Collaborative filtering

Các hệ thống gợi ý được sử dụng rộng rãi trên các nền tảng như Amazon, Netflix, các ứng dụng giao đồ ăn, v.v. để đề xuất sản phẩm, phim, hoặc nhà hàng cho người dùng

Lấy ví dụ với bài toán gợi ý phim

Bạn có một hệ thống gồm:

- **Người dùng** (Alice, Bob, Carol, Dave – được đánh số từ 1 đến 4).
- **Phim** (Love at Last, Romance Forever, Cute Puppies of Love, Nonstop Car Chases, Sword vs Karate – tổng cộng 5 phim).
- Người dùng đánh giá các phim bằng thang điểm từ **0 đến 5 sao**, và có thể **chưa xem** một số phim (ký hiệu bằng dấu hỏi `?`).

Mục tiêu của hệ thống gợi ý:

- **Dự đoán** điểm số mà người dùng có thể đánh giá cho các phim họ **chưa xem**.
- Từ đó **gợi ý** các phim mà người dùng **có khả năng sẽ đánh giá cao** (ví dụ: 5 sao).

Hệ thống gợi ý khi có đặc trưng của item

1. Giả định thêm: Có đặc trưng của từng item

2. Mô hình dự đoán đánh giá (rating prediction model)

- Mỗi người dùng **j** có một bộ tham số riêng:
  - Vector trọng số `w(j)` (có kích thước bằng số đặc trưng n).
  - Hệ số chệch `b(j)`.
- Dự đoán đánh giá của người dùng **j** cho item **i**:

$$\hat{y}(i,j) = w(j)^T \cdot X(i) + b(j)$$

- Đây là **mô hình hồi quy tuyến tính** riêng biệt cho từng người dùng.

3. Hàm mất mát (Cost Function)

- Để học `w(j)` và `b(j)`, dùng hàm mất mát:

$$2J(w(j), b(j)) = \frac{1}{2m(j)} \sum_{i:r(i,j)=1} \left(w(j)^T \cdot X(i) + b(j) - y(i,j)\right)^2 + \frac{\lambda}{2m(j)} \sum_{k=1}^{n} w(j)_k^2$$

  - Chỉ tính trên những phim mà người dùng **j** đã đánh giá (`r(i,j) = 1`).

- `m(j)` là số phim người dùng **j** đã đánh giá.
- Term thứ hai là **regularization** để tránh overfitting.

- Thực tế, **bỏ** `1/m(j)` cũng được vì không ảnh hưởng đến kết quả tối ưu.

4. Tổng thể cho toàn bộ người dùng

- Tổng hợp hàm mất mát của tất cả người dùng:

$$J_{\text{total}} = \sum_{j=1}^{n_u} J(w(j), b(j))$$

- Dùng **gradient descent** hoặc thuật toán tối ưu khác để học tất cả các tham số.

Học đặc trưng (feature) của item khi không có sẵn

Khi không có đặc trưng X của phim:

- Thay vì có đặc trưng, giờ **không biết giá trị của X** nữa → thay bằng dấu `?`.
- Giả sử đã biết các **tham số người dùng** (`w^j`, `b^j`) → có thể **ngược lại tìm ra X** sao cho mô hình dự đoán gần đúng với đánh giá thật.

Ý tưởng chính: Học đặc trưng X từ dữ liệu đánh giá

- Dự đoán của người dùng j cho item i:

$$\hat{y}_{i,j} = w^j \cdot x^i + b^j$$

- Nếu đã biết w^j, b^j của các người dùng **đã đánh giá** item i, thì có thể tìm ra x^i sao cho mô hình dự đoán đúng các đánh giá đó.
- Với mỗi item x^i, tối ưu hóa để dự đoán tốt các đánh giá từ người dùng đã đánh giá nó → **học đặc trưng từ chính hành vi đánh giá của người dùng**.

Hàm mất mát để học đặc trưng item

Đối với **một item i**, muốn tìm `x^i` sao cho dự đoán tốt các đánh giá:

$$J(x^i) = \frac{1}{2} \sum_{j:r(i,j)=1} \left(w^j \cdot x^i + b^j - y(i,j)\right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} \left(x_k^i\right)^2$$

- Chỉ tính trên các user j đã đánh giá item i.

Đối với **toàn bộ các item**:

$$J(w, b, x) = \sum_{(i,j):r(i,j)=1} \left(w^j \cdot x^i + b^j - y(i,j)\right)^2 + \lambda \left(\sum_j \|w^j\|^2 + \sum_i \|x^i\|^2 \right)$$

- Tối ưu bằng Gradient Descent

Collaborative Filtering

- Gọi là **lọc cộng tác** vì:
  - Người dùng **cùng đánh giá** một item giúp "cộng tác" để xác định đặc trưng của item đó.
  - Sau đó, đặc trưng đó lại được dùng để dự đoán cho những người **chưa từng sử dụng item đó**

Binary labels

Bài toán mới

- Trước đây: dự đoán đánh giá số (sao từ 0–5).
- Bây giờ: **nhãn nhị phân** `y(i,j) ∈ {1, 0, ?}`:
  - `1`: người dùng thích (hoặc tương tác).
  - `0`: người dùng không thích (hoặc không tương tác sau khi thấy).
  - `?`: người dùng chưa từng thấy item đó.

Từ hồi quy tuyến tính → hồi quy logistic

- Trước đây (rating dạng số):

$$\hat{y}_{i,j} = w^j \cdot x^i + b^j$$

- Bây giờ (dự đoán xác suất nhãn nhị phân):

$$\hat{y}_{i,j} = g(w^j \cdot x^i + b^j)$$

  với:

$$g(z) = \frac{1}{1 + e^{-z}} \quad (\text{hàm sigmoid})$$

  → đây là **mô hình giống như logistic regression**.

Hàm mất mát mới: Binary Cross-Entropy

- Với mỗi cặp user–item có đánh giá:

$$\mathcal{L}(y_{i,j}, \hat{y}_{i,j}) = -y_{i,j} \log \hat{y}_{i,j} - (1 - y_{i,j}) \log(1 - \hat{y}_{i,j})$$

- Tổng chi phí toàn bộ hệ thống:

$$J(w,b,x) = \sum_{(i,j):r(i,j)=1} \mathcal{L}(y_{i,j}, g(w^j \cdot x^i + b^j)) + \lambda \left(\sum_j \|w^j\|^2 + \sum_i \|x^i\|^2 \right)$$