Recommender Systems

Đặng Hoàng Vũ

Ban công nghệ tập đoàn FPT

Giới thiệu

- Recommender systems còn gọi là các hệ khuyến nghị.
- Mục đích nhằm lựa chọn các sản phẩm phù hợp nhất với người dùng, cá nhân hóa trải nghiệm người dùng.
- Được sử dụng rộng rãi trong nhiều lĩnh vực.

Lợi ích

- Cải thiện trải nghiệm người dùng.
- Tối ưu hóa doanh thu qua up-sale, cross-sale…
- Tăng hiệu năng hoạt động bằng tự động hóa.
- Biến khách hàng tiềm năng thành khách hàng thật.
- Hỗ trợ business intelligence.

Tổng quan

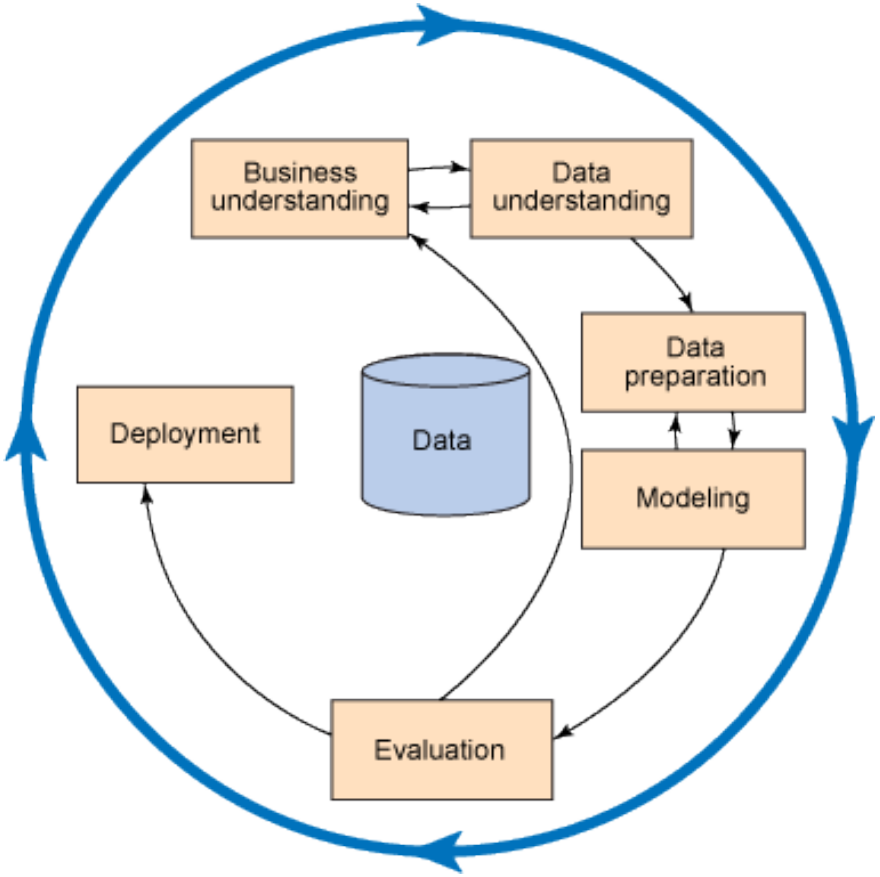
Các thành phần chính:

- Tập hợp người dùng.
- Tập hợp sản phẩm.
- Thông tin về người dùng, sản phẩm.
- Lịch sử giao dịch.

Giả thiết: mức độ phù hợp của mỗi sản phẩm với mỗi người dùng có thể số hóa và mô hình hóa.

Chu trình

- Thu thập dữ liệu.
- Thiết kế feature.
- Xây dựng mô hình.
- Đánh giá mô hình.
- Cải tiến…



Đánh giá recommender systems

Đánh giá offline: sử dụng dữ liệu đã thu thập sẵn.

- Kết quả tốt là kết quả trùng khớp với dữ liệu.
- Có thể sử dụng các độ đo như MAE, RMSE...
- Dữ liệu test có thể cùng hoặc khác loại với dữ liệu để xây dựng mô hình.

Đánh giá online:thử nghiệm mô hình trên thực tế.

- A/B testing nếu có điều kiện.
- Đánh giá bằng các độ đo có ý nghĩa thực tế.

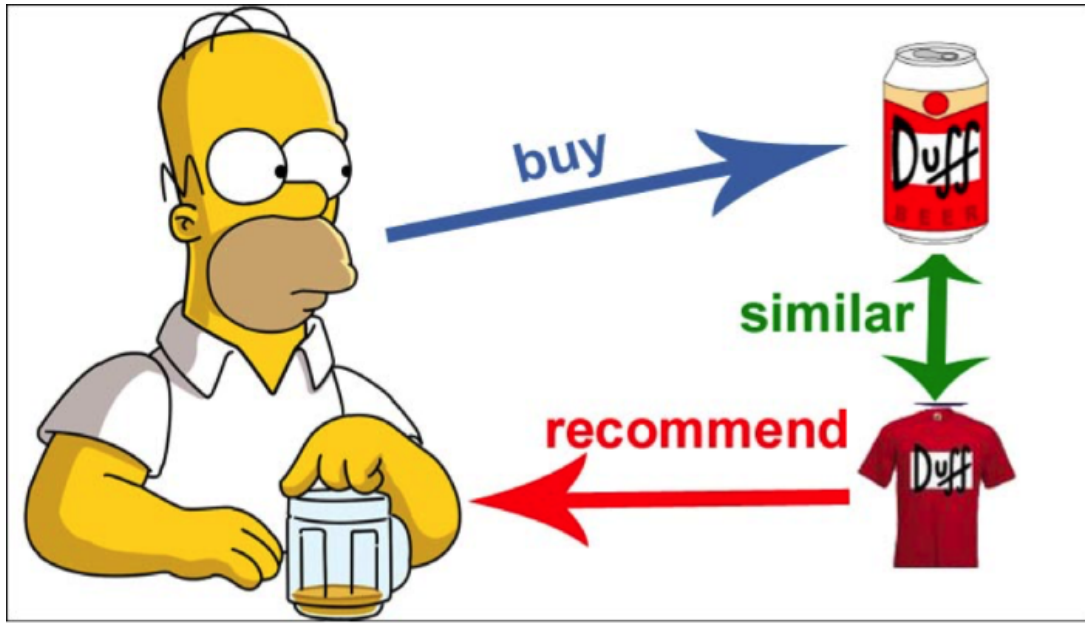
Chú ý: chính xác chưa chắc đã tốt!

Một số kỹ thuật bổ trợ recommendation

- Natural language understanding là một chuyên ngành lớn với rất nhiều ứng dụng. Các kỹ thuật quan trọng cho ứng dụng recommendation có named entity recognition, topic modeling, sentiment analysis…
- Dimensionality reduction phục vụ hai mục đích: giảm khối lượng dữ liệu và giảm độ thưa của dữ liệu. Các kỹ thuật thông dụng có clustering, PCA, hashing…
- Regression, classification là các bài toán căn bản trong machine learning, có thể ứng dụng trực tiếp vào dự đoán các sản phẩm phù hợp. Các kỹ thuật phổ biến: linear regression, ridge regression, SVM, random forests…

Content-based recommendation

- Dựa vào các thuộc tính của sản phẩm.
- Mạnh với các sản phẩm giàu nội dung như trong lĩnh vực truyền thông, quảng cáo, y tế…
- Có thể recommend sản phẩm mới, thích hợp khi danh sách sản phẩm được cập nhật liên tục.



Các bước chính

- Biểu diễn mỗi sản phẩm dưới dạng một vector thuộc tính.
- Recommend các sản phẩm tương tự nhau.
- Hoặc xây dựng profile người dùng theo các thuộc tính sản phẩm và recommend sản phẩm có thuộc tính phù hợp với profile người dùng.

ID	Name	Cuisine	Service	Cost
10001	Mike's Pizza	Italian	Counter	Low
10002	Chris's Cafe	French	Table	Medium
10003	Jacques Bistro	French	Table	High

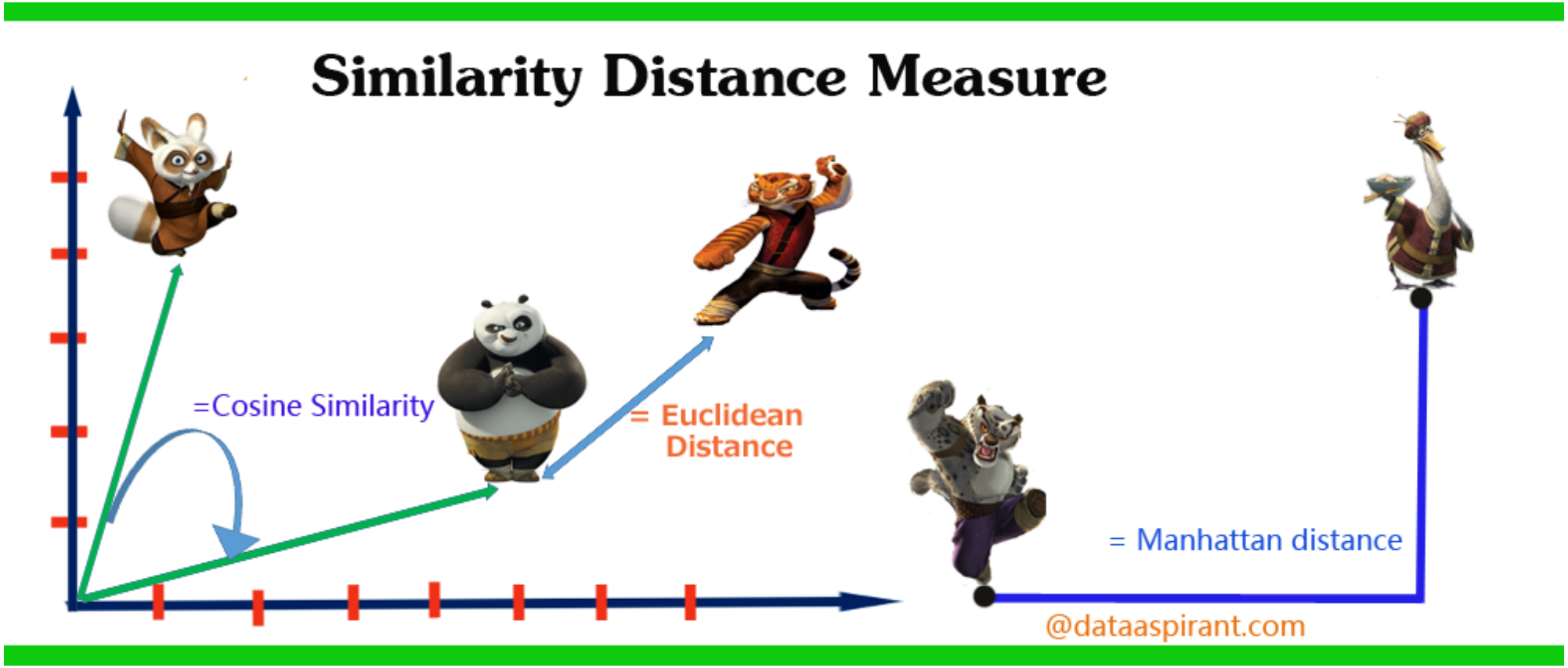
Thuộc tính và độ đo tương tự

- Các thuộc tính có thể là số thực, số nguyên hoặc rời rạc.
- Các độ đo tương tự khác nhau đòi hỏi dạng dữ liệu khác nhau.
- Chuyển đổi từ số sang rời rạc bằng cách chia khoảng.
- Chuyển ngược lại bằng one-hot encoding.

ID	Gender
1	Male
2	Female
3	Not Specified
4	Not Specified
5	Female

ID	Male	Female	Not Specified
1	1	0	0
2	0	1	0
3	0	0	1
4	0	0	1
5	0	1	0

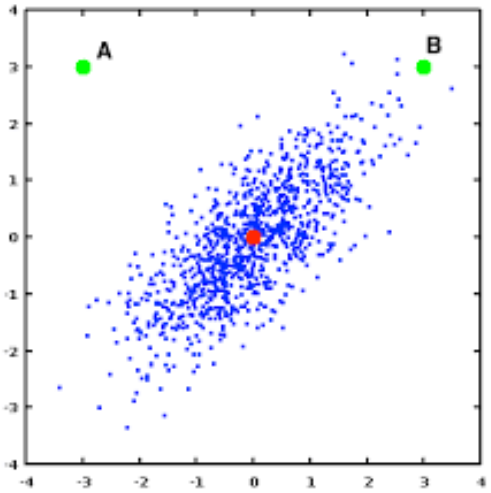
Các độ đo tương tự



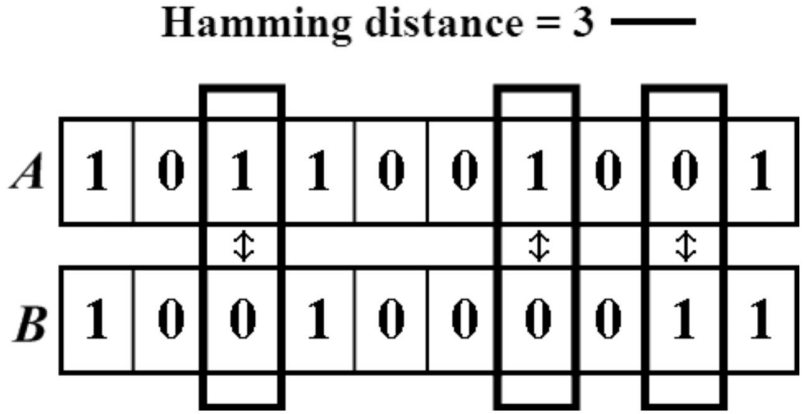
Các độ đo tương tự

Mahalanobis distance

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{s_i^2}},$$



Các độ đo tương tự

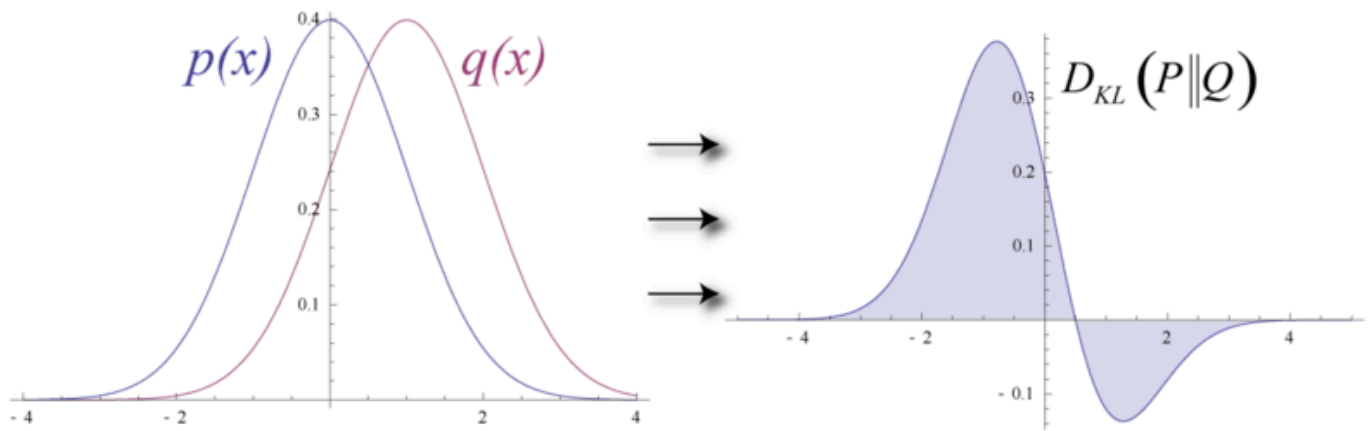


$$\text{Jaccard}(\mathcal{S}_1, \mathcal{S}_2) = \frac{|\mathcal{S}_1 \cap \mathcal{S}_2|}{|\mathcal{S}_1 \cup \mathcal{S}_2|}$$

Các độ đo tương tự

- KL divergence giữa hai phân bố xác suất

$$D(p||q) = \sum_{x} p(x) \log \frac{p(x)}{q(x)}$$

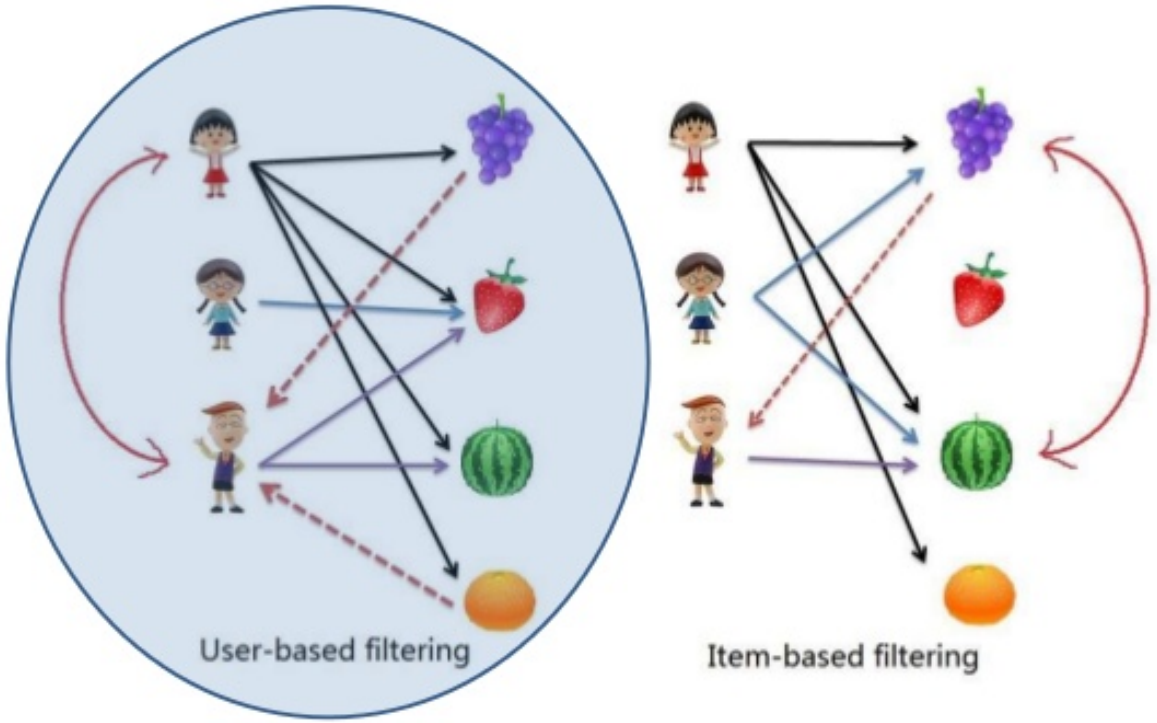


Collaborative filtering

- Dựa vào lịch sử giao dịch để tìm các quy luật tương tác giữa người dùng và sản phẩm.
- Không cần biết thuộc tính sản phẩm.
- Có khả năng khai thác thông tin ngoài phạm vi của các thuộc tính sản phẩm.
- Giả thiết căn bản: người dùng tương tự nhau quan tâm đến sản phẩm tương tự nhau.

Ưu điểm

User-based vs item-based



User-based collaborative filtering

- Biểu diễn mỗi người dùng bằng một vector các sản phẩm đã tương tác, có thể có trọng số.
- Tính độ tương tự giữa các vector đại diện cho người dùng.
- Đối với người dùng A, ước tính độ phù hợp của sản phẩm dựa vào lịch sử của nhóm người dùng tương tự như A.
- Có thể chọn k người dùng gần giống A nhất, hoặc chọn tất cả người dùng nhưng thêm trọng số để ưu tiên những người giống A hơn.

Công thức tính độ phù hợp

$$pr_{x,k} = m_x + \frac{\sum_{u_y \in N_x} \left(r_{y,k} - m_y\right) \mathrm{sim}\left(\boldsymbol{u}_x, \boldsymbol{u}_y\right)}{\sum_{u_y \in N_x} \left|\mathrm{sim}\left(\boldsymbol{u}_x, \boldsymbol{u}_y\right)\right|}$$

Công thức tính độ phù hợp

- $pr_{x,k}$ = độ phù hợp ước tính của sản phẩm k với người dùng x
- m_x = độ phù hợp trung bình của các sản phẩm với người dùng x
- $r_{y,k}$ = độ phù hợp của sản phẩm k với người dùng y
- m_y = độ phù hợp trung bình của các sản phẩm với người dùng y

Công thức tính độ phù hợp

Trong trường hợp giới hạn k người dùng gần giống nhất:

- $sim(u_x, u_y) = 1$
- N_x = tập hợp k người dùng gần giống x nhất

Trong trường hợp không giới hạn người dùng tương tự:

- $sim(u_x, u_y)$ = độ tương tự giữa các vector người dùng x và y, hoặc một hàm đồng biến dựa trên độ tương tự
- N_x = tập hợp người dùng có ít nhất một sản phẩm tương tác trùng với x

Khối lượng tính toán

Giả sử có m người dùng và n sản phẩm:

- Trong trường hợp xấu nhất, độ phức tạp của thuật toán là O(mn).
- Trên thực tế phần lớn người dùng chỉ tương tác với số sản phẩm giới hạn nên khối lượng tính toán cho nhóm này là O(m).
- Với số ít người dùng tương tác với nhiều sản phẩm, khối lượng tính toán cho nhóm này là O(n).
- Vậy độ phức tạp trên thực tế là O(m + n).

Kỹ thuật phụ trợ

- Với số lượng người dùng lớn, có thể phân nhóm để giảm khối lượng tính toán.
- Phân nhóm bằng dimensionality reduction: ánh xạ tập hợp người dùng hoặc tập hợp sản phẩm lên không gian ít chiều để thu nhỏ dữ liệu.
- Phân nhóm bằng clustering: chia tập hợp người dùng thành nhiều cụm nhỏ, khi recommend cho người dùng nào thì xem xét cụm đó.
- Thay vì công thức tính trung bình, có thể dùng các mô hình classification, regression để tính độ phù hợp ở bước cuối cùng.

Item-based collaborative filtering

- Biểu diễn mỗi sản phẩm bằng một vector người dùng.
- Tính độ tương tự giữa các sản phẩm.
- Đối với người dùng A, tìm các sản phẩm tương tự với các sản phẩm A đã tương tác.
- Recommend sản phẩm cho A từ các sản phẩm nói trên, bằng các tiêu chí như trọng số cao, nhiều người tương tác…

Tính độ tương tự giữa các sản phẩm

Với mỗi sản phẩm A:

Với mỗi người dùng X đã tương tác với A:

Với mỗi sản phẩm B khác A mà X đã tương tác:

Lưu dữ kiện là một người dùng đã tương tác với cả A và B.

Với mỗi sản phẩm B khác A đã có cùng người dùng tương tác :

Tính độ tương tự giữa A và B dựa trên các dữ kiện đã lưu.

Khối lượng tính toán

- Khối lượng tính toán lớn nhất nằm ở phần tính độ tương tự giữa các sản phẩm.
- Giả sử có m người dùng và n sản phẩm, thì độ phức tạp trong trường hợp xấu nhất là $O(n^2m)$.
- Trên thực tế độ phức tạp thường là O(mn) vì dữ liệu thưa, đại đa số các cặp sản phẩm không có cùng người dùng tương tác.
- Tuy nhiên bước này chỉ cần tính một lần cho tất cả các lượt người dùng.

Kỹ thuật phụ trợ

- Độ tương tự giữa các sản phẩm có thể tính trước và thỉnh thoảng mới cập nhật.
- Có thể bổ sung thông tin về thuộc tính sản phẩm khi tính độ tương tự để giải quyết trường hợp sản phẩm ít tương tác.
- Có thể áp dụng các kỹ thuật phân nhóm, classification, regression… như với user-based collaborative filtering.

Matrix factorization

- Áp dụng được cho cả user-based và item-based collaborative filtering.
- Thuộc loại latent factor model, mô tả người dùng và sản phẩm bằng các nhân tố ẩn.
- Có tác dụng giảm số chiều dữ liệu (dimensionality reduction).

Các bước chính

- Lập user-item matrix.
- Tách user-item matrix thành hai thừa số: user matrix và item matrix.
- Mỗi người dùng tương ứng với một hàng trên user matrix.
- Mỗi sản phẩm tương ứng với một cột trên item matrix.
- Ước tính độ phù hợp bằng tích vô hướng của vector người dùng và vector sản phẩm.
- Ước tính độ tương tự bằng cách so sánh các vector người dùng với nhau (hoặc các vector sản phẩm với nhau) nếu các ma trận thỏa mãn một số điều kiện nhất định.

User-item matrix

- Mỗi người dùng trên một dòng, mỗi sản phẩm trên một cột.
- Có thể chuẩn hóa bằng cách trừ đi giá trị trung bình.
- Thường là ma trận thưa, vì đa số các cặp user-item không có tương tác.

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	**0**	**3**	**0**	**3**	**0**
User 2	**4**	**0**	**0**	**2**	**0**
User 3	**0**	**0**	**3**	**0**	**0**
User 4	**3**	**0**	**4**	**0**	**3**
User 5	**4**	**3**	**0**	**4**	**0**

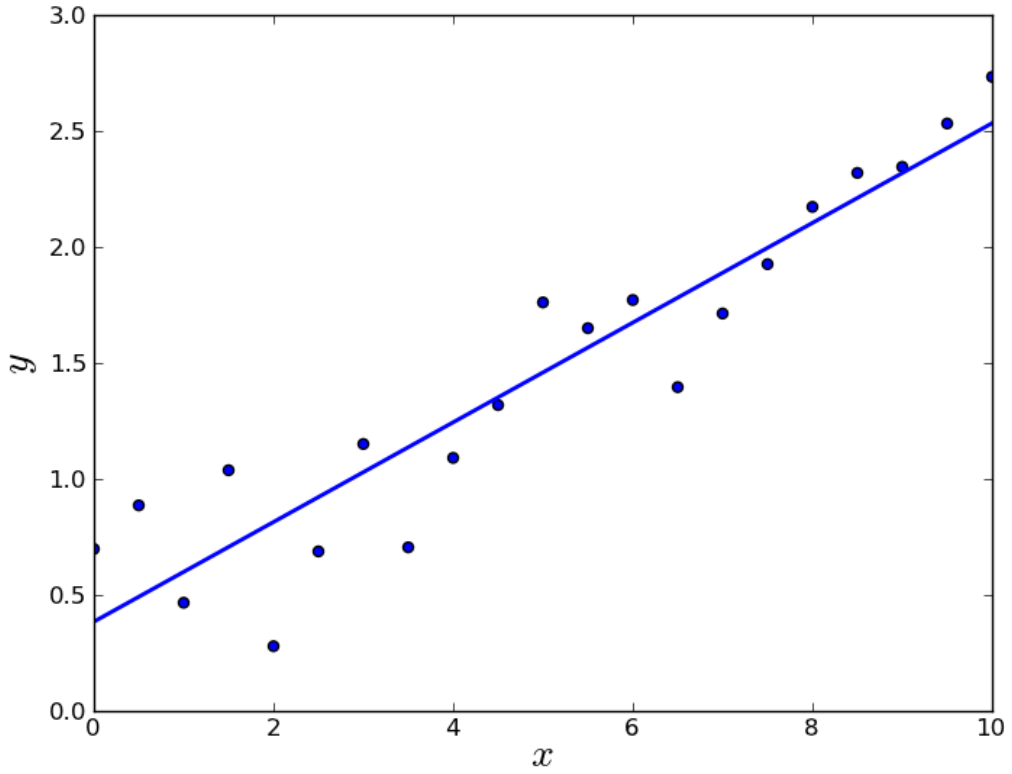
Giả thiết

- Tương tác giữa người dùng và sản phẩm chịu ảnh hưởng của hai loại tác nhân: các quy luật tương tác và các yếu tố ngẫu nhiên (do cá nhân người dùng, tác động ngoại cảnh…)
- Các yếu tố ngẫu nhiên không thể dự đoán, nhưng tổng hiệu ứng trên toàn tập hợp là nhỏ.
- Các quy luật tương tác là tác nhân chính và có thể mô hình hóa.
- Mục tiêu là ước tính độ phù hợp theo các quy luật tương tác, bỏ qua hiệu ứng của các yếu tố ngẫu nhiên.

Giả thiết

- User-item matrix R = S + N
- S là hiệu ứng của các quy luật
- N là hiệu ứng của các yếu tố ngẫu nhiên
- N có biên độ nhỏ.
- S có intrinsic dimensionality nhỏ.

Intrinsic dimensionality

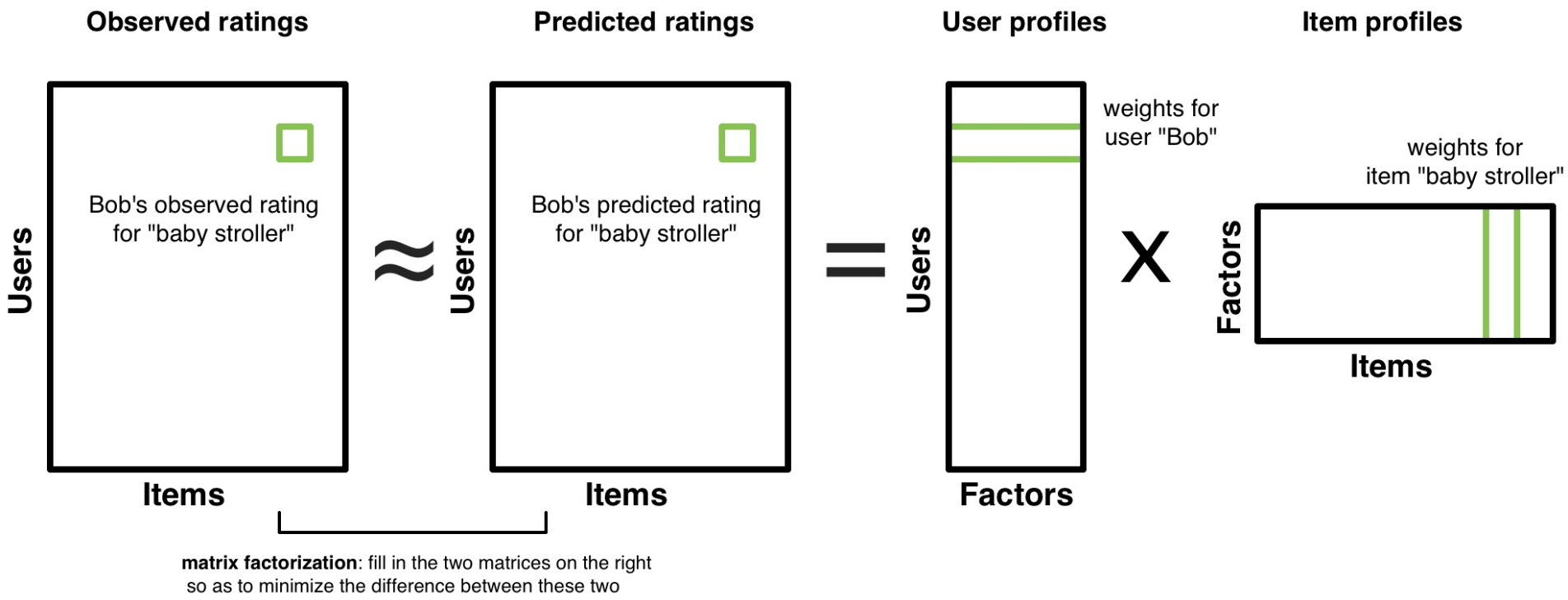


Factorization

Giả sử có m người dùng và n sản phẩm:

- User-item matrix R ~ U I kích thước m x n.
- U là user matrix, kích thước m x d.
- I là item matrix, kích thước d x n.
- d là tham số, nhỏ hơn nhiều so với m, n.
- S = U I là ma trận xấp xỉ với R, có intrinsic dimension d.

Factorization



Ước tính độ phù hợp

Để ước tính độ phù hợp của sản phẩm y với người dùng x:

- Vector đại diện x là dòng U^x trên ma trận U.
- Vector đại diện y là cột V_y trên ma trận V.
- Ước tính độ phù hợp bằng tích vô hướng:

  $U^x . V_y = S_{xy}$ (phần tử ở dòng x cột y của ma trận S)

- $S_{xy} \sim R_{xy}$ là phần còn lại khi đã loại bỏ các hiệu ứng ngẫu nhiên, tức là độ phù hợp của y với x theo các quy luật tương tác.

Nhận xét

- Bằng cách giảm chiều dữ liệu, ta loại bớt thông tin để tránh việc kết quả chỉ lặp lại user-item matrix (S quá gần với R).
- Giả thiết quan trọng là phương pháp factorization loại bỏ đúng các hiệu ứng ngẫu nhiên.
- Mỗi vector người dùng hoặc sản phẩm có d chiều, ứng với d nhân tố ẩn (latent factor).

SVD

- Thuật toán cho kết quả chính xác.
- Các phần tử trên ma trận S xếp theo thứ tự giảm dần.

$$\underset{m \times n}{X} = \underset{m \times r}{U} \; \underset{r \times r}{S} \; \underset{r \times n}{V^{\mathsf{T}}}$$

SVD

USER

ITEM

$$\underset{m \times n}{X} = \underset{m \times r}{U} \; \underset{r \times r}{S} \; \underset{r \times n}{V^\top}$$

$$\underset{m \times n}{X} = \underset{m \times r}{U} \; \underset{r \times r}{S} \; \underset{r \times n}{V^\top}$$

USER

ITEM

Giảm chiều bằng SVD

- Loại bỏ thông tin bằng cách bỏ bớt các chiều cuối cùng.
- Phần còn lại đảm bảo là gần với X nhất trong tất cả các cách factorization.

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}_{m \times n} = \begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}_{m \times r} \begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}_{r \times r} \begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}_{r \times n}$$

U S V^T

$m \times d$ $d \times d$ $d \times n$

Factorization bằng tối ưu hóa

- Dùng tối ưu hóa để ước tính user matrix và item matrix trực tiếp.
- Đại lượng cần tối ưu là sai số giữa độ phù hợp ước tính và dữ liệu thật trên user-item matrix.
- Dùng regularization để loại bớt thông tin.
- Tối ưu bằng SGD hoặc ALS.

SVD hàng nhái

$$\min_{b_*,q_*,p_*} \sum_{(u,i)\in K} (r_u i - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda(b_i^2 + b_u^2 + ||q_i||^2 + ||p_u||^2).$$

THẬT

GIẢ TẠO TÍNH

REGULARIZER

SVD++

- Mỗi sản phẩm có thêm một vector phụ.
- Ước tính độ phù hợp dùng cả giá trị vector phụ của tất cả các sản phẩm người dùng đó đã tương tác.
- Phải thêm cả các vector phụ vào regularization.

SVD++

$$\min_{b_*,q_*,p_*} \sum_{(u,i)\in K} (r_u i - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda(b_i^2 + b_u^2 + ||q_i||^2 + ||p_u||^2).$$

$$q_i^T \left(p_u + \frac{1}{\sqrt{|N_u|}} \sum_{k \in N_u} y_k\right) + \sum_{k \in N_u} ||y_k||^2$$

Độ tương tự giữa người dùng hoặc sản phẩm

- Có thể ước tính độ tương tự giữa các người dùng bằng cách so sánh các vector người dùng (thường dùng cosine similarity).
- Tuy nhiên cần điều kiện: item matrix phải là orthogonal matrix.
- Ngược lại muốn so sánh vector sản phẩm thì user matrix phải là orthogonal matrix.
- Một số phương pháp factorization không đảm bảo điều kiện trên, nhưng có thể dùng biến đổi đại số để khắc phục.