Collaborative Filtering Algorithm with Stochastic Gradient Descent Optimization on Spark

JONGSE PARK

2016.12.13

Questions

1. What is Collaborative Filtering?
2. How do we train CF with Stochastic Gradient Descent Optimization?
3. How can we parallelize the optimization on Spark?

Recommender System

Group of Users

Group of Items

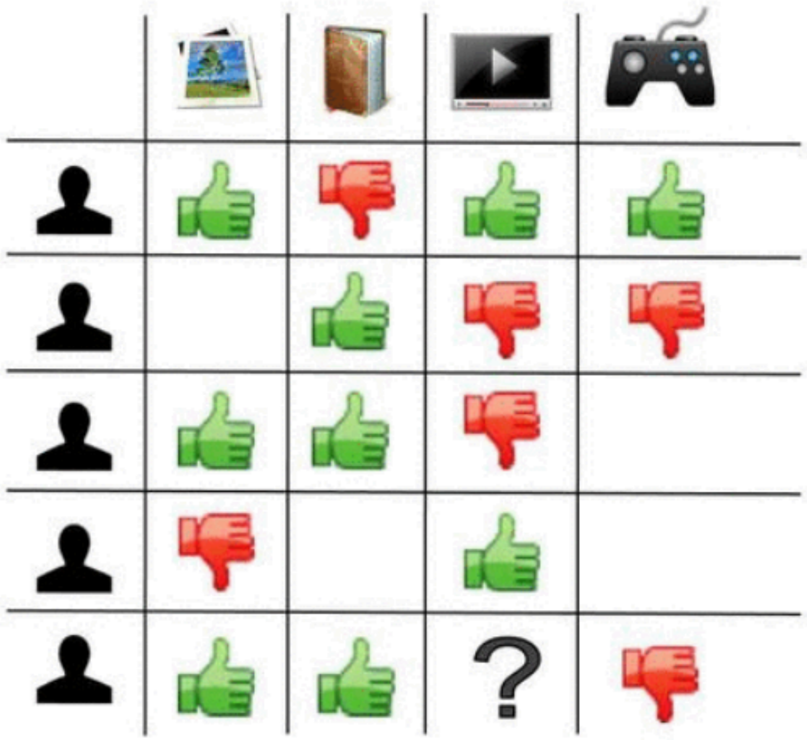
Collaborative Filtering

A method of making automatic predictions (**filtering**) about the interest of **a user** by collecting preferences or taste information from **many users** (**collaborating**).

Project Goal

Develop a **parallelized collaborative filtering algorithm using stochastic gradient descent optimization** on a state-of-the-art distributed system, **Spark**.

User-Item Matrix



Prediction

	i0	i1	i2	i3	i4
u0	3				
u1			5	4	3
u2					2
u3		2	5	1	2
u4				5	4
u5		5			

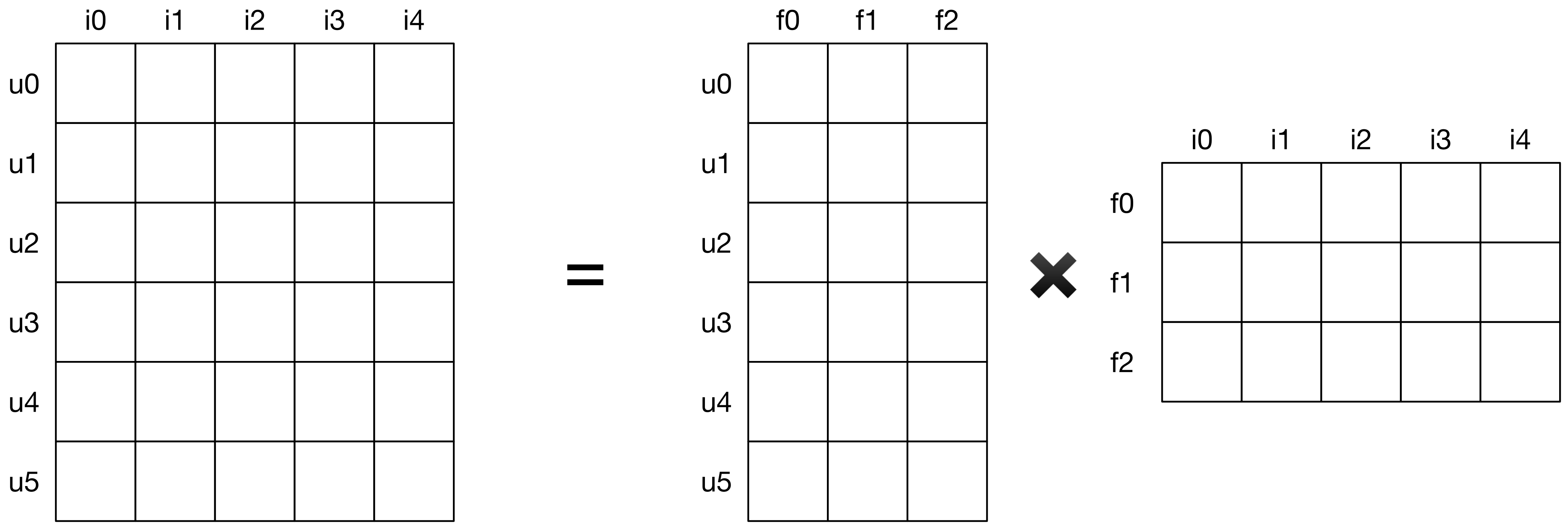
Prediction

	i0	i1	i2	i3	i4
u0	3				
u1			5	4	3
u2					2
u3		2	5	1	2
u4				5	4
u5		5			

→

	i0	i1	i2	i3	i4
u0	3	?	?	?	?
u1	?	?	5	4	3
u2	?	?	?	?	2
u3	?	2	5	1	2
u4	?	?	?	5	4
u5	?	5	?	?	?

Matrix Factorization



Matrix Factorization

	i0	i1	i2	i3	i4
u0					
u1					
u2					
u3					
u4					
u5					

=

	f0	f1	f2
u0	1.1	1.2	-3.1
u1	-3.1	4.1	4.1
u2	2.4	3.9	1.2
u3	-4.1	3.3	-2.8
u4	-4.9	-2.5	1.9
u5	4.5	1.3	3.0

×

	i0	i1	i2	i3	i4
f0	-4.1	1.2	5.0	-1.1	3.9
f1	-1.5	3.3	3.3	-0.7	2.5
f2	3.7	-0.9	-2.3	-0.2	1.1

Matrix Factorization

	f0	f1	f2
u0	1.1	1.2	-3.1
u1	-3.1	4.1	4.1
u2	2.4	3.9	1.2
u3	-4.1	3.3	-2.8
u4	-4.9	-2.5	1.9
u5	4.5	1.3	3.0

×

	i0	i1	i2	i3	i4
f0	-4.1	1.2	5.0	-1.1	3.9
f1	-1.5	3.3	3.3	-0.7	2.5
f2	3.7	-0.9	-2.3	-0.2	1.1

=

	i0	i1	i2	i3	i4
u0	-17.8	8.0	16.6	-1.43	3.87
u1	21.7	6.1	-11.4	-0.3	2.67
u2	-11.3	14.7	22.1	-5.6	20.4
u3	1.5	8.5	-3.2	2.7	-10.8
u4	30.8	-15.9	-37.1	6.7	-23.3
u5	-9.3	6.9	19.9	-6.5	24.1

Training

	i0	i1	i2	i3	i4
u0	-17.8	8.0	16.6	-1.43	3.87
u1	21.7	6.1	-11.4	-0.3	2.67
u2	-11.3	14.7	22.1	-5.6	20.4
u3	1.5	8.5	-3.2	2.7	-10.8
u4	30.8	-15.9	-37.1	6.7	-23.3
u5	-9.3	6.9	19.9	-6.5	24.1

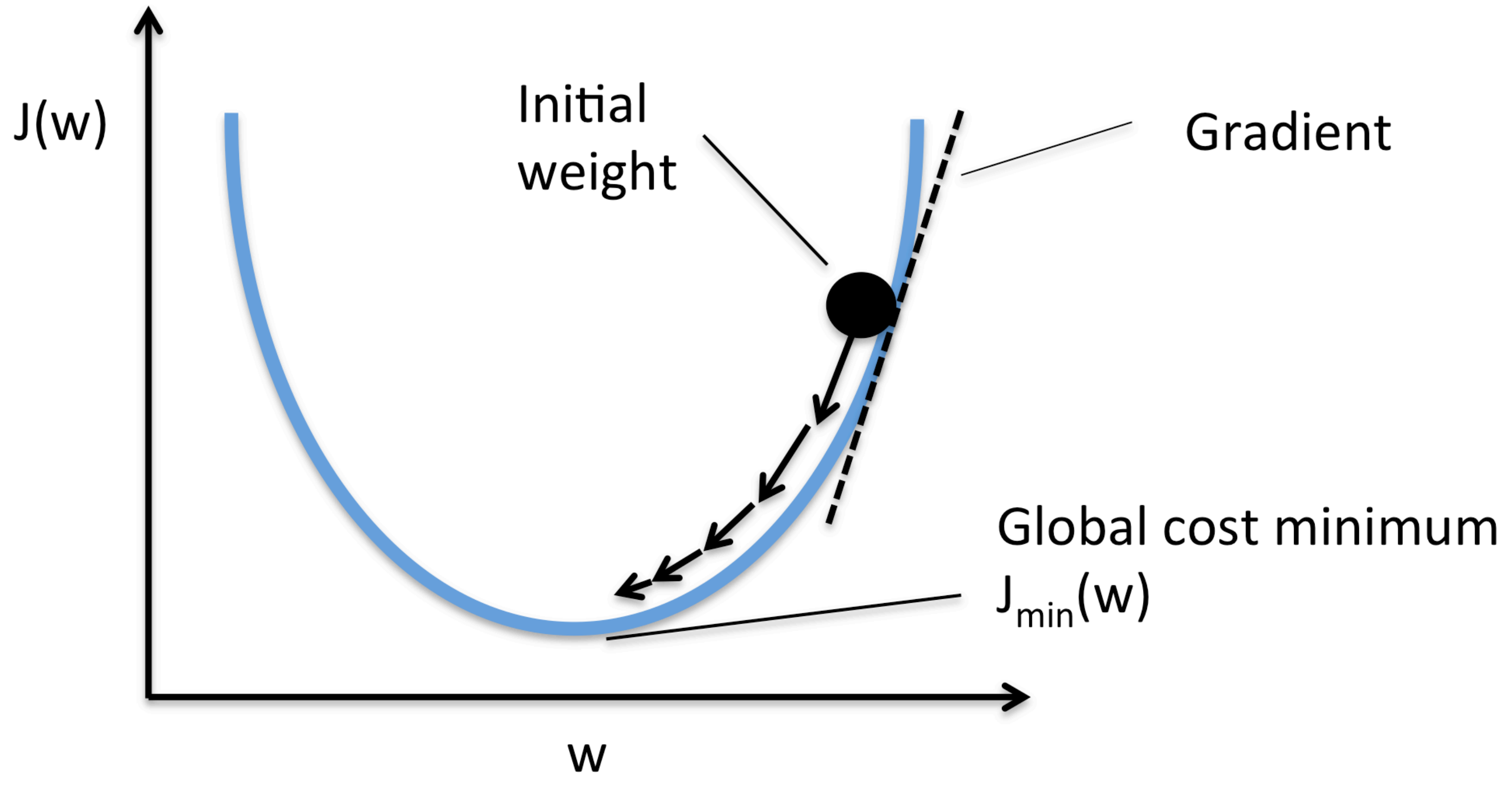
$\neq$

	i0	i1	i2	i3	i4
u0	3				
u1			5	4	3
u2					2
u3		2	5	1	2
u4				5	4
u5		5			

Optimization Problem

$$\text{Minimize} \quad \sum_{i,j} \{ \ Y_{ij} - W_j X_i \ \}$$

Gradient Descent



Stochastic Gradient Descent (SGD)

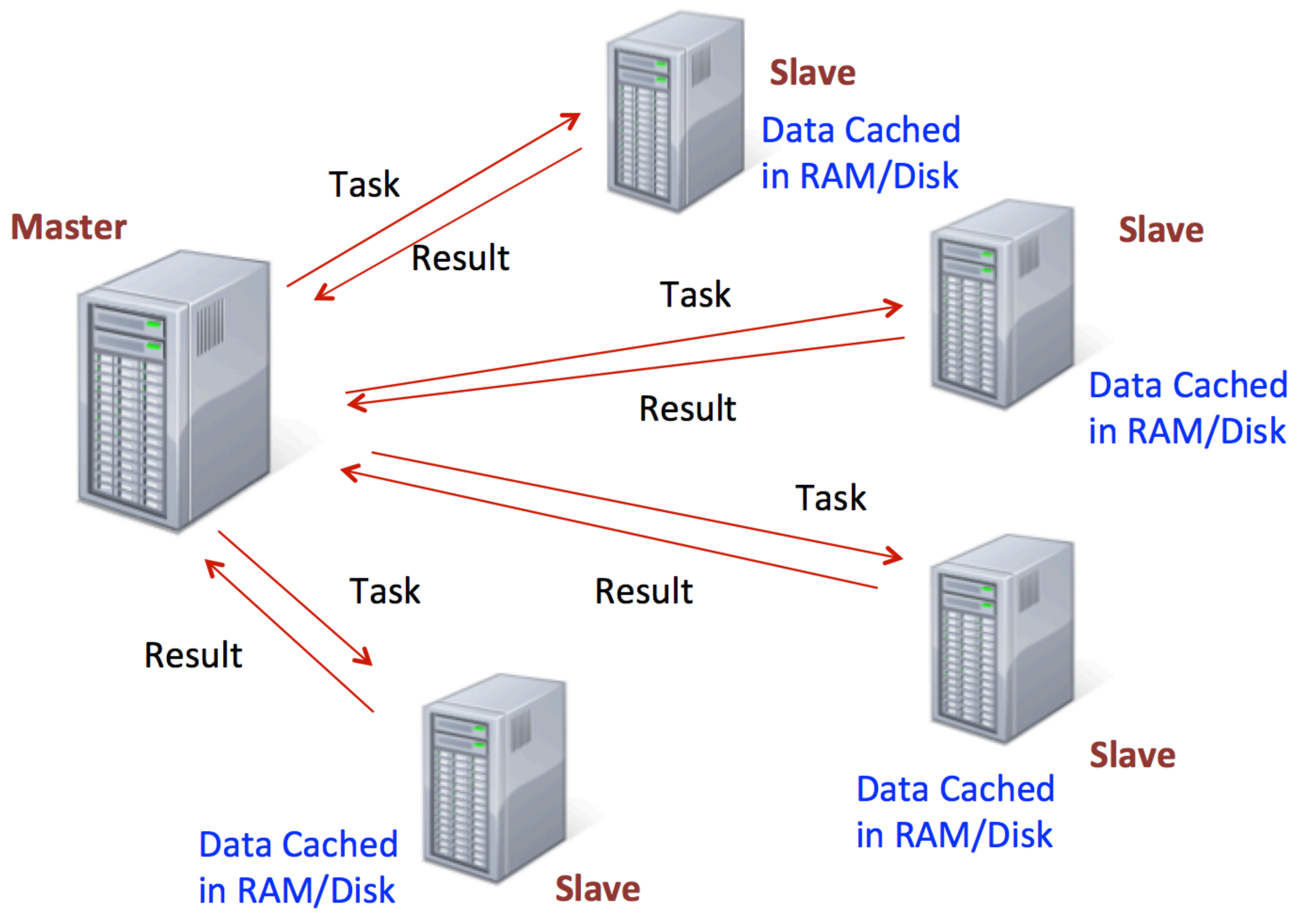
$$\text{Gradient} \;=\; \textstyle\sum_i \, (Y_{ij} - W_j X_i) X_i \;,\; \sum_j \, (Y_{ij} - W_j \, X_i) W_j$$

Instead of computing gradients for the entire dataset, SGD reads **an input data** and compute the gradient and update the feature matrices

Parallel SGD

- Gradient is a linear operation
- Model update can be run in parallel
- Workflow
  1. Broadcast the feature matrices
  2. Locally train the model on a subset of input dataset
  3. Aggregate the updates, average, and update the global model
  4. Repeat step 1 - 3

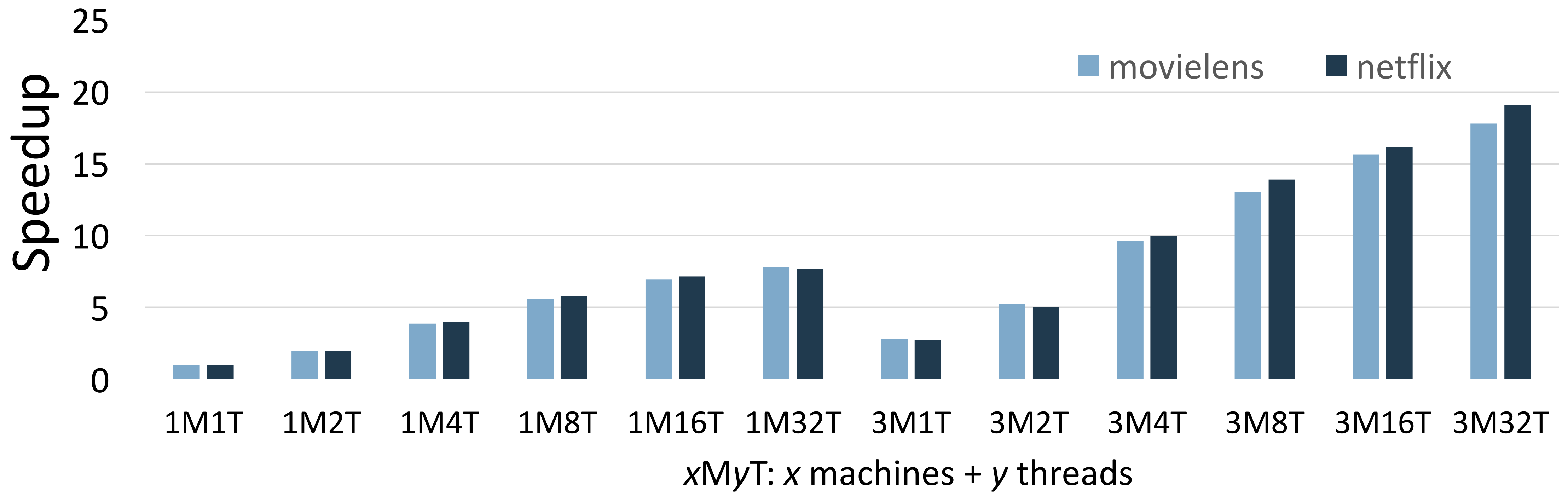
Spark


Master
Task
Result
Slave
Data Cached
in RAM/Disk
Task
Result
Slave
Data Cached
in RAM/Disk
Task
Result
Slave
Data Cached
in RAM/Disk
Task
Result
Slave
Data Cached
in RAM/Disk

Evaluation

- Three machines
  1. Intel Skylake Core i7 6700K Quad-core Processors
  2. 32G memory
- Hadoop Distributed File System
- 1Gbps interconnection network
- Movielens and NetFlix datasets

Performance Scaling



1M32T is **7.0x** faster than 1M1T

3M32T is **2.4x** faster than 1M32T

Conclusion

Developed collaborative filtering algorithm using parallel stochastic gradient descent optimization method on Spark

The experiment demonstrates that the implemented version of collaborative filtering algorithm is **scalable**

Thank you! Questions?