Maxios: Large-scale Nonnegative Matrix Factorization for Collaborative Filtering

UC Berkeley

Simon Shaolei Du, Yilin Liu, Boyi Chen, Lei Li

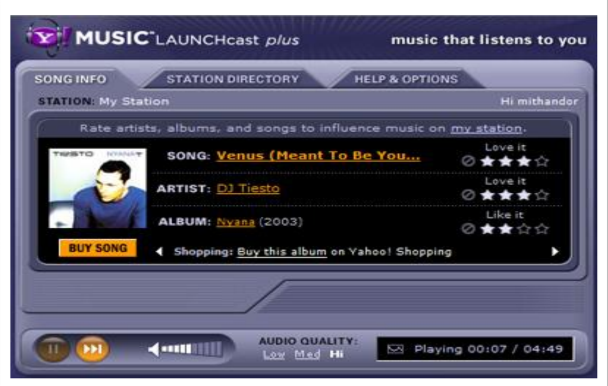
{simonshaoleidu,daniel.liu,bchen91}@berkeley.edu, lilei22@baidu.com

Baidu Research

Predicting ratings for recommendation

Movie ratings

Music ratings



Problem Description

Predicting missing values in User-Item matrix

	movie I	movie II	movie III	movie IV	...
User A	1	?	5	4	...
User B	?	2	3	?	...
User C	4	1	2	?	...
User D	?	5	1	3	...
User E	1	2	?	?	...
⋮	⋮	⋮	⋮	⋮	⋱

Problem characteristics:

- **Large scale**: millions of users, sub-millions of items
- **Highly Sparse**
- Need to **interpret** ratings
  - non-negativity constraints

Limitation of Existing Methods

- **EM based methods [Liu 2010]**: time consuming to compute a full user-item matrix (HUGE!) each iter
- **ALS based method [Zhang et al 2006, Kim & Choi 2009]:** costly update in each iteration
- **Multiplicative updates [Lee & Seung 1999]:** slow convergence

Proposed Maxios

Weighted NMF formulation:

$$\min_{U\geq 0, V\geq 0} ||A - W \odot (UV)||_F^2$$

$$W_{ij} = 1 \quad \text{if } A_{ij} \text{ is not missing}$$
$$= 0 \quad \text{otherwise}$$

$\odot$: elementwise product

Reformulation using ADMM

$$\min ||A - W \odot (UV)||_F^2 + I_+(X) + I_+(Y)$$
$$\text{s.t.} U = X, V = Y$$

Parallel update steps:

Update each row of U independently

$$U_i^{t+1} = ((A_i \odot W_i)(V^t \odot W_i^k)^T + \alpha X_i^t - \Lambda_i^t) \cdot ((V^t \odot W_i^k)(V^t \odot W_i^k)^T + \alpha I_k)^{-1}$$

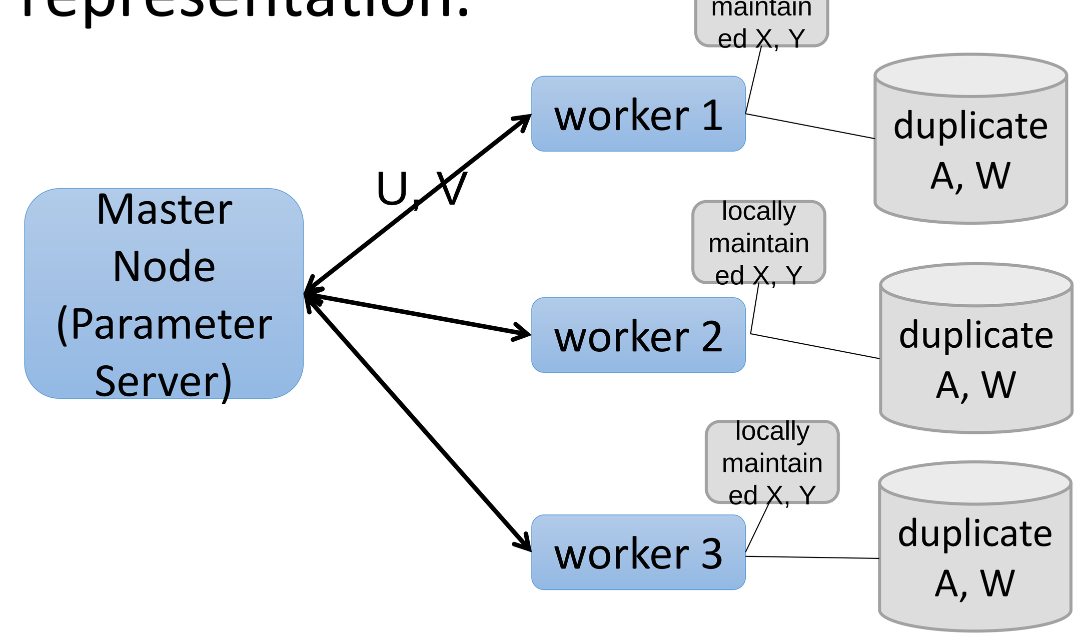
Update each row of X independently

$$X^{t+1} = P_+(U^{t+1} + \frac{\Lambda^t}{\alpha})$$

Updates for V and Y are in similar fashion.

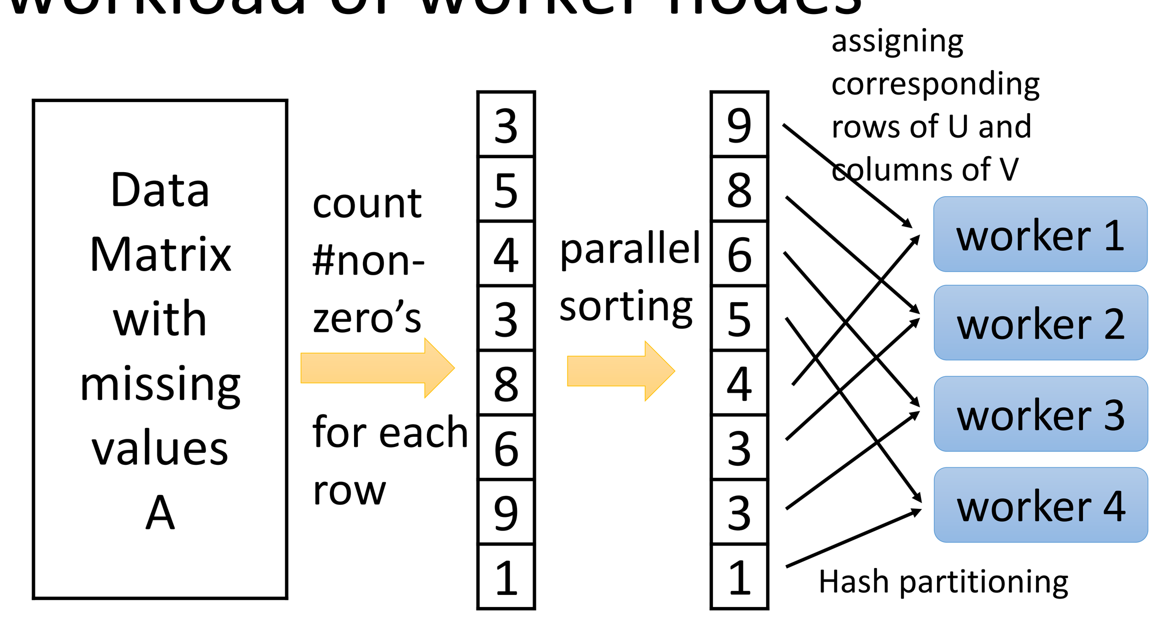
Implementation

Maxios is built on top of Spark, a distributed in-memory computing platform. Sparse data representation.



Workload Allocation

Preprocess to balance the workload of worker nodes

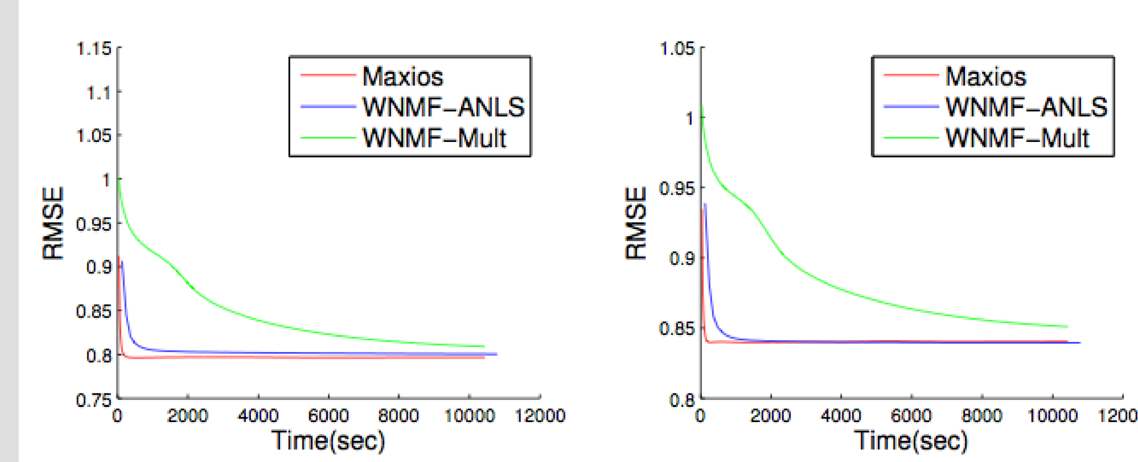


Experiments

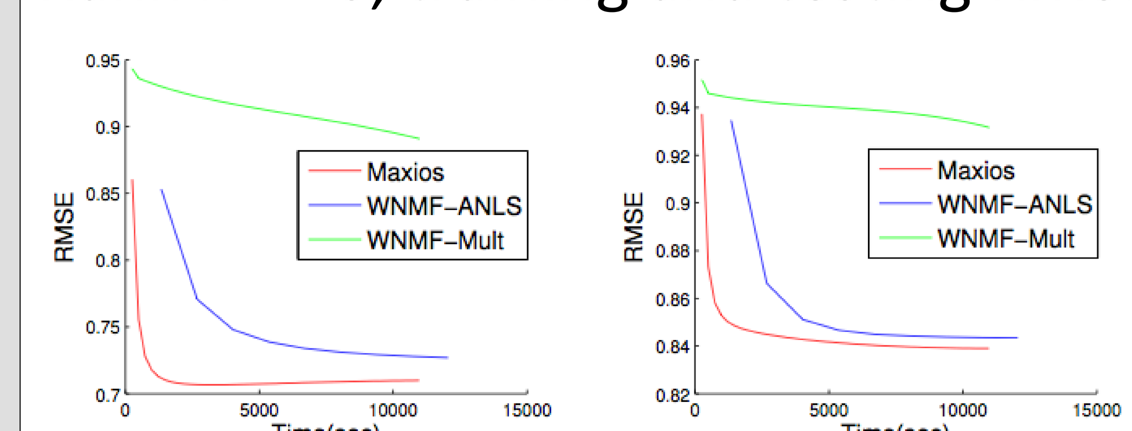
Data	users	Items	nnz	sparsity
Netflix	0.5M	17770	0.7B	1.18%
Yahoo	2M	98213	0.1B	0.06%

- Baseline Algorithms
  - Multiplicative Updates
  - Alternating Least Squares

Netflix Results

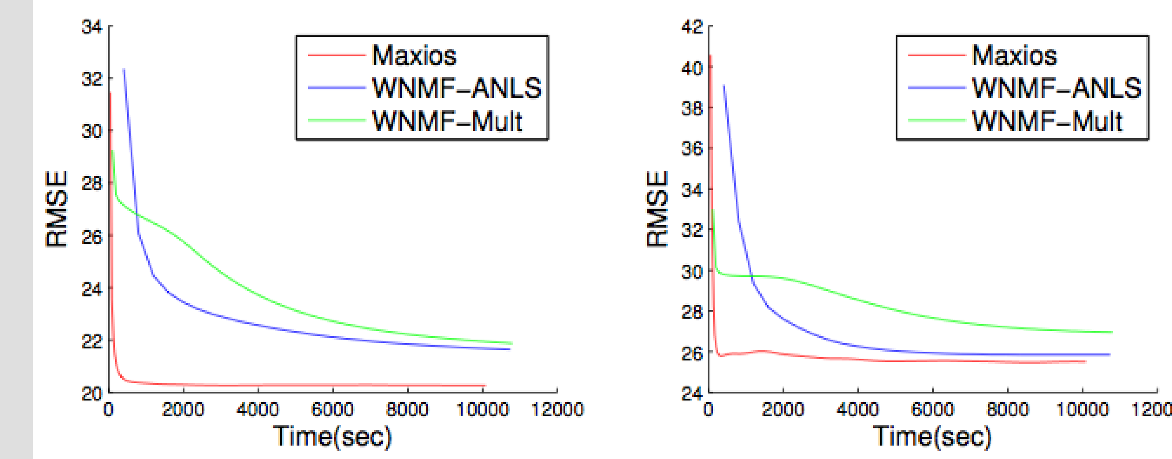


Rank k = 10, training and testing RMSE

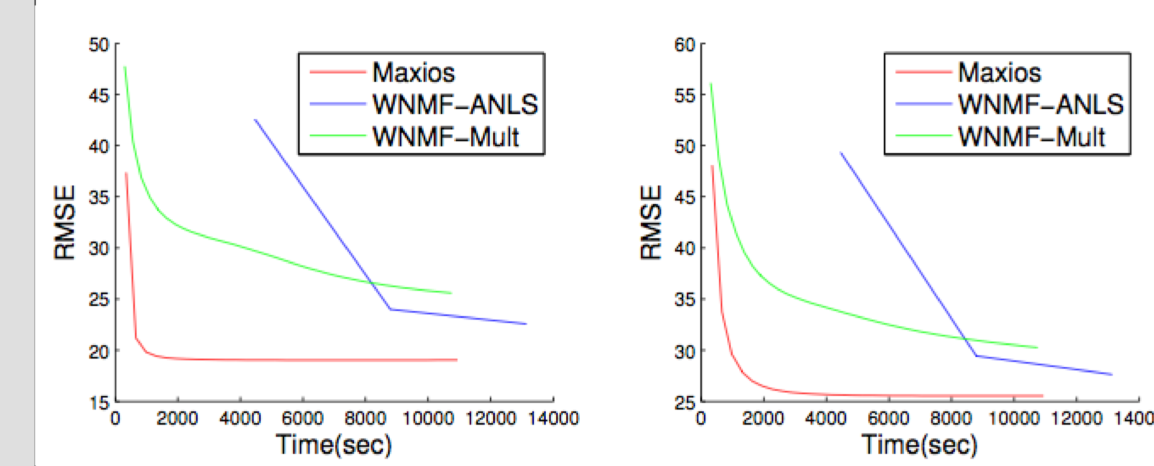


Rank k = 50, training and testing RMSE

Yahoo! Music Results



Rank k = 10, training and testing RMSE



Rank k = 50, training and testing RMSE

Contribution

Proposed a scalable NMF solution. Benefits of Maxios:

- Reducing computation overhead by utilizing sparsity. Weighted formulation avoids computing a User-Item matrix in each iteration.
- Highly scalable
  - independent update of each row of U,X and each column of V, Y
- Fast Updating
  - Maxios enables closed-form updates for U,V,X,Y via ADMM
  - benefits from distributed in-memory computing in Spark

Neural Information Processing Systems, 2014, Distributed Machine Learning and Matrix Computations workshop