GCN) (Kipf & Welling, 2017)

Hasil dari 3-layer GCN (3 iterations)

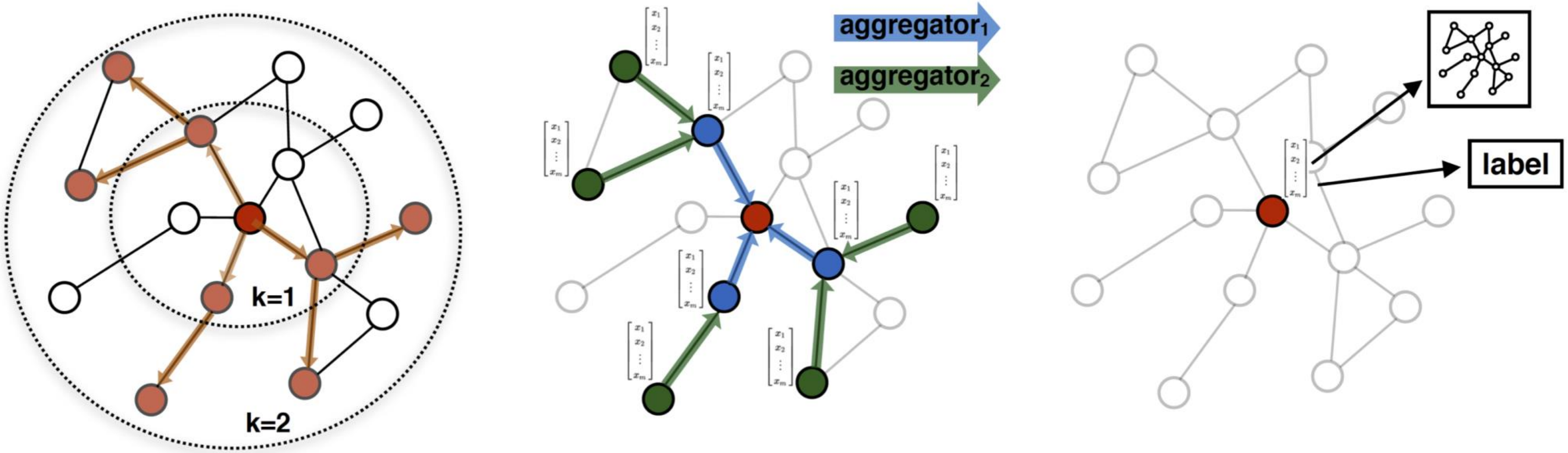


Karate Network (Brandes et al., 2008)



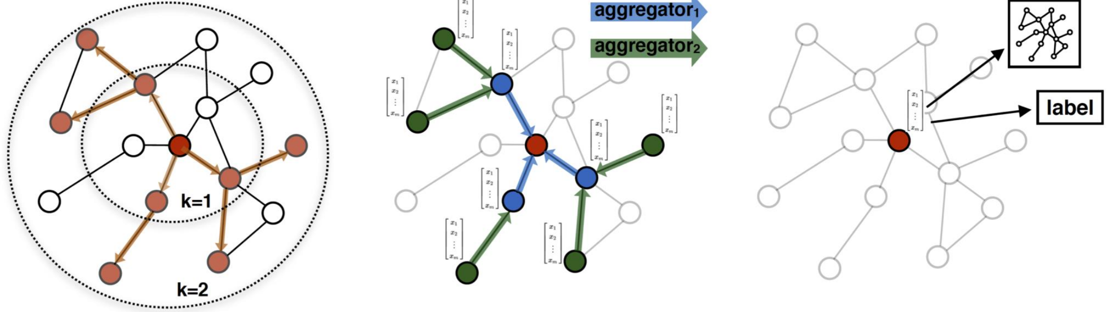
GraphSAGE (Hamilton et al., NIPS 2017)

- A generalization of GCN
- Addressing scalability --> **don't use all neighbors!** Just a random sample of them



1. Sample neighborhood
2. Aggregate feature information from neighbors
3. Predict graph context and label using aggregated information

GraphSAGE (Hamilton et al., NIPS 2017)



1. Sample neighborhood
2. Aggregate feature information from neighbors
3. Predict graph context and label using aggregated information

$$n_u^{(l)} = Aggregator_l\left(\left\{h_k^{(l)} | \forall k \in N'(u)\right\}\right)$$

Mean, LSTM, Max Pooling, …
Banyak opsi

A fixed-size, uniform draw from the set of all neighbors.

Untuk setiap iterasi, kita menggunakan sample yang berbeda-beda.

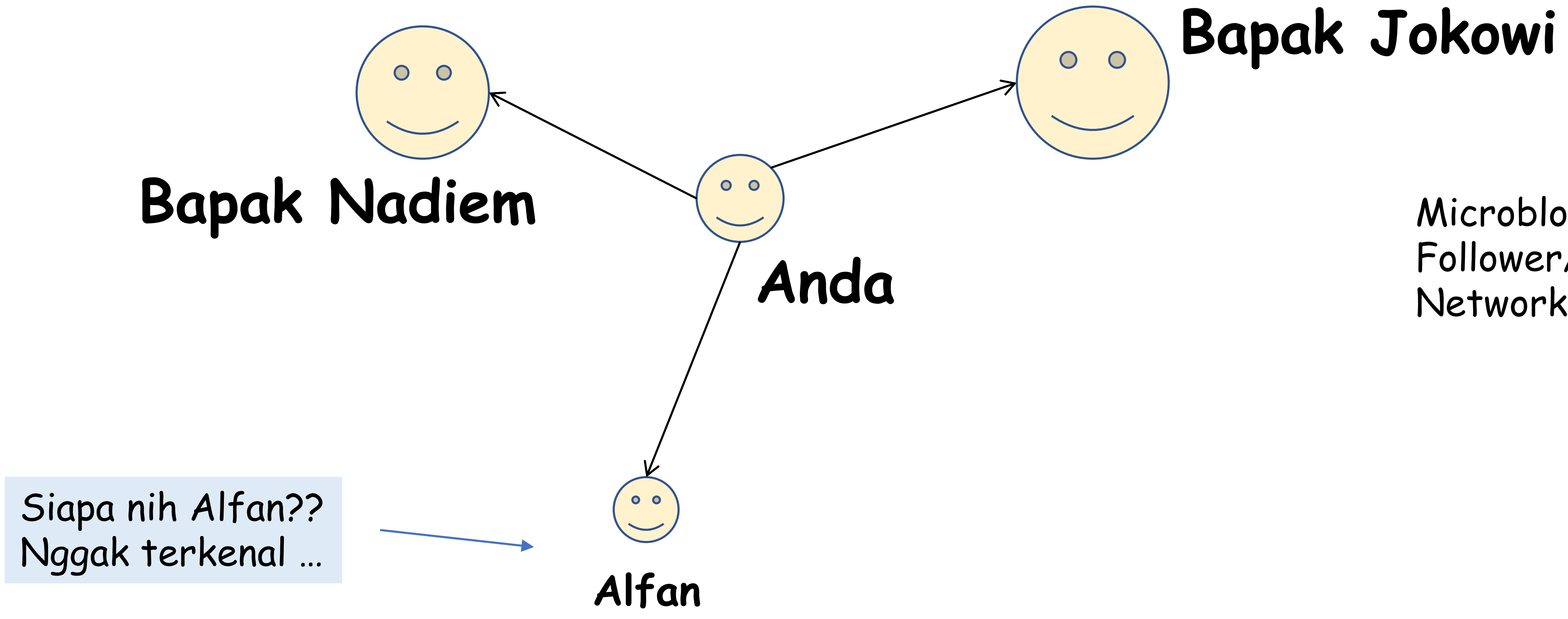
$$h_u^{(l+1)} = Updater_l\left(h_u^{(l)}, n_u^{(l)}\right) = \delta\left(W^{(l)} \cdot \left[h_u^{(l)} \oplus n_u^{(l)}\right]\right)$$

Any non-linear activation function

Concatenation layer

Graph Attention Networks (GAT) (Velickovic et al., 2017)

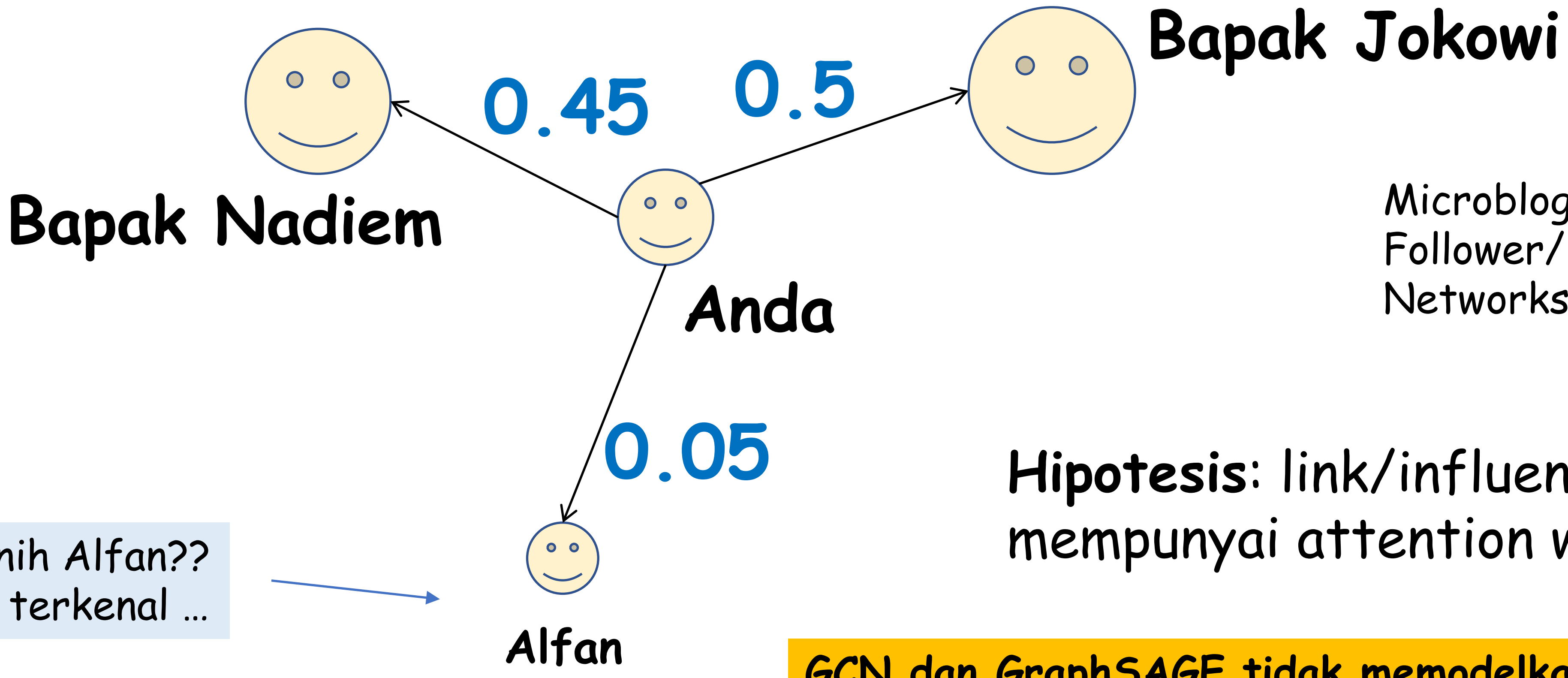
Do you think all neighbors have the same "**weights**"?



Microblog's Follower/Followee Networks

Graph Attention Networks (GAT) (Velickovic et al., 2017)

Do you think all neighbors have the same "**weights**"?



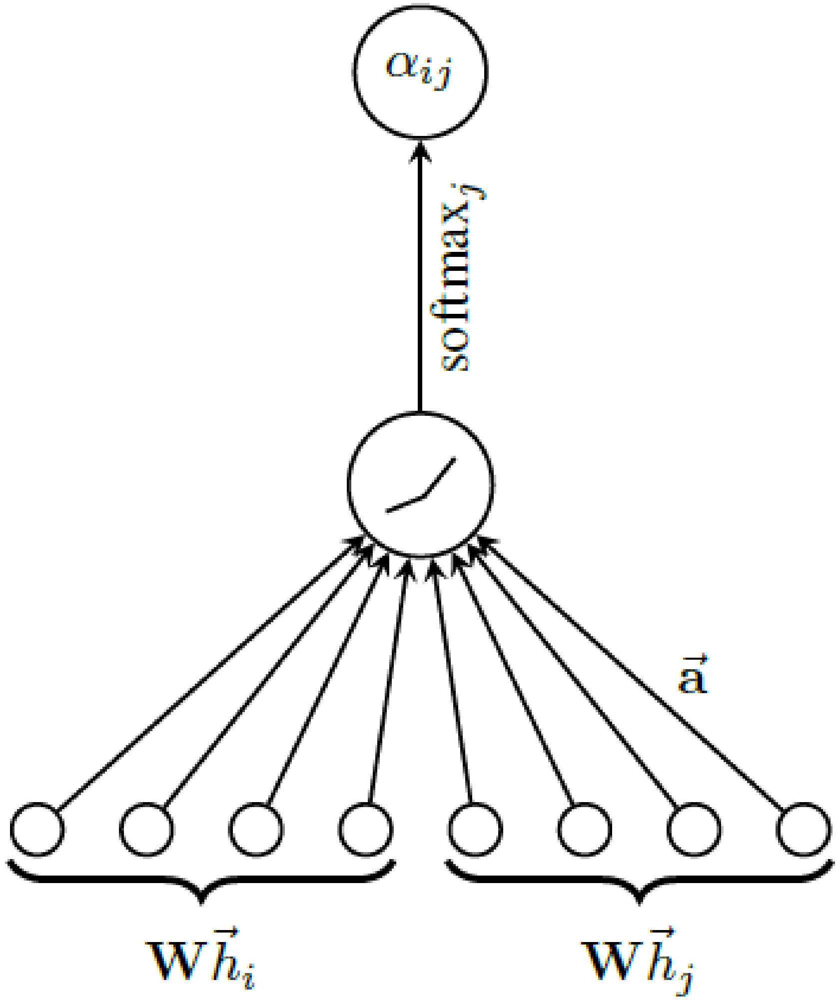
Microblog's Follower/Followee Networks

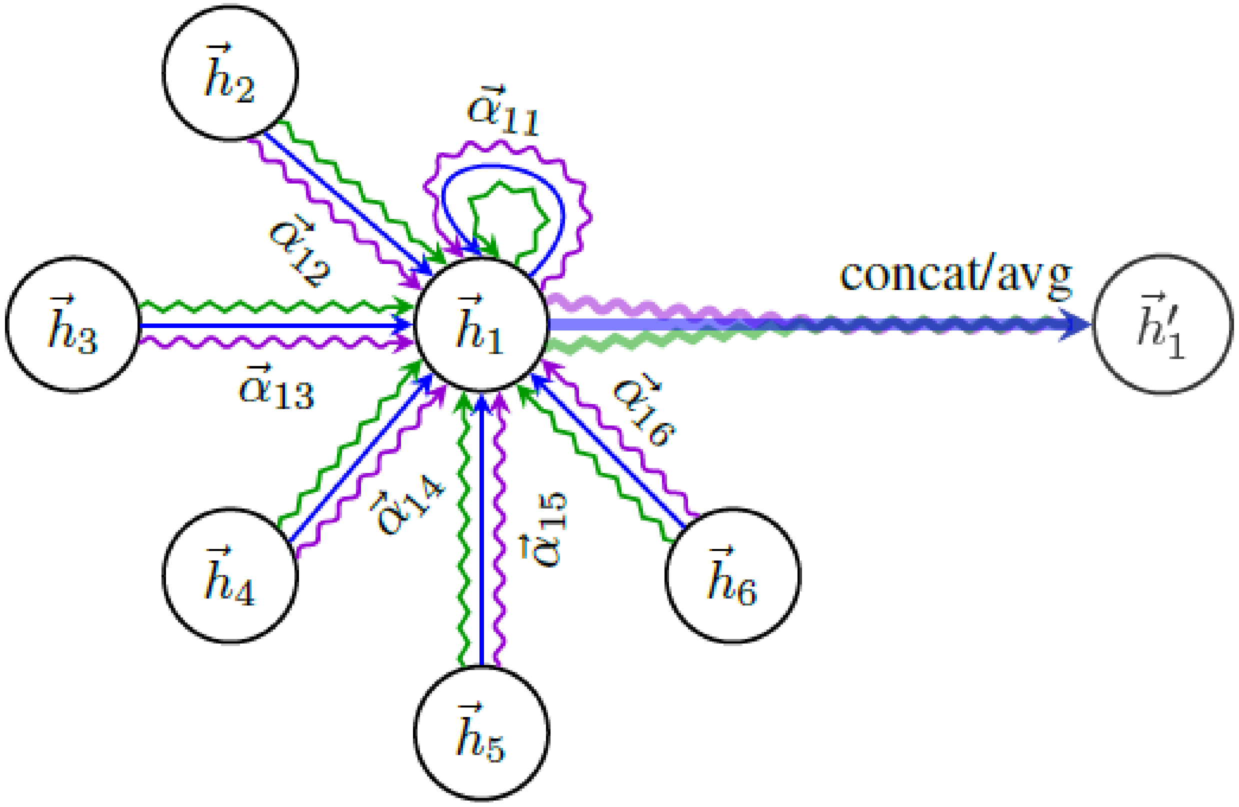
Siapa nih Alfan??
Nggak terkenal ...

Hipotesis: link/influence mempunyai attention weight

GCN dan GraphSAGE tidak memodelkan hal ini

Graph Attention Networks (GAT) (Velickovic et al., 2017)

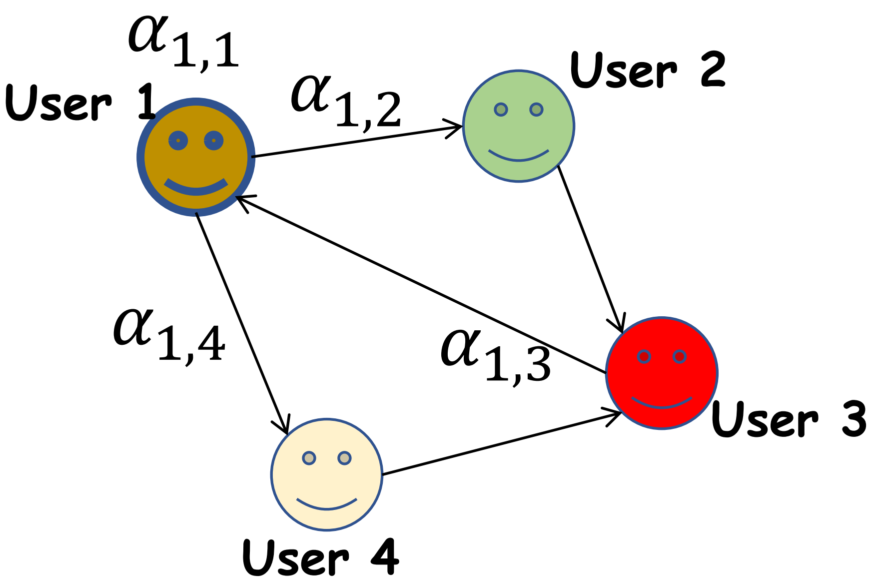




Graph Attention Networks (GAT) (Velickovic et al., 2017)

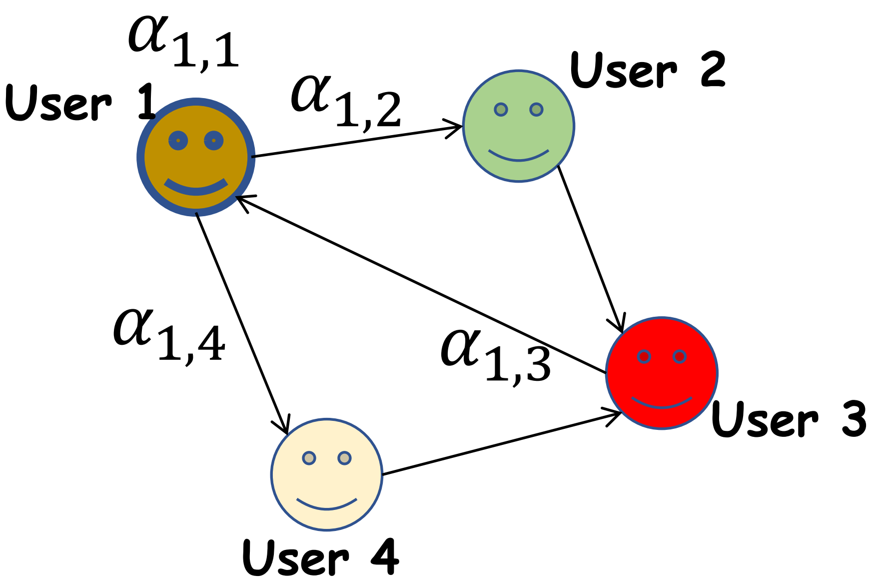
$$n_u^{(l)} = Aggregator_l\left(\left\{h_k^{(l)} | \forall k \in N(u)\right\}\right) = \sum_{k \in N(u)} \alpha_{u,k} \cdot h_k^{(l)}$$

$$h_u^{(l+1)} = Updater_l\left(h_u^{(l)}, n_u^{(l)}\right) = \delta\left(W^{(l)} n_u^{(l)}\right)$$



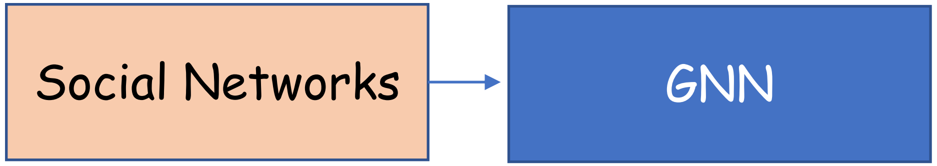
Graph Attention Networks (GAT) (Velickovic et al., 2017)

$$\alpha_{u,j} = \frac{exp\left(leakyReLU\left(a^T\left[W^{(l)}h_u^{(l)} \oplus W^{(l)}h_j^{(l)}\right]\right)\right)}{\sum_{k \in N(u)} exp\left(leakyReLU\left(a^T\left[W^{(l)}h_u^{(l)} \oplus W^{(l)}h_k^{(l)}\right]\right)\right)}$$

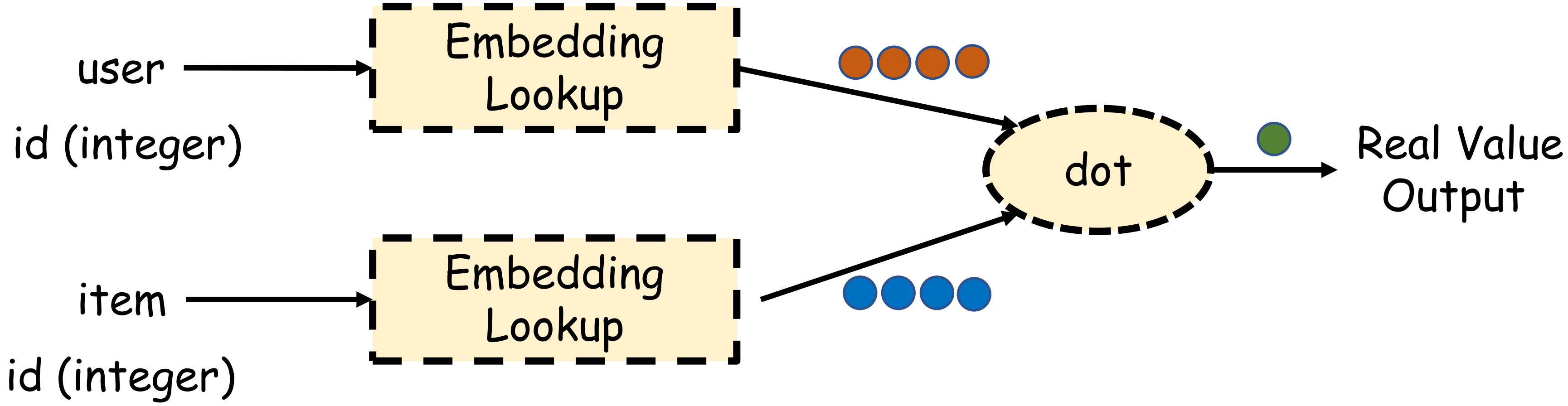


Bagaimana GCN, GraphSAGE, atau GAT diterapkan di model kita?

Salah satu cara



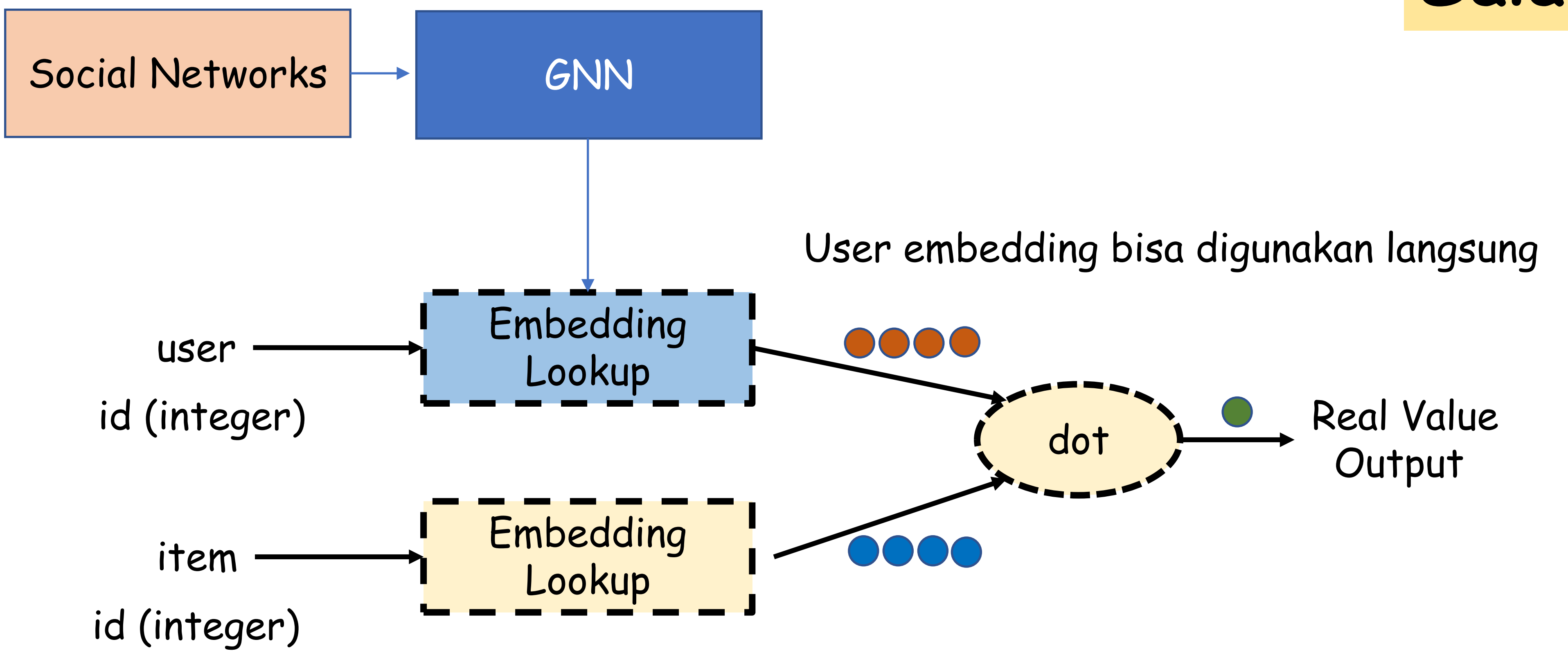
GNN (GCN, GraphSAGE, GAT) dilatih terlebih dahulu secara unsupervised (**pre-training**)



Jika output adalah real value

Bagaimana GCN, GraphSAGE, atau GAT diterapkan di model kita?

Salah satu cara



Jika output adalah real value

Terima Kasih