Matrix Factorization for Recommendation

Advanced Techniques

Klaudia Balcer

Computational Intelligence Research Group, Institute of Computer Science

Contents

Introduction

Recap – Recommender Systemts

We are given:

- a set of N items $\mathcal{I} = \{i_1, i_2, \ldots, i_N\}$,
- a set of M users $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$,
- historical data $\mathcal{D}$ consisting of user-item interactions:
  - $\mathcal{D} = \{(u, i, r, t) :$ user u interacted with item i at time stamp t with rating $r\}$,
  - $\mathcal{D} = \{(u, i, r) :$ user u interacted with item i with rating $r\}$,
  - $\mathcal{D} = \{(u, i, t) :$ user u interacted with item i at time stamp $t\}$,
  - $\mathcal{D} = \{(u, i) :$ user u interacted with item $i\}$,
- In matrix approaches we will use the notation r_{ui} for the rating of user u for item i and R for the matrix of all ratings.

The task of a Recommender System (RS) is to provide a list of K recommended items that meet the user's interest based on historical behavior of this and other users.

Recap – Matrix Factorization

In Matrix Factorization approach:

- $\mathrm{P} \in \mathbb{R}^{M \times D}$ - user embeddings ($\mathrm{p_u}$ - vector corresponding to user u);
- $\mathrm{Q} \in \mathbb{R}^{N \times D}$ - item embeddings ($\mathrm{q_i}$ - vector corresponding to item i)

We optimize them so as to obtain:

- $r_{ui} \approx \mathrm{p_u} \times \mathrm{q_i}$,
- $\mathrm{R} \approx \mathrm{P} \times \mathrm{Q}^\top$.

To serve recommendations:

- calculate the scores for all items,
- omit items from training data,
- order ratings based on their predicted ratings / scores,
- we recommend K items with highest predicted ratings / scores.

Loss functions for MF

Loss functions for MF

Probabilistic MF

Probabilitic Matrix Factorization

- User prior: $\mathbb{P}(\mathrm{P}) = \prod_u \mathcal{N}(\mathrm{p}_u|0, \sigma_U^2 I)$
- Item prior: $\mathbb{P}(\mathrm{Q}) = \prod_i \mathcal{N}(\mathrm{q}_i|0, \sigma_I^2 I)$
- Rating probability: $R_{i,j} \sim \mathcal{N}(.|\mathrm{p_u} \times \mathrm{q_i}, \sigma^2)$,
- $\mathbb{1}_{ui}$ - indicates if user i has interacted with item j
- Likelihood:

$$p(R|\mathrm{P}, \mathrm{Q}, \sigma^2) = \prod_{u,i} \left[\mathcal{N}(r_{ui}|\mathrm{p_u} \times \mathrm{q_i}, \sigma^2)\right]^{\mathbb{1}_{ui}}$$

- Posterior:

$$p(R|\sigma_U^2, \sigma_V^2, \sigma^2) = \prod_{u,i} \left[\mathcal{N}(r_{ui}|\mathrm{p_u} \times \mathrm{q_i}, \sigma^2)\mathcal{N}(\mathrm{p_u}|0, \sigma_U^2 I)\mathcal{N}(\mathrm{q_i}|0, \sigma_V^2 I)\right]^{\mathbb{1}_{ui}}$$

Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In Proceedings of the 21st International Conference on Neural Information Processing Systems (NIPS'07). Curran Associates Inc., Red Hook, NY, USA, 1257–1264.

Loss function

Maximizing the log-posterior over movie and user features with hyperparameters (i.e. the observation noise variance and prior variances) kept fixed is equivalent to minimizing the sum-of-squared-errors objective function with L_2 regularization terms:

$$E = \frac{1}{2}\sum_{u,i} \mathbb{1}_{ui}(r_{ui} - \mathrm{p_u} \times \mathrm{q_i})^2 + \frac{\lambda_U}{2}\sum_{u} \|\mathrm{p_u}\|^2 + \frac{\lambda_I}{2}\sum_{i} \|\mathrm{q_i}\|^2$$

where $\lambda_U = \sigma^2/\sigma_U^2$, $\lambda_V = \sigma^2/\sigma_V^2$.

Optimization: gradient descent in P and Q.

Scales linearly with number of observations.

Additional Tricks

- Scale Ratings to $[0, 1]$: $t(x) = (x - 1)/(K - 1)$
- Scale Dot-products to $[0, 1]$: $g(x) = 1/(1 + \exp(-x))$

- **Constrained version**:

$$\mathrm{p_u} = \tilde{\mathrm{p}}_u + \frac{\sum_i \mathbb{1}_{ui} W_i}{\sum_i \mathbb{1}_{ui}}$$

where $\tilde{\mathrm{p}}_u$ is the user-specific part and $W \in \mathbb{R}^{D \times M}$ is the item specific part.

Loss functions for MF

MF-BPR

Bayesian Personalized Ranking

Idea:
Provide a ranking of items – total order on all items:

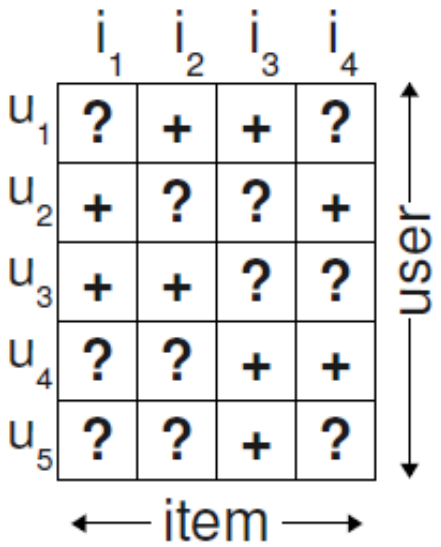
$$\forall i, j \in I : i \neq j \Rightarrow i \geq_u j \vee j \geq_u i \quad \text{(totality)}$$
$$\forall i, j \in I : i \geq_u j \wedge j \geq_u i \Rightarrow i = j \quad \text{(antisymmetry)}$$
$$\forall i, j, k \in I : i \geq_u j \wedge j \geq_u k \Rightarrow i \geq_u k \quad \text{(transitivity)}$$

Intuition: optimize for correctly ranking item pairs instead of scoring single items

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009, June). BPR: Bayesian personalized ranking from implicit feedback.

https://arxiv.org/ftp/arxiv/papers/1205/1205.2618.pdf

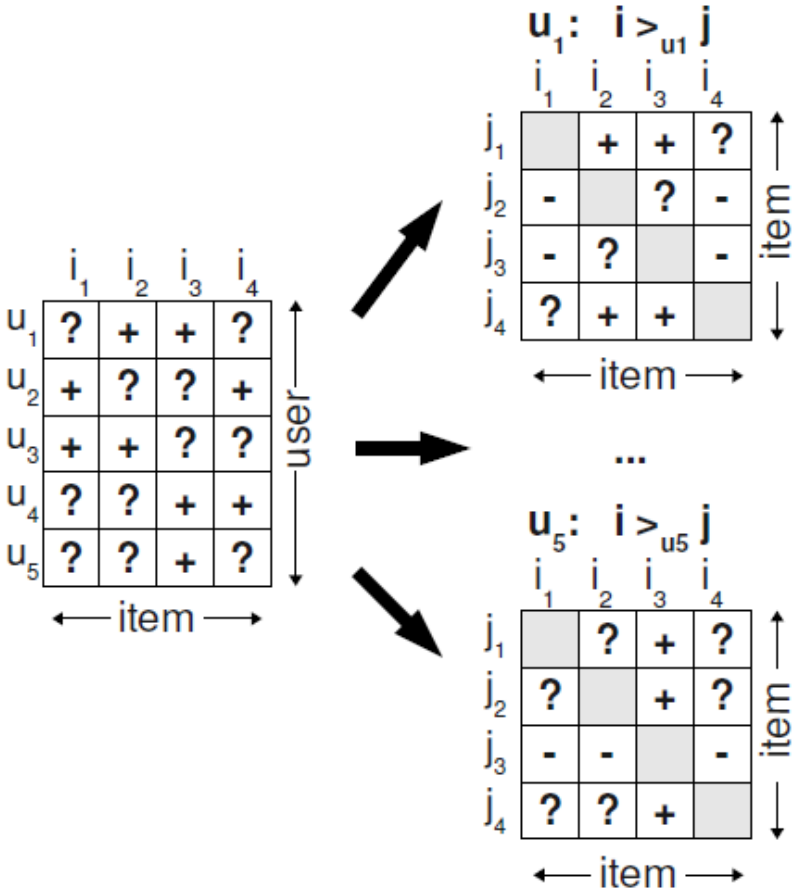
Data Representation:



$$D_S := \{(u, i, j) \mid i \in I_u^+ \wedge j \in I \setminus I_u^+\}$$

triples: $(user, positive, negative)$

Example



Loss function

$$\prod_{u \in U} p(\geq_u |\Theta) = \prod_{(ui,j) \in D_S} p(i \geq_u j|\Theta)$$

$$p(i \geq_u j|\Theta) := \sigma(\hat{x}_{ui,j}(\Theta))$$

$$\begin{aligned} BPR - OPT &:= \ln p(\Theta| \geq_u) \\ &= \dots \\ &= \sum_{(ui,j) \in D_S} \ln \sigma(\hat{x}_{ui,j}) - \lambda_\Theta \|\Theta\|^2 \end{aligned}$$

Loss functions for MF

DirectAU

Uniformity & Alignment

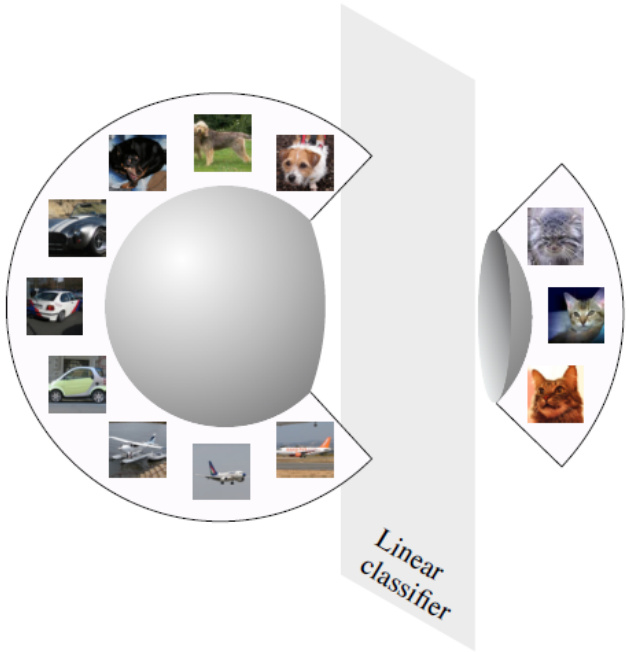


Figure 2: **Hypersphere:** When classes are well-clustered (forming spherical caps), they are linearly separable. The same does not hold for Euclidean spaces.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In Proceedings of the 37th International Conference on Machine Learning (ICML'20), Vol. 119. JMLR.org, Article 921, 9929–9939.

Uniformity & Alignment for RS

$$l_{align} = \mathbb{E}_{(ui)\sim positive} \|f(u) - f(i)\|^2$$

$$l_{uniformity} = \log \mathbb{E}_{(u,u')} \frac{1}{2} e^{-2\|f(u)-f(u')\|^2} + \log \mathbb{E}_{(i,i')} \frac{1}{2} e^{-2\|f(i)-f(i')\|^2}$$

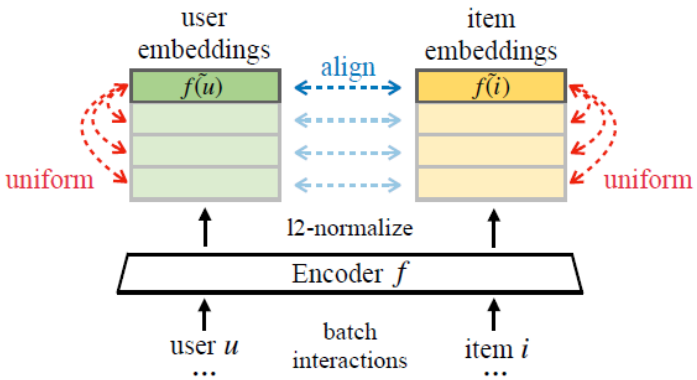
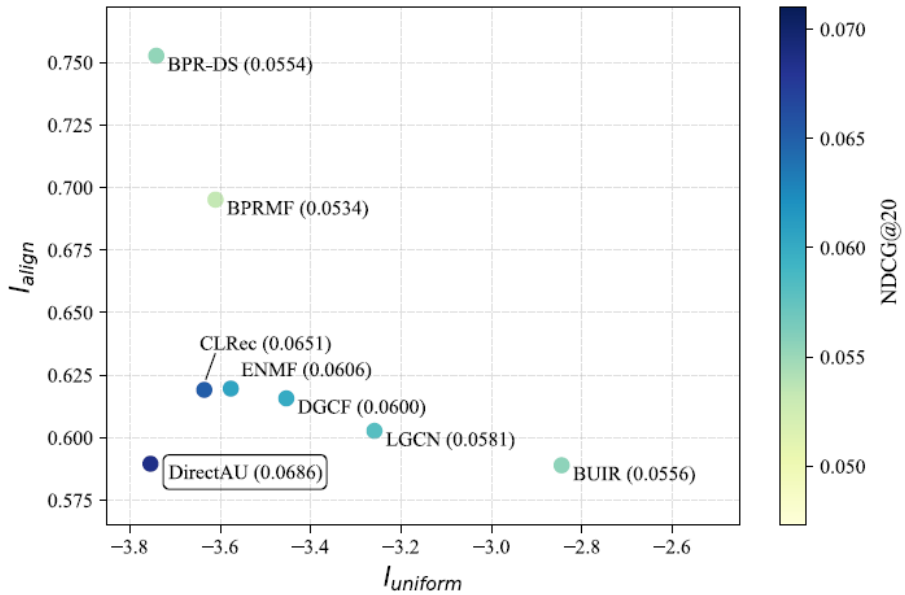


Figure 3: Overview of the proposed DirectAU. We directly optimize 1) representation alignment for positive user-item pairs and 2) in-batch uniformity for users/items.

Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards Representation Alignment and Uniformity in Collaborative Filtering. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22). Association for Computing Machinery, New York, NY, USA, 1816–1825. https://doi.org/10.1145/3534678.3539253

Relation to other metrics



Loss functions for MF

Example

Example

https://colab.research.google.com/drive/1uD5Zo9W1Pb4gtmZ9rKDR2osm56wwwJ0

From Factorization Machines to Neural Approaches

From Factorization Machines to Neural Approaches

Factorization Machines

Factorization Machines

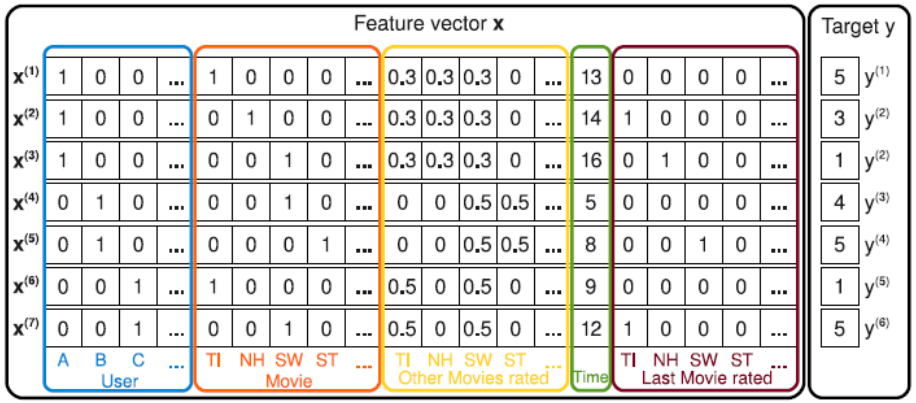


Fig. 1. Example for sparse real valued feature vectors $\mathbf{x}$ that are created from the transactions of example 1. Every row represents a feature vector $\mathbf{x}^{(i)}$ with its corresponding target $y^{(i)}$. The first 4 columns (blue) represent indicator variables for the active user; the next 5 (red) indicator variables for the active item. The next 5 columns (yellow) hold additional implicit indicators (i.e. other movies the user has rated). One feature (green) represents the time in months. The last 5 columns (brown) have indicators for the last movie the user has rated before the active one. The rightmost column is the target – here the rating.

S. Rendle, "Factorization Machines," 2010 IEEE International Conference on Data Mining, Sydney, NSW, Australia, 2010, pp. 995-1000, doi: 10.1109/ICDM.2010.127. keywords: Mathematical model;Support vector machines;Frequency modulation;Predictive models;Data models;Equations;Computational modeling;factorization machine;sparse data;tensor factorization;support vector machine,

FM

1) Model Equation: The model equation for a factorization machine of degree $d = 2$ is defined as:

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i \, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j \qquad (1)$$

where the model parameters that have to be estimated are:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^n, \quad \mathbf{V} \in \mathbb{R}^{n \times k} \qquad (2)$$

And $\langle \cdot, \cdot \rangle$ is the dot product of two vectors of size k:

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle := \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \qquad (3)$$

A row $\mathbf{v}_i$ within $\mathbf{V}$ describes the i-th variable with k factors. $k \in \mathbb{N}_0^+$ is a hyperparameter that defines the dimensionality of the factorization.

A 2-way FM (degree $d = 2$) captures all single and pairwise interactions between variables:

- w_0 is the global bias.
- w_i models the strength of the i-th variable.
- $\hat{w}_{i,j} := \langle \mathbf{v}_i, \mathbf{v}_j \rangle$ models the interaction between the i-th and j-th variable. Instead of using an own model parameter $w_{i,j} \in \mathbb{R}$ for each interaction, the FM models the interaction by factorizing it. We will see later on, that this is the key point which allows high quality parameter estimates of higher-order interactions ($d \geq 2$) under sparsity.

From Factorization Machines to Neural Approaches

LightFM

LightFM Features & Embeddings

$$q_u = \sum_{j \in f_u} e_j^U, \quad b_u = \sum_{j \in f_u} b_j^U$$

$$p_i = \sum_{j \in f_i} e_j^I, \quad b_i = \sum_{j \in f_i} b_j^I$$

$$\hat{r}_{ui} = \sigma(q_u \cdot p_i + b_u + b_i)$$

$$\mathcal{L}(e^U, e^I, b^U, b^I) = \prod_{(ui) \in positive} \hat{r}_{ui} \prod_{(ui) \in negative} (1 - \hat{r}_{ui})$$

If the feature sets consist solely of indicator variables for each user and item, LightFM reduces to the standard MF model. If the feature sets also contain metadata features shared by more than one item or user, LightFM extends the MF model by letting the feature latent factors explain part of the structure of user interactions.

Kula, Maciej. "Metadata embeddings for user and item cold-start recommendations." arXiv preprint arXiv:1507.08439 (2015).

From Factorization Machines to Neural Approaches

Neural Collaborative Filtering

NCF

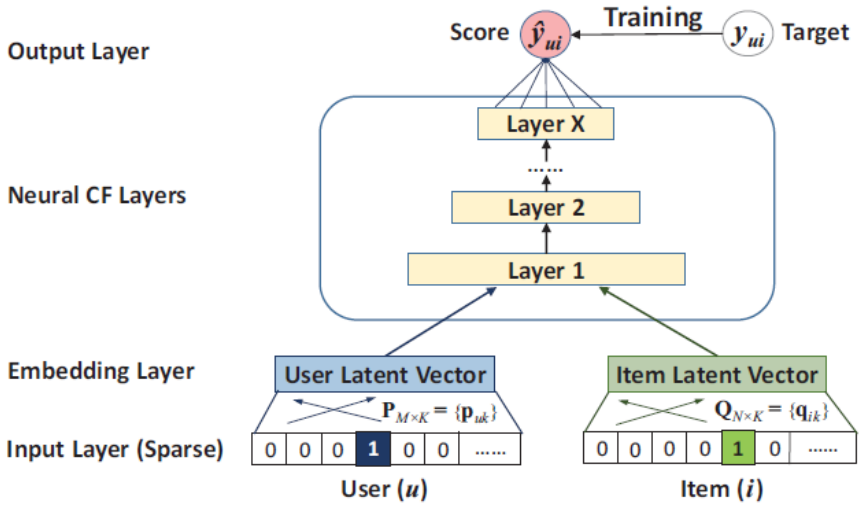


Figure 2: Neural collaborative filtering framework

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web (WWW '17). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. https://doi.org/10.1145/3038912.3052569

NCF Optimization

Note! The authors proposed a different approach to combining user and item embeddings. Interestingly, the use of deep learning has become SOTA for RS, however, scoring is still mostly based on the dot product.
The loss function is binary cross-entropy:

$$\mathcal{L} = \sum_{(ui) \in positive} y_{ui} \log(\hat{y}_{ui}) + \sum_{(ui) \in negative} (1 - y_{ui}) \log(1 - \hat{y}_{ui})$$

NeuMF

Generalized Matrix Factorization – single layer MLP on dot product:

$$\hat{y}_{ui} = \sigma(h^\top (u \circ v))$$

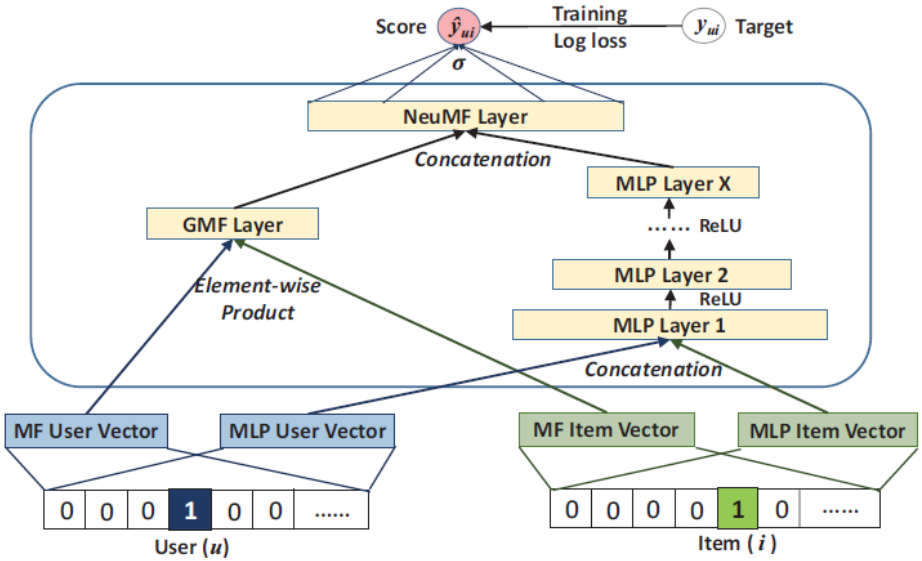


Figure 3: Neural matrix factorization model

MLP: *Tower structure* of layers (half the size in each layer) + ReLU activation.
Note! Both parts are pre-trained separately.

Summary & Further Directions for learning

Methods covered today

1. Introduction

2. Loss functions for MF
   - Probabilistic MF
   - MF-BPR
   - DirectAU
   - Example

3. From Factorization Machines to Neural Approaches
   - Factorization Machines
   - LightFM
   - Neural Collaborative Filtering

4. Summary & Further Directions for learning

Loss functions

Optimization:

- MSE,
- BPR,
- Uniformity & Alignment,
- Binary Cross-Entropy.

Evaluation:

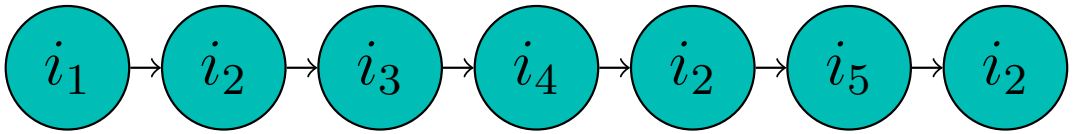
- recall,
- NDCG,
- MRR,
- HR,
- ...

Data representations

- matrix

$$\begin{bmatrix} 0 & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \end{bmatrix}$$

- sequence



- graph