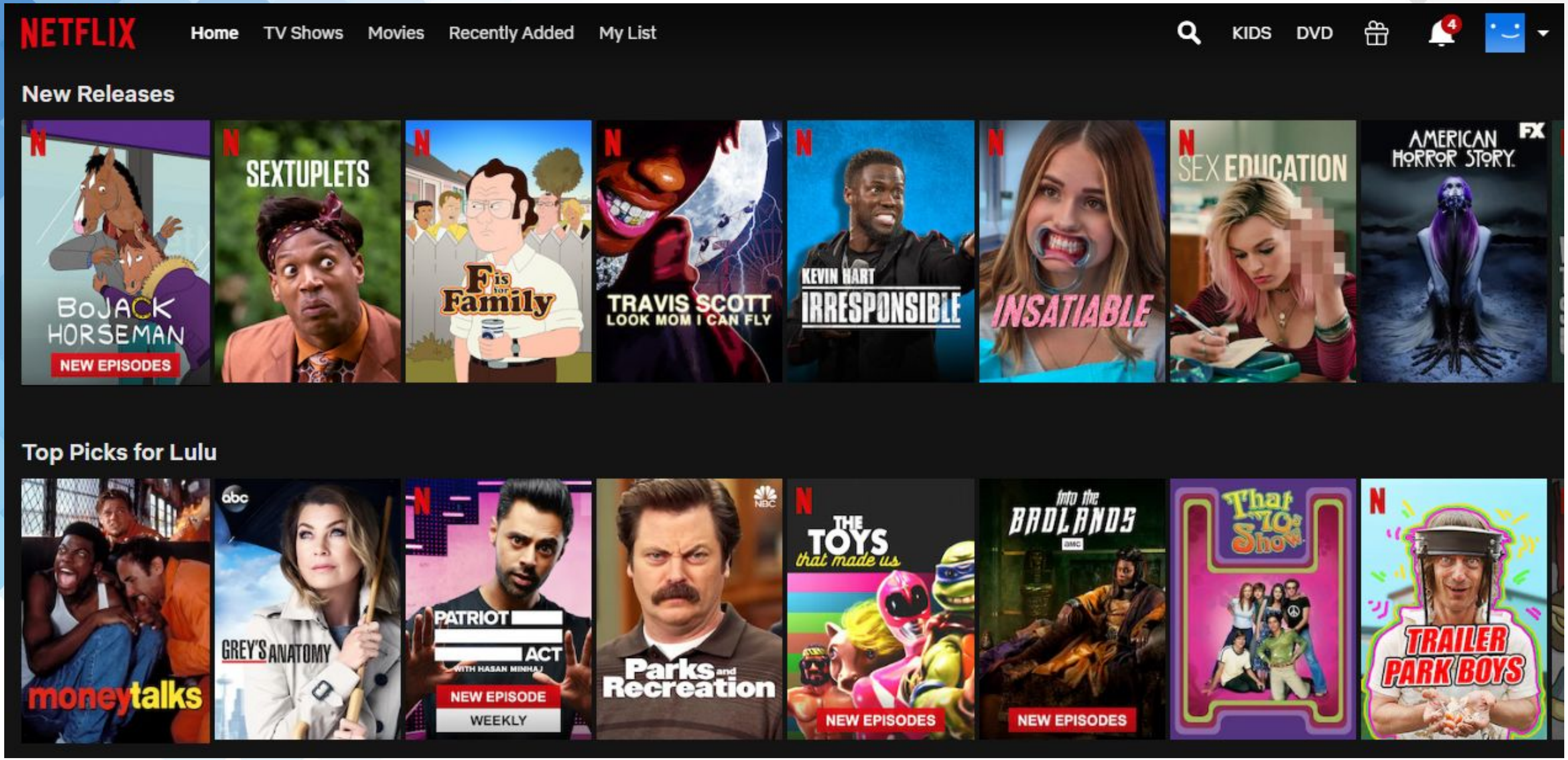
Collaborative Filtering Algorithms Evaluation

Gradient Descent with Probabilistic Assumptions VS Alternating Least Squares

Ting Cai, Qichao Chen, Lulu Dong, Kangkang Zhang


NETFLIX
Home
TV Shows
Movies
Recently Added
My List
KIDS
DVD
New Releases
BoJACK HORSEMAN
NEW EPISODES
SEXTUPLETS
F is for Family
TRAVIS SCOTT
LOOK MOM I CAN FLY
KEVIN HART
IRRESPONSIBLE
INSATIABLE
SEX EDUCATION
AMERICAN HORROR STORY
Top Picks for Lulu
money talks
GREY'S ANATOMY
PATRIOT ACT
WITH HASAN MINHAJ
NEW EPISODE
WEEKLY
Parks and Recreation
THE TOYS that made us
NEW EPISODES
Into the BADLANDS
NEW EPISODES
That '70s Show
TRAILER PARK BOYS

Objectives

- Algorithms:
  - Gradient Descent with Probabilistic Assumptions
  - Alternating Least Squares
- Compare Prediction Accuracy for those two algorithems before postprocessing
- Postprocessing:
  - SVD with kernel ridge regression
- Compare Prediction Accuracy for those two algorithems after postprocessing

Gradient Descent with Probabilistic Assumptions

user vector and movie vector follow gaussian distributions, $p_u \sim N(0, \sigma_p^2)$ $q_i \sim N(0, \sigma_q^2)$

the conditional distribution over the observed ratings:

$$r_{iu}|q_i, p_u, \sigma^2 \sim N(q_i^T p_u, \sigma^2)$$

According to Bayes theroem,

$$\begin{aligned} f(p, q|r) &= \frac{f(p, q, r)}{f(r)} \\ &= \frac{f(r|p, q) f(p) f(q)}{f(r)} \\ &\propto f(r|p, q) f(p) f(q) \end{aligned}$$

Gradient Descent with Probabilistic Assumptions (PMF)

Log-likelihood function:

$$\begin{aligned}\log f(p,q|r) &\propto \log\big(f(r|p,q)f(p)f(q)\big)\\ &= \log f(r|p,q) + \log f(p) + \log f(q)\\ &= -\frac{1}{\sigma^2}\sum_{i=1}^{M}\sum_{u=1}^{N} I_{iu}(r_{iu} - q_i^T p_u)^2 - \frac{1}{\sigma_q^2}\sum_{i=1}^{M}||q_i||^2 - \frac{1}{\sigma_p^2}\sum_{u=1}^{N}||p_u||^2 + C\end{aligned}$$

Can be converted to a minimum objective function by multiplying $-\frac{\sigma^2}{2}$

$$\frac{1}{2}\sum_{i=1}^{M}\sum_{u=1}^{N} I_{iu}(r_{iu} - q_i^T p_u)^2 + \frac{\sigma^2}{2\sigma_q^2}\sum_{i=1}^{M}||q_i||^2 + \frac{\sigma^2}{2\sigma_p^2}\sum_{u=1}^{N}||p_u||^2$$

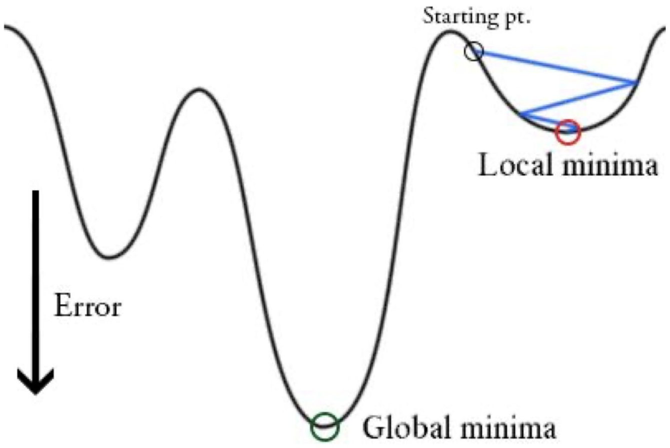
Gradient Descent with Probabilistic Assumptions

Update with learning rate α

$$p_u^{(t+1)} = p_u^{(t)} + \alpha\left(\sum_{i=1}^{M} I_{iu}(r_{iu} - q_i'^{(t)} p_u^{(t)}) q_i^{(t)} - \overbrace{\frac{\sigma^2}{\sigma_p^2}}^{\lambda_p} p_u^{(t)}\right)$$

$$q_i^{(t+1)} = q_i^{(t)} + \alpha\left(\sum_{u=1}^{N} I_{iu}(r_{iu} - q_i'^{(t)} p_u^{(t)}) p_u^{(t)} - \underbrace{\frac{\sigma^2}{\sigma_q^2}}_{\lambda_q} q_i^{(t)}\right)$$

However, for non-convex function, gradient descent might only find local minimum and cost lots of iterations.



Alternating Least Squares

$$min_{q^*p^*} \sum_{(u,i)\in K} (r_{ui} - q_i^T p_u)^2 + \lambda(\sum_i n_{q_i} ||q_i||^2 + \sum_u n_{p_u} ||p_u||^2)$$

- Step 1 Initialize matrix q by assigning the average rating for that movie as the first row, and small random numbers for the remaining entries.
- Step 2 Fix q, solve p by minimizing the objective function;
- Step 3 Fix p, solve q by minimizing the objective function similarly;
- Step 4 Repeat Steps 2 and 3 until a stopping criterion is satisfied.

$$pi = (Q_{Ii}Q_{Ii}{}^T + lambda * n_{pi} * E)^{-1} * Q_{Ii}R^T(i, Ii)$$

$$qj = (P_{Ij}P_{Ij}{}^T + lambda * n_{qj} * E)^{-1} * P_{Ij}R(Ij, j)$$

Postprocessing SVD with kernel ridge regression

Discard all weights p_{uk}

Define y as vector of ratings by users u;

X: for each row of X, normalized vector of factors for movie rated by user u

$$X_{.i} = \frac{q_i}{||q_i||}$$

$$y_{n\times 1} = X_{n\times f} \cdot \beta_{f\times 1}$$

Solve ridge regession:

$$\hat{\beta} = (X^T X + \lambda I)^{-1} X^T y$$

Prediction:

$$\hat{r}_i = K(x_i^T, X)(K(X, X) + \lambda I)^{-1} y$$

Experiment

Gradient Descent with Probabilistic Assumptions

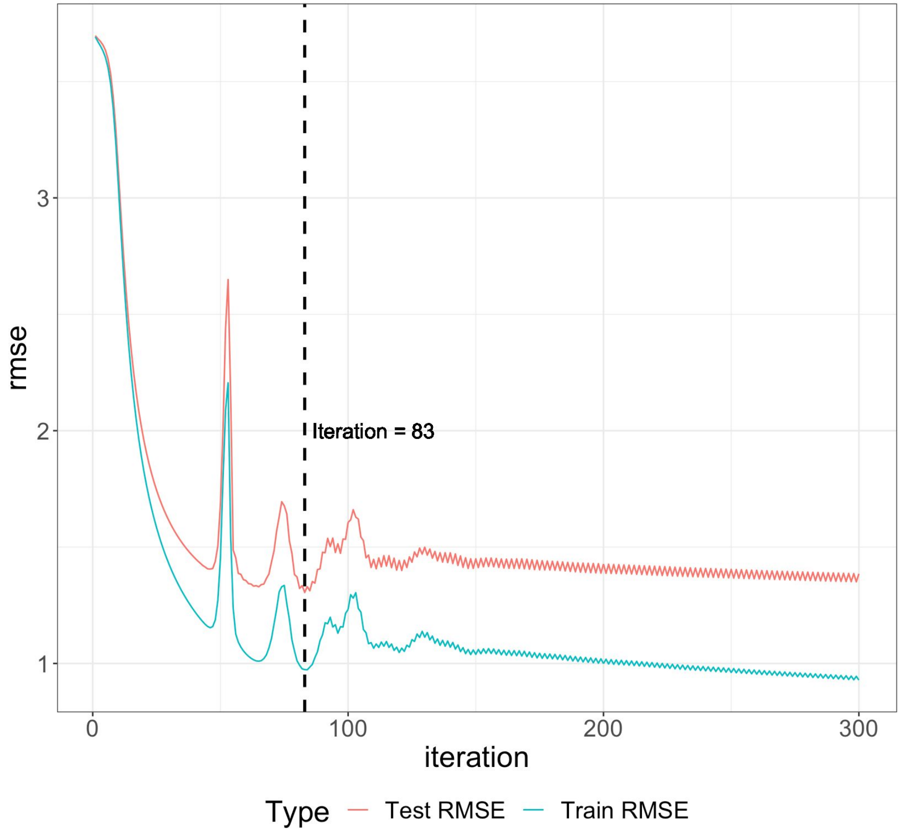
18 different parameter sets for 5-Fold CV

Result:

λ_p = 0.001

λ_q = 0.1

feature = 5



Experiment

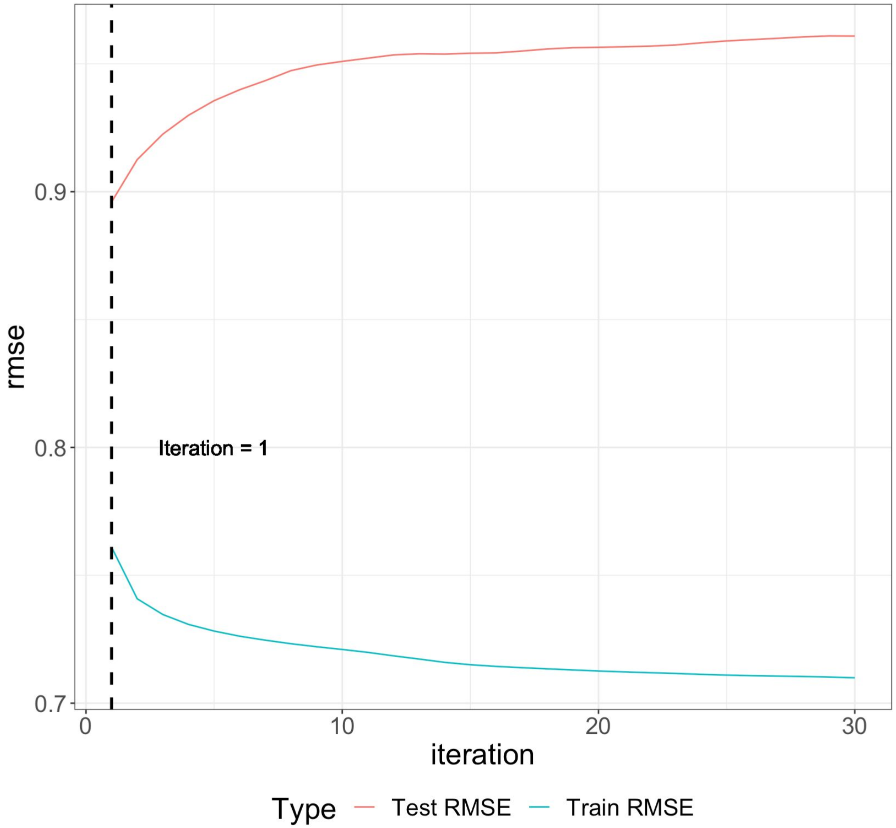
Alternating Least Squares

4 different parameter sets for 5-Fold CV

Result:

λ = 0.01

feature = 2



Postprocessing SVD with kernel ridge regression

Gaussian radial basis function with sigma = 0.05

$$k(x, x') = \exp(-\sigma \|x - x'\|^2)$$

For each user, build a kernel ridge regression using normalized movie vectors in train set as predictors

Evaluation

Root Mean Square Error (RMSE)

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - \hat{d}_i)^2}$$

d_i is the actual rating
$\hat{d}_i$ is the predicted rating
n is the amount of ratings

Evaluation Result

	Gradient Descent with Probabilistic Assumptions	Alternative Least Squares
RMSE before postprocessing	train: 0.974 test: 1.305	train: 0.770 test: 0.896
RMSE after postprocessing	train: 0.847 test: 0.924	train: 0.884 test : 0.956
Model training time	38 min 51 sec	9 min 15 sec

Conclusion

- Gradient descent with Probabilistic Assumptions
  - much more time-consuming than ALS
  - hard to find the global minimum
- RDF Kernel ridge regression
  - improves the accuracy of Gradient descent, while it decreases the accuracy of ALS
- Given RDF Kernel ridge regression:
  - gradient descent has better accuracy performance on both train and test sets
- Without RDF Kernel ridge regression
  - ALS has better accuracy performance on both train and test sets

Thanks for listening!