

```


$$q_i = q_i + \beta \cdot (e_{ui} \cdot p_u - \lambda_3 \cdot q_i)$$


$$b_u = b_u + \beta \cdot (e_{ui} - \lambda_1 \cdot b_u)$$


$$b_i = b_i + \beta \cdot (e_{ui} - \lambda_2 \cdot b_i)$$

} // end all i ∈ R(u)
for(all i ∈ R(u)) { // tổng bên trong cho x

$$x_i = x_i + \beta \cdot \left( |R(u)|^{-\frac{1}{2}} \cdot (r_{ui} - b_{ui}) \cdot \text{sum} - \lambda_4 \cdot x_i \right)$$

}
} // end all u
} // end steps Iterations
return (qi, xi, bi | i=1,...,n; bu | u=1,...,m)
} // end Algorithm

```

Các tham số của mô hình lúc này là q_i, x_i, b_u, b_i , Koren đề xuất phân biệt cách tính các thành phần bias khác nhau [12], khi xét quan hệ giữa các mục tin thì vẫn chọn các bias là hằng số ban đầu, còn các b_u, b_i bên ngoài quan hệ sẽ được tính toán và cập nhật trong quá trình huấn luyện. Với tiếp cận này thì độ phức tạp về thời gian tính toán dự đoán là $O(f - \sum_u |R(u)|)$ và độ phức tạp về không gian tuyến tính với kích cỡ đầu vào $O(m+nf)$.

4 KẾT QUẢ THỰC NGHIỆM

4.1 Tập dữ liệu huấn luyện

Nghiên cứu này, chúng tôi sử dụng tập dữ liệu MovieLens 100K¹, mỗi người dùng đánh giá ít nhất 20 bộ phim với các tập dữ liệu dùng cho kiểm tra chéo 5-fold sẵn có được chia ra tập train (uX.base) và tập test (uX.test) với X từ 1 đến 5. Chúng tôi tiền xử lý các tập dữ liệu này bao gồm xóa bỏ trường timestamp, trộn ngẫu nhiên các mẫu, chèn vào số mẫu ở đầu mỗi tập.

4.2 Độ đo

Khi huấn luyện, chúng tôi thử tìm kiếm trên

Bảng 1: Iters=50, NF=64

LR	Rb _u	Rb _i	Rq _i	Rx _i	RMSE	Time(m)
0.01	0.005	0.005	0.05	0.5	0.9289	0.2290
0.005	0.05	0.05	0.05	0.5	0.9224	0.2233
0.001	0.05	0.005	0.005	0.005	0.9318	0.2249

Bảng 2: Iters=100, NF=64

LR	Rbu	Rbi	Rqi	Rxi	RMSE	Time(m)
0.01	0.05	0.005	0.5	0.05	0.9346	0.4467
0.005	0.05	0.005	0.05	0.5	0.9288	0.4456
0.001	0.05	0.005	0.05	0.05	0.9245	0.4449

Bảng 3: Iters=200, NF=64

LR	Rb _u	Rb _i	Rq _i	Rx _i	RMSE	Time(m)
0.01	0.005	0.005	0.5	0.05	0.9347	0.896
0.005	0.005	0.005	0.5	0.05	0.9328	0.892
0.001	0.05	0.05	0.05	0.5	0.9225	0.891

hiều trường hợp của các siêu tham số (meta-parameters) khác nhau, sử dụng kỹ thuật tìm kiếm lưới (grid search = raw search + smooth search) để đạt được các siêu tham số cho lỗi RMSE trên tập kiểm tra tốt nhất. Lỗi RMSE được xác định bằng công thức:

$$RMSE = \sqrt{\frac{1}{|D_{test}|} \sum_{u,i,r \in D_{test}} (r_{ui} - \hat{r}_{ui})^2} \quad (5)$$

4.3 Kỹ thuật huấn luyện và kết quả

Chúng tôi sử dụng bước lặp giới hạn (Max_Num_Iters) thay vì lặp cho đến khi hội tụ để quá trình huấn luyện nhanh hơn và giải quyết vấn đề phân tử mới với dự đoán trung bình toàn cục.

Sau đây, chúng tôi thống kê một số kết quả theo từng tốc độ học (LR), số nhân tố (NF) và các regularization (Rb_u, Rb_i, Rq_i, Rx_i). Kết quả thực nghiệm tìm kiếm thô được thể hiện trên 3 tốc độ học: 0.001, 0.005 và 0.01 với số nhân tố 16, 32, 64, cùng với số vòng lặp giới hạn như các bảng sau (trích từ dữ liệu với lỗi RMSE thấp):