PROJECT 01:  
RECOMMENDER SYSTEMS WITH SPARK

**Thông tin thành viên**

1712284 – Hoàng Gia Bảo

1712295 – Nguyễn Thanh Bình

1712309 – Nguyễn Hoàng Tuấn Cường

**SPARSE FC  
(SPARSE FULLY-CONNECTED)**

**1/ Ý tưởng chính của phương pháp**

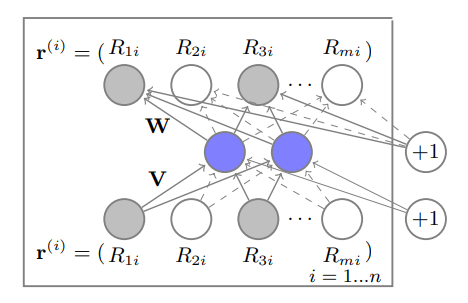
* Cải tiến dựa trên phương pháp I-AutoRec

**Tổng quan về I-AutoRec**

Ta có m người dùng, n mặt hàng, và một ma trận R chứa các đánh giá m người dùng đối với n mặt hàng đó.

Mỗi mặt hàng i có một vec-tơ đánh giá(r(i)) của m người dùng, r(i) = (R1i, …, Rmi)

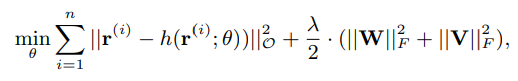
Mục tiêu là thiết kế một mô hình tự động mã hóa dựa trên từng mặt hàng i, nhận input là từng vector r(i), chiếu nó sang một miền không gian tiềm ẩn có số chiều thấp, sau đó tái tạo lại vec-tơ r(i) đó trong miền không gian đầu ra, ta sẽ dự đoán được các đánh giá chưa có của khách hàng đối với mặt hàng i đó



Hình 1. Mô hình Item-based AutoRec. Hình tròn màu xám thể hiện người dùng có đánh giá mặt hàng i này, hình tròn trắng thể hiện người dùng đó không đánh giá mặt hàng i này, hình tròn xanh là lớp tiềm ẩn có số chiều thấp. Các mũi tên đang thể hiện quá trình học (lan truyền ngược) để cập nhật trọng số. Chỉ có các hình tròn xám mới được quan tâm để lan truyền ngược(mũi tên liền), các hình tròn trắng thì không quan tâm(mũi tên nét đứt)

Hàm để tái tạo lại các vec-tơ đầu vào r:



Ta muốn cực tiểu hóa độ lỗi của quá trình học

Kí hiệu F ở đây nghĩa là tính norm-2 toàn bộ ma trận

Chỉ tính norm-2 các dự đoán của những đánh giá có tồn tại ban đầu (hay ta dễ hiểu hơn, là học làm sao cho các dự đoán của các đánh giá ban đầu, sau khi được tái tạo sẽ gần đúng với đánh giá đầu vào nhất)

**Sparse FC**

Có kiến trúc mạng tương tự kiến trúc của I-AutoRec, nhưng ma trận trọng số W, V được tham số hóa lại bằng cách:

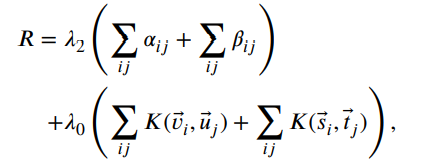
* Ta sẽ nhận ma trận kiểu Hadamard ma trận W, V với ma trận trọng số kernel(w\_hat) để có được ma trận thể hiện các liên kết thưa thớt. Từng giá trị của w\_hat sẽ được tính bằng công thức:



D là khoảng cách 2 vec-tơ

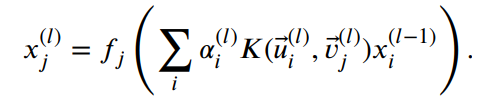
Cách tính độ lỗi sau mỗi mẫu huấn luyện cũng có thay đổi nhỏ so với phương pháp I-AutoRec, đó là việc cộng thêm sai số của ma trận trọng số kernel (w\_hat) để học trong quá trình lan truyền ngược nữa:

Đây chính là tổng norm-2 của ma trận W và V(giống như trong hàm cực tiểu độ lỗi phương pháp I-AutoRec bên trên).  
λ2 > 0 là tham số kiểm soát overfitting của mô hình



Đây là norm-2 của ma trận w\_hat.  
λ0 là tham số thể hiện mật độ của dữ liệu trong ma trận user-item ban đầu

Công thức định nghĩa mạng nơ-ron kernel có d chiều

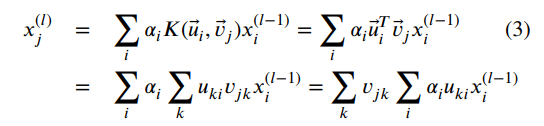


Định nghĩa hàm kernel:

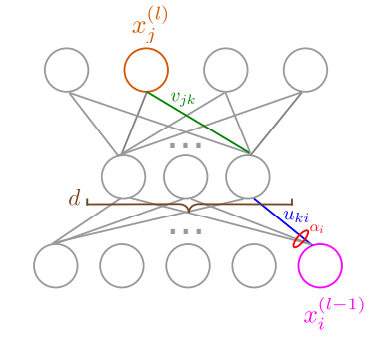


Nếu là mạng fully-connected, ta cho hàm Φ thành hàm tuyến tính (Φ(x) = x), tham số alphai = 1,

hàm K tích bằng tích vô hướng, ta sẽ có được như sau:



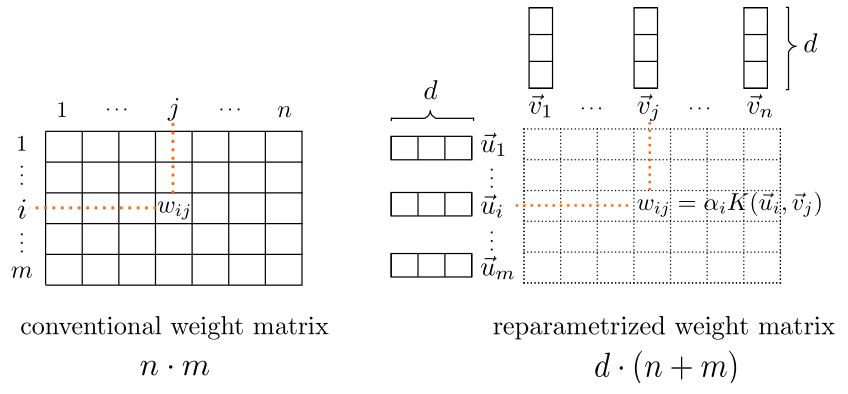
Lúc này (3) sẽ được biểu diễn như mô hình mạng dưới đây:



Giả sử tầng (l – 1) có m phần tử, tầng (l) có n phần tử

Lúc này, thay vì chỉ có 1 ma trận trọng số W(m x n) giữa 2 tầng liên tiếp trong mạng, ta đã tách ra được 2 ma trận khác nhau: ma trận U(m x d) và ma trận V(d x n). Có thể giải thích điều này như là sự phân rã ma trận W lớn thành 2 ma trận con nhỏ hơn. Thao tác này mạng lại lợi ích:

* Bản chất ma trận item-user ban đầu có độ thưa thớt cực cao, và đánh giá của phim cũng dựa vào số lượng tương đối nhỏ các tính năng (VD: thể loại, diễn viên…). Nên ta giảm chiều xuống để tăng tính tổng quát hóa, tìm ra mỗi tương quan giữa các đánh giá, ở không gian nhỏ hơn
* Ta thấy số lượng tham số có được = (m x d + d x n) < m x n 🡪 d < (m\*n)/(m + n) (\*). Nếu ta chọn d thỏa điều kiện (\*) ta thấy số lượng tham số của mô hình giảm xuống đáng kể, giúp tăng tốc khả năng học



Lựa chọn kernel sao cho các nơ-ron ở càng xa thì liên kết sẽ càng yếu, các nơ-ron ở càng gần thì sẽ liên kết càng mạnh (đây là lí do ta chọn hàm kernel Kfs bên trên)

**Ảnh có chứa bàn

Mô tả được tạo tự động2/ Kết quả thực nghiệm**

Bảng 1. So sánh các phương pháp khác nhau trên các tập dữ liệu MovieLens.  
Giá trị trong bảng là giá trị RMSE trung bình của các đánh giá cần điền khuyết  
(giá trị càng thấp càng tốt) từ phương pháp 5-Folds Cross-Validation.  
Đối với tập dữ liệu ML-10M và ML-1M ta chia tập train/validation theo tỉ lệ 90/10,  
với tập dữ liệu ML-100K ta chia tập train/validation theo tỉ lệ 80/20.  
Giá trị nằm trong dấu ngoặc đơn sau tên mỗi phương pháp là số tầng ẩn của mạng nơ-ron

Ảnh có chứa bàn

Mô tả được tạo tự động

Bảng 2. Phương pháp SparseFC đã giảm được phép toán nhân tích lũy (MACs)  
cho các đánh giá cần điền khuyết

**3/ Ưu, nhược điểm của phương pháp Sparse FC**

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| * Giúp tăng tốc khả năng dự đoán của mô hình, do giảm được MACs * Khả năng dự đoán tốt hơn so với một số các phương pháp tiên tiến khác | * Chậm hơn so với I-AutoRec(mô hình cơ bản) một xíu trong quá trình học, do có nhiều tham số hơm |