GENERALIZED LINEAR MODELS

IN

COLLABORATIVE FILTERING

HAO WU

STANFORD UNIVERSITY

Collaborative Filtering

	Alt. Linear Regression	Alt. Logistic Regression
application	direct feedbacks (rating)	indirect feedbacks (click)
distribution	normal	binomial
link function	$\boldsymbol{\mu} = \mathbf{U}\mathbf{M}^T$	$\log\left(\mathbf{P}/(1-\mathbf{P})\right) = \mathbf{U}\mathbf{M}^T$
loss function	square error	logistic loss
sufficient stat.	sum	# of 1s

Generalized Linear Models

$$\min_{\mathbf{U},\mathbf{M}} \quad \sum_{i=0}^{N} L(y_i, \mathbf{u}_{u_i}^T \mathbf{m}_{m_i})$$

$$+ \lambda\Big[\alpha\big(\sum_{i}^{n_U} \|\mathbf{u}_i\|_1 + \sum_{i}^{n_M} \|\mathbf{m}_i\|_1\big) + (1-\alpha)\big(\|\mathbf{U}\|_F + \|\mathbf{M}\|_F\big)\Big]$$

$$\text{s.t.} \qquad \mathbf{U} \in \mathbb{R}^{n_U \times k}, \ \mathbf{M} \in \mathbb{R}^{n_M \times k}$$

Distributed Algorithm

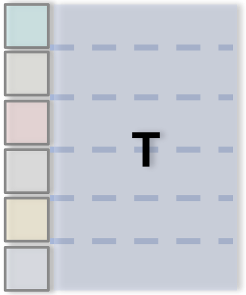
(assuming $n_m \times k$ fits in a single machine)

RDD[u, m, y] → RDD[u, map(m -> (t, n))]

Input entries

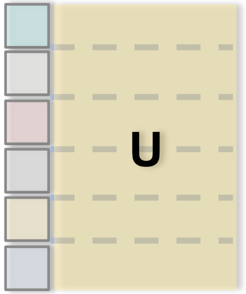
reduceByKey + aggregateByKey

Sufficient Stat.



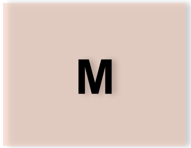
RDD[u, vector]

User features



Array[vector]

Movie features



for each iteration:

- Join **U** with **T** to from **D** (co-partitioned join)
- Update **M**
- Broadcast **M** (communication: $\log(p)(n_M k)$)
- Update **U**

Update M

for each movie:

- prepare dataframe by filter() and map() on **D**
- distributed logistic regression
  - LogisticRegression()
  - reduction and broadcast of size k

Update U

Map local logistic regression to users

- added local training method to LogisticRegression()
- no communication of data

Summary

- Sparsity is preserved with condensed entries
- Scales in n_U, but not n_M or k
- Communication cost: $\log(p)(n_M k)$
- Computational depth: $\log(n_U)(n_M k)$