

# Large Scale Recommender Systems in Spark

Kirk Hunter, Mrunmayee H. Bhagwat, Swetha Reddy

# Data & Tools

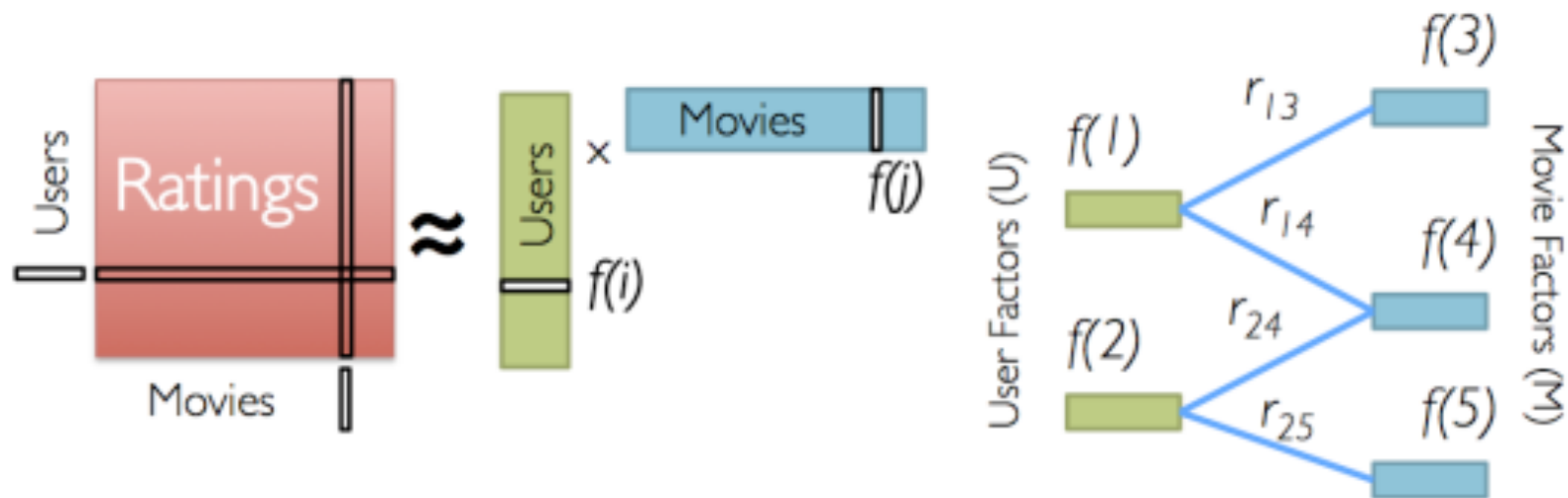
- Yahoo! Music Ratings (user id, song id, rating)
- Train: 700 million ratings, 1.8 million users, 136K songs
- Test: 18 million ratings, 1.8 million users, 136K songs
- Stored data in S3
- 5 node cluster running Spark on AWS EMR

# Methods

- Alternating Least Squares (ALS)
- Locality Sensitive Hashing (LSH)

# Alternating Least Squares (ALS)

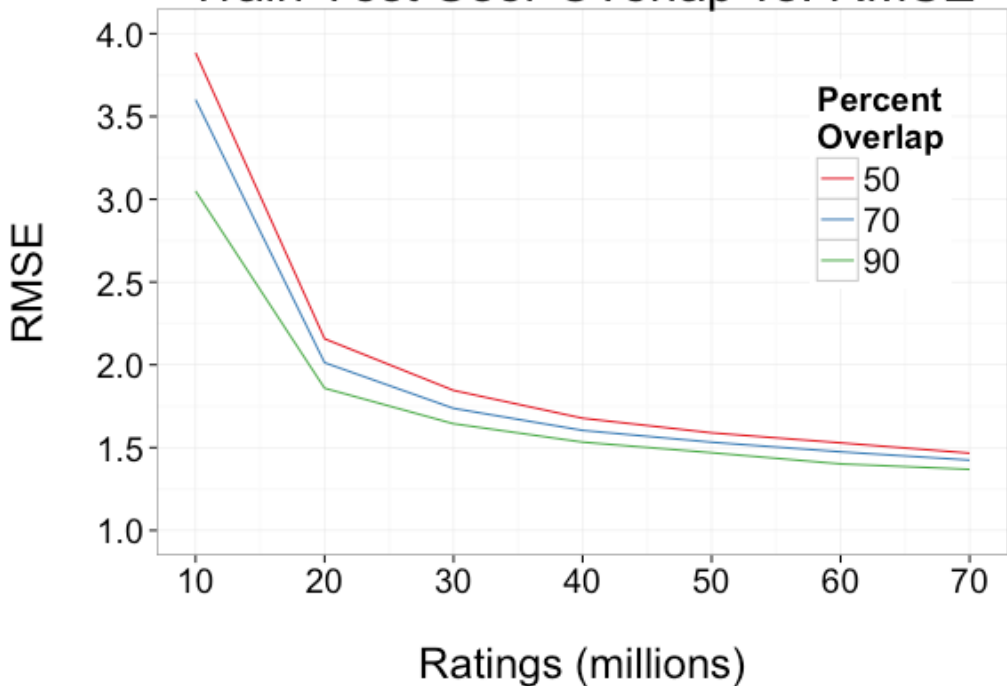
Low-Rank Matrix Factorization:



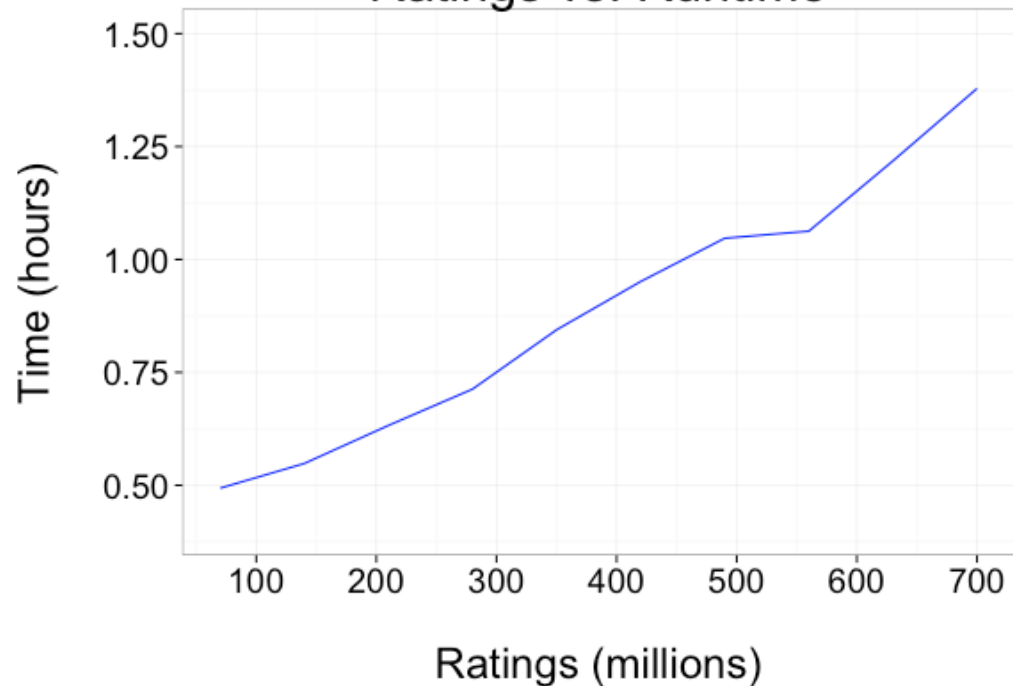
Iterate:

$$f[i] = \arg \min_{w \in \mathbb{R}^d} \sum_{j \in \text{Nbrs}(i)} (r_{ij} - w^T f[j])^2 + \lambda ||w||_2^2$$

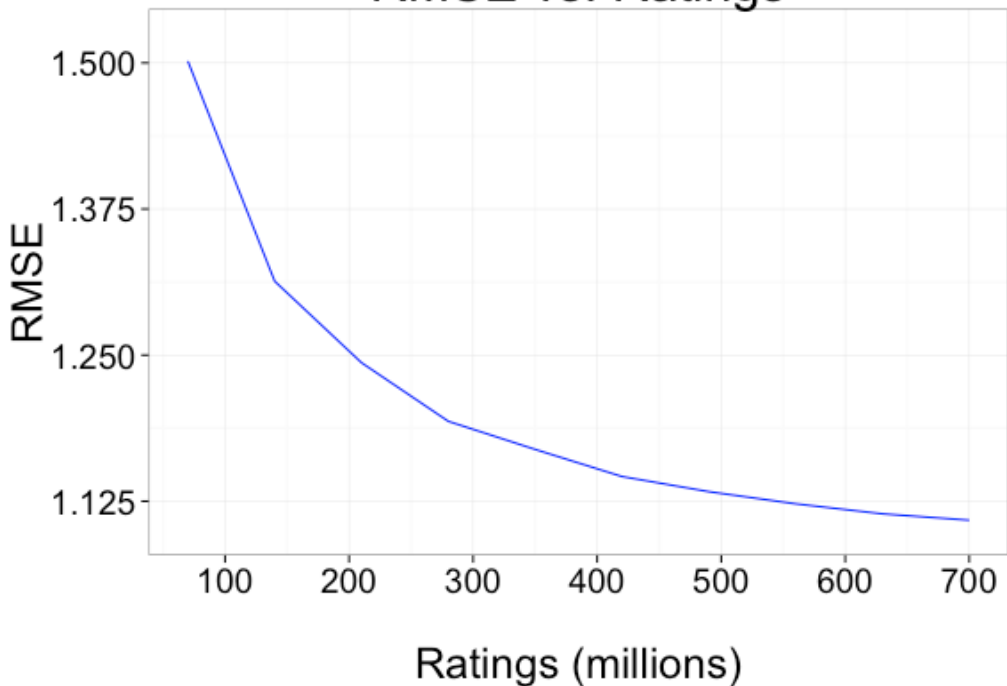
### Train Test User Overlap vs. RMSE



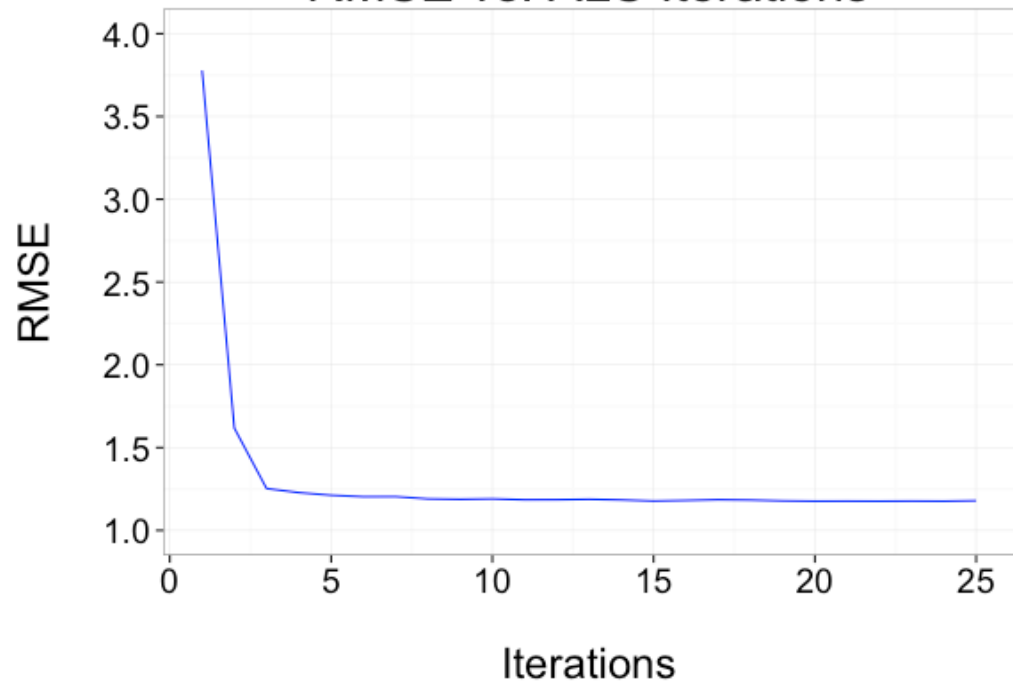
### Ratings vs. Runtime



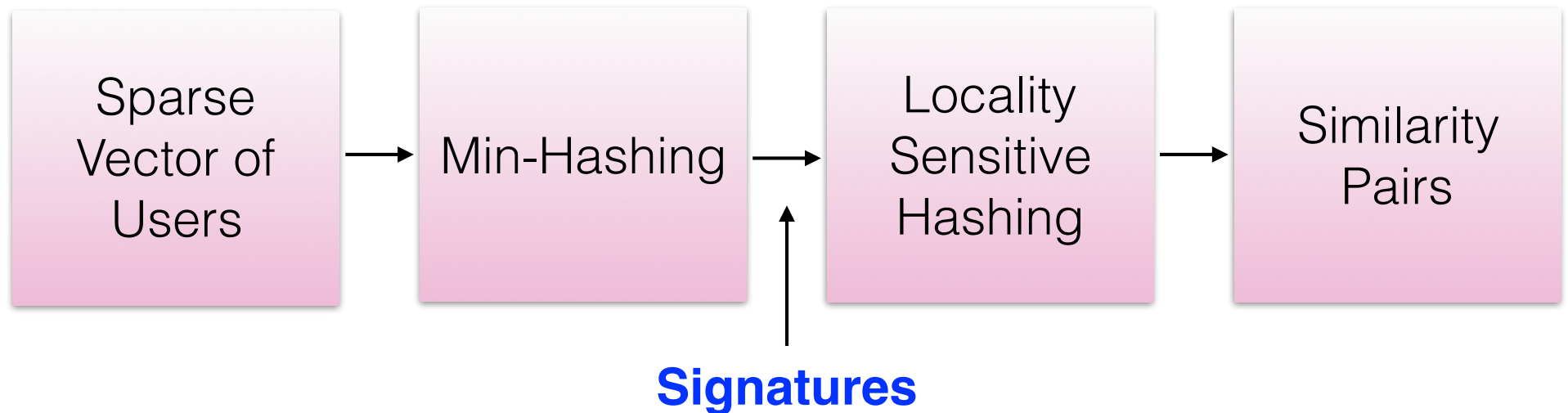
### RMSE vs. Ratings



### RMSE vs. ALS Iterations

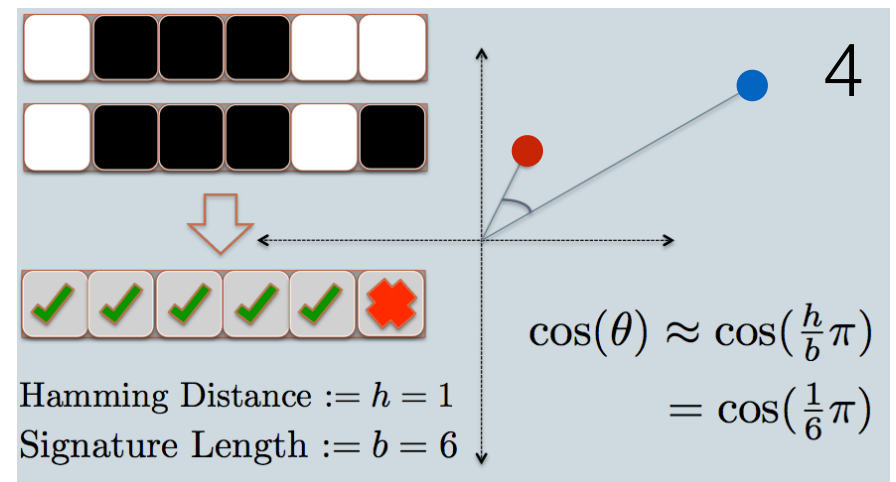
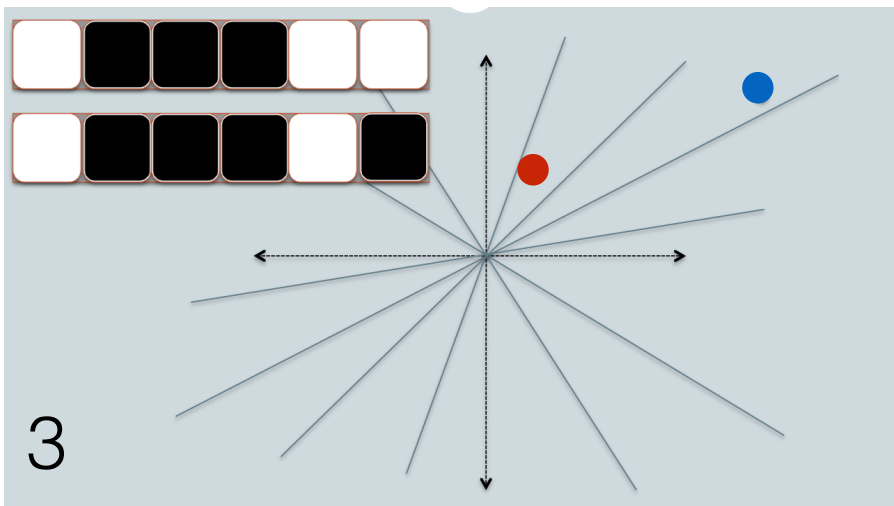
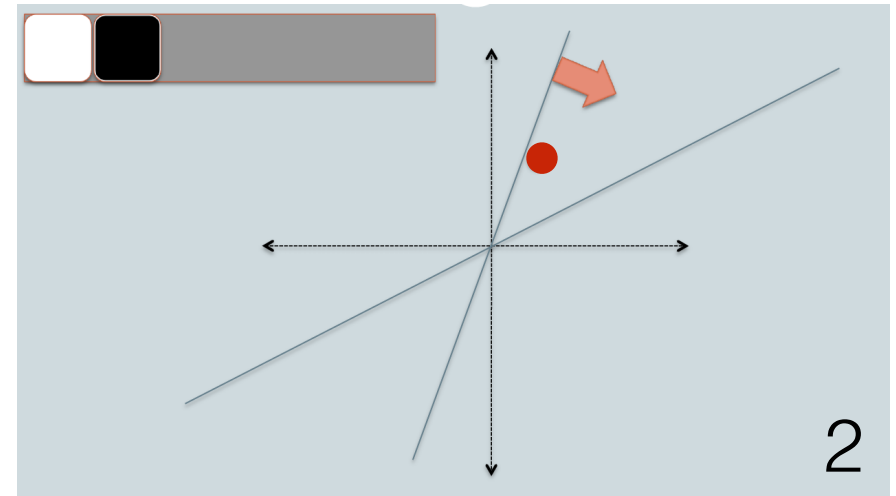
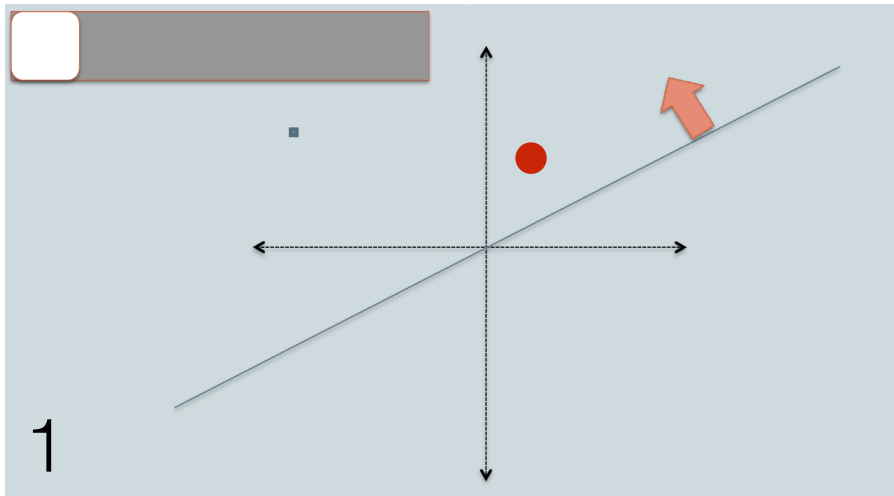


# Locality Sensitive Hashing (LSH)

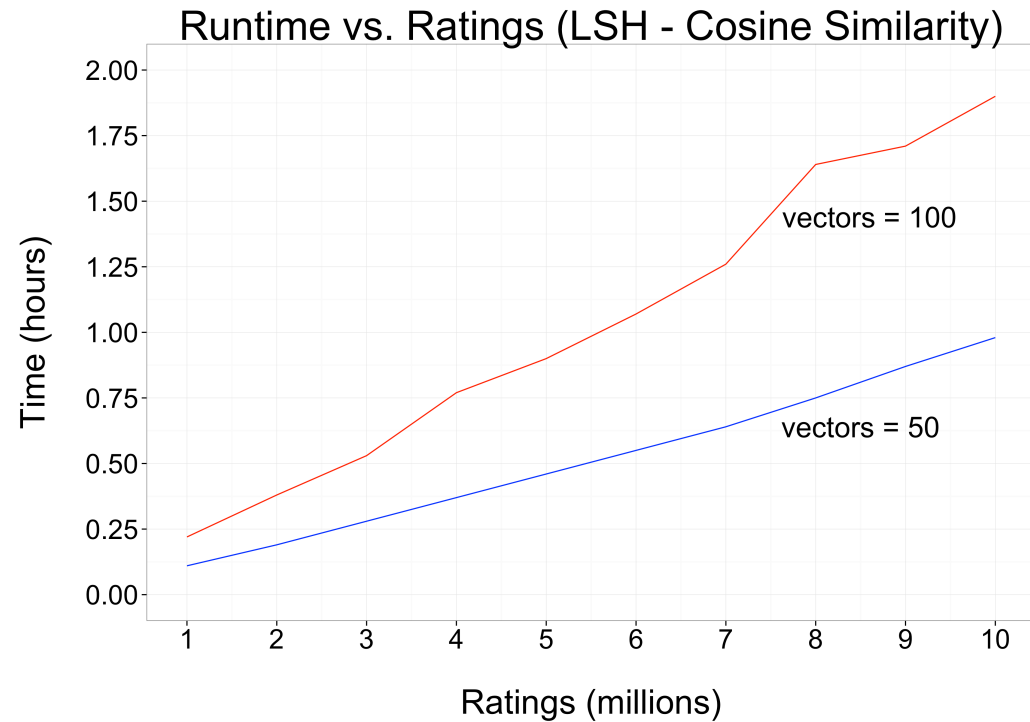
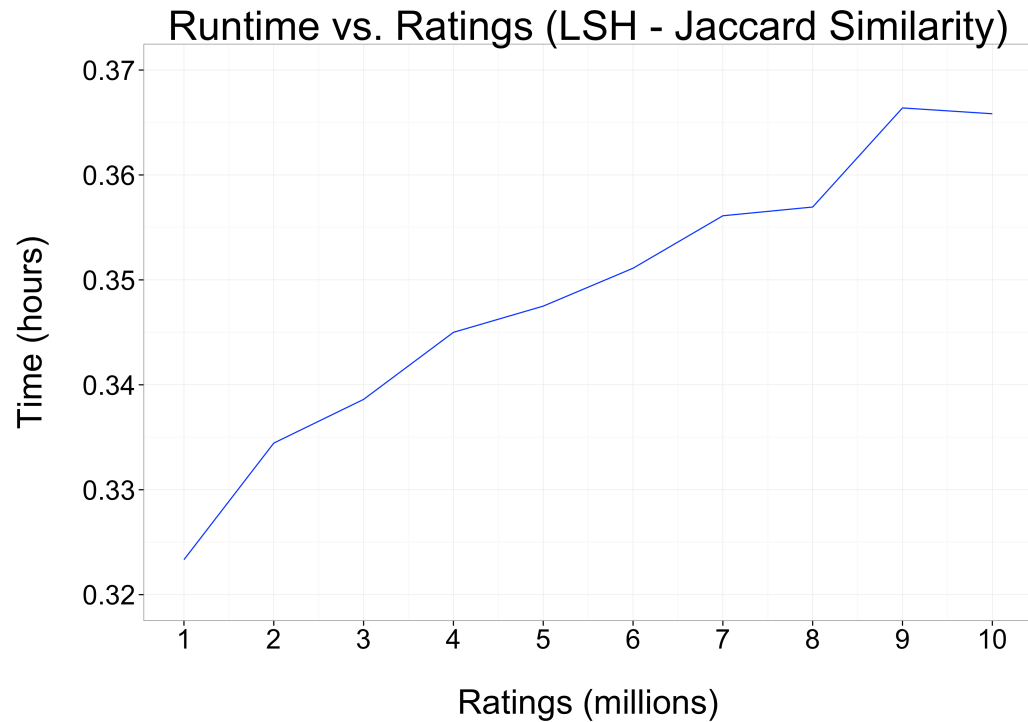


Short integer vectors that represent sets, and reflect their similarity

# LSH - Cosine Similarity



# LSH Results & Output





# Summary

- Simple collaborative filtering doesn't scale
- Went from  $O(n^2)$  to  $O(n)$  with ALS and LSH

Thank You!