Divide-and-Conquer Matrix Factorization

Lester Mackey†

Collaborators:

Ameet Talwalkar‡
Michael I. Jordan††

†Stanford University
‡UCLA
††UC Berkeley

December 14, 2015
Motivation: Large-scale Matrix Completion

Goal: Estimate a matrix $L_0 \in \mathbb{R}^{m \times n}$ given a subset of its entries

$$
\begin{bmatrix}
? & ? & 1 & \ldots & 4 \\
3 & ? & ? & \ldots & ? \\
? & 5 & ? & \ldots & 5
\end{bmatrix} \rightarrow 
\begin{bmatrix}
2 & 3 & 1 & \ldots & 4 \\
3 & 4 & 5 & \ldots & 1 \\
2 & 5 & 3 & \ldots & 5
\end{bmatrix}
$$

Examples

- Collaborative filtering: How will user $i$ rate movie $j$?
  - Netflix: 40 million users, 200K movies and television shows
- Ranking on the web: Is URL $j$ relevant to user $i$?
  - Google News: millions of articles, 1 billion users
- Link prediction: Is user $i$ friends with user $j$?
  - Facebook: 1.5 billion users
Goal: Estimate a matrix $L_0 \in \mathbb{R}^{m \times n}$ given a subset of its entries

\[
\begin{bmatrix}
? & ? & 1 & \ldots & 4 \\
3 & ? & ? & \ldots & ? \\
? & 5 & ? & \ldots & 5 \\
\end{bmatrix}
\rightarrow
\begin{bmatrix}
2 & 3 & 1 & \ldots & 4 \\
3 & 4 & 5 & \ldots & 1 \\
2 & 5 & 3 & \ldots & 5 \\
\end{bmatrix}
\]

State of the art MC algorithms
- **Strong estimation guarantees**
- **Plagued by expensive subroutines** (e.g., truncated SVD)

This talk
- Present divide and conquer approaches for **scaling up** any MC algorithm while **maintaining strong estimation guarantees**
Exact Matrix Completion

Goal: Estimate a matrix $L_0 \in \mathbb{R}^{m\times n}$ given a subset of its entries
**Goal:** Given entries from a matrix $M = L_0 + Z \in \mathbb{R}^{m \times n}$ where $Z$ is entrywise noise and $L_0$ has rank $r \ll m, n$, estimate $L_0$

- **Good news:** $L_0$ has $\sim (m + n)r \ll mn$ degrees of freedom

- **Factored form:** $AB^\top$ for $A \in \mathbb{R}^{m \times r}$ and $B \in \mathbb{R}^{n \times r}$

- **Bad news:** Not all low-rank matrices can be recovered

**Question:** What can go wrong?
What can go wrong?

Entire column missing

\[
\begin{bmatrix}
1 & 2 & ? & 3 & \ldots & 4 \\
3 & 5 & ? & 4 & \ldots & 1 \\
2 & 5 & ? & 2 & \ldots & 5 \\
\end{bmatrix}
\]

- No hope of recovery!

Solution: Uniform observation model

Assume that the set of $s$ observed entries $\Omega$ is drawn uniformly at random:

$\Omega \sim \text{Unif}(m, n, s)$
What can go wrong?

Bad spread of information

\[ L = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} [1] \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \]

- Can only recover \( L \) if \( L_{11} \) is observed

Solution: Incoherence with standard basis (Candès and Recht, 2009)

A matrix \( L = U \Sigma V^\top \in \mathbb{R}^{m \times n} \) with \( \text{rank}(L) = r \) is incoherent if

Singular vectors are not too skewed:

\[
\begin{align*}
\max_i \|UU^\top e_i\|^2 &\leq \frac{\mu r}{m} \\
\max_i \|VV^\top e_i\|^2 &\leq \frac{\mu r}{n}
\end{align*}
\]

and not too cross-correlated:

\[\|UV^\top\|_{\infty} \leq \sqrt{\frac{\mu r}{mn}}\]

(In this literature, it’s good to be incoherent)
How do we estimate $\mathbf{L}_0$?

First attempt:

$$\begin{align*}
\text{minimize}_A & \quad \text{rank}(A) \\
\text{subject to} & \quad \sum_{(i,j) \in \Omega} (A_{ij} - M_{ij})^2 \leq \Delta^2.
\end{align*}$$

Problem: Computationally intractable!

Solution: Solve convex relaxation (Fazel, Hindi, and Boyd, 2001; Candès and Plan, 2010)

$$\begin{align*}
\text{minimize}_A & \quad \|A\|_* \\
\text{subject to} & \quad \sum_{(i,j) \in \Omega} (A_{ij} - M_{ij})^2 \leq \Delta^2
\end{align*}$$

where $\|A\|_* = \sum_k \sigma_k(A)$ is the trace/nuclear norm of $A$.

Questions:

- Will the nuclear norm heuristic successfully recover $\mathbf{L}_0$?
- Can nuclear norm minimization scale to large MC problems?
Noisy Nuclear Norm Heuristic: Does it work?

Yes, with high probability.

Typical Theorem

If $L_0$ with rank $r$ is incoherent, $s \gtrsim r n \log^2(n)$ entries of $M \in \mathbb{R}^{m \times n}$ are observed uniformly at random, and $\hat{L}$ solves the noisy nuclear norm heuristic, then

$$\|\hat{L} - L_0\|_F \leq f(m, n) \Delta$$

with high probability when $\|M - L_0\|_F \leq \Delta$.

- See Candès and Plan (2010); Mackey, Talwalkar, and Jordan (2014b); Keshavan, Montanari, and Oh (2010); Negahban and Wainwright (2010)
- Implies exact recovery in the noiseless setting ($\Delta = 0$)
Noisy Nuclear Norm Heuristic: Does it scale?

Not quite...

- **Standard interior point methods** (Candès and Recht, 2009):
  \[ O(|\Omega|(m + n)^3 + |\Omega|^2(m + n)^2 + |\Omega|^3) \]

- More efficient, tailored algorithms:
  - Singular Value Thresholding (SVT) (Cai, Candès, and Shen, 2010)
  - Augmented Lagrange Multiplier (ALM) (Lin, Chen, Wu, and Ma, 2009a)
  - Accelerated Proximal Gradient (APG) (Toh and Yun, 2010)
  - All require rank-\(k\) truncated SVD on **every** iteration

**Take away:** These provably accurate MC algorithms are too expensive for large-scale or real-time matrix completion

**Question:** How can we scale up a given matrix completion algorithm and still retain estimation guarantees?
Divide-Factor-Combine (DFC)

Our Solution: Divide and conquer

1. Divide $M$ into submatrices.
2. Factor each submatrix in parallel.
3. Combine submatrix estimates to estimate $L_0$.

Advantages

- Submatrix completion is often much cheaper than completing $M$
- Multiple submatrix completions can be carried out in parallel
- DFC works with any base MC algorithm
- With the right choice of division and recombination, yields estimation guarantees comparable to those of the base algorithm
DFC-Proj: Partition and Project

1. Randomly partition $M$ into $t$ column submatrices
   
   \[ M = \begin{bmatrix} C_1 & C_2 & \cdots & C_t \end{bmatrix} \text{ where each } C_i \in \mathbb{R}^{m \times l} \]

2. Complete the submatrices **in parallel** to obtain
   
   \[ \begin{bmatrix} \hat{C}_1 & \hat{C}_2 & \cdots & \hat{C}_t \end{bmatrix} \]
   
   - **Reduced cost**: Expect $t$-fold speed-up per iteration
   - **Parallel computation**: Pay cost of one cheaper MC

3. Project submatrices onto a single low-dimensional column space
   
   Estimate column space of $L_0$ with column space of $\hat{C}_1$
   
   \[ \hat{L}^{proj} = \hat{C}_1 \hat{C}_1^+ \begin{bmatrix} \hat{C}_1 & \hat{C}_2 & \cdots & \hat{C}_t \end{bmatrix} \]
   
   - Common technique for randomized low-rank approximation
     
     (Frieze, Kannan, and Vempala, 1998)
   - **Minimal cost**: $O(mk^2 + lk^2)$ where $k = \text{rank}(\hat{L}^{proj})$

4. **Ensemble**: Project onto column space of each $\hat{C}_j$ and average
DFC: Does it work?

Yes, with high probability.

**Theorem** (Mackey, Talwalkar, and Jordan, 2014b)

If $L_0$ with rank $r$ is incoherent and $s = \omega \left( r^2 n \log^2(n) / \epsilon^2 \right)$ entries of $M \in \mathbb{R}^{m \times n}$ are observed uniformly at random, then $l = o(n)$ random columns suffice to have

$$\| \hat{L}^\text{proj} - L_0 \|_F \leq (2 + \epsilon) f(m, n) \Delta$$

with high probability when $\| M - L_0 \|_F \leq \Delta$ and the noisy nuclear norm heuristic is used as a base algorithm.

- Can sample vanishingly small fraction of columns ($l/n \to 0$)
- Implies exact recovery for noiseless ($\Delta = 0$) setting
- Analysis streamlined by matrix Bernstein inequality
Yes, with high probability.

Proof Ideas:

1. If $L_0$ is incoherent (has good spread of information), its partitioned submatrices are incoherent w.h.p.

2. Each submatrix has sufficiently many observed entries w.h.p.

$\Rightarrow$ Submatrix completion succeeds

3. Random submatrix captures the full column space of $L_0$ w.h.p.
   - Analysis builds on randomized $\ell_2$ regression work of Drineas, Mahoney, and Muthukrishnan (2008)

$\Rightarrow$ Column projection succeeds
Figure: Recovery error of DFC relative to base algorithm (APG) with $m = 10K$ and $r = 10$. 
Figure: Speed-up over base algorithm (APG) for random matrices with $r = 0.001m$ and 4% of entries revealed.
Task: Given a sparsely observed matrix of user-item ratings, predict the unobserved ratings

Issues

- Full-rank rating matrix
- Noisy, non-uniform observations

The Data

- Netflix Prize Dataset\(^1\)
  - 100 million ratings in \(\{1, \ldots, 5\}\)
  - 17,770 movies, 480,189 users

\(^1\)http://www.netflixprize.com/
Application: Collaborative filtering

**Task:** Predict unobserved user-item ratings

<table>
<thead>
<tr>
<th>Method</th>
<th>Netflix</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>Time</td>
</tr>
<tr>
<td>APG</td>
<td>0.8433</td>
<td>2653.1s</td>
</tr>
<tr>
<td>DFC-PROJ-25%</td>
<td>0.8436</td>
<td>689.5s</td>
</tr>
<tr>
<td>DFC-PROJ-10%</td>
<td>0.8484</td>
<td>289.7s</td>
</tr>
<tr>
<td>DFC-PROJ-ENS-25%</td>
<td>0.8411</td>
<td>689.5s</td>
</tr>
<tr>
<td>DFC-PROJ-ENS-10%</td>
<td>0.8433</td>
<td>289.7s</td>
</tr>
</tbody>
</table>
Goal: Given a matrix $M = L_0 + S_0 + Z$ where $L_0$ is low-rank, $S_0$ is sparse, and $Z$ is entrywise noise, recover $L_0$ (Chandrasekaran, Sanghavi, Parrilo, and Willsky, 2009; Candès, Li, Ma, and Wright, 2011; Zhou, Li, Wright, Candès, and Ma, 2010)

Examples:

- Background modeling/foreground activity detection

(Candès, Li, Ma, and Wright, 2011)
**Goal:** Given a matrix $M = L_0 + S_0 + Z$ where $L_0$ is low-rank, $S_0$ is sparse, and $Z$ is entrywise noise, recover $L_0$ (Chandrasekaran, Sanghavi, Parrilo, and Willsky, 2009; Candès, Li, Ma, and Wright, 2011; Zhou, Li, Wright, Candès, and Ma, 2010).

- $S_0$ can be viewed as an outlier/gross corruption matrix.
- Ordinary PCA breaks down in this setting.
- **Harder than MC:** outlier locations are unknown.
- **More expensive than MC:** dense, fully observed matrices.

---

**S_0** can be viewed as an outlier/gross corruption matrix.

- Ordinary PCA breaks down in this setting.

**Harder than MC:** outlier locations are unknown.

**More expensive than MC:** dense, fully observed matrices.
How do we recover $L_0$?

First attempt:

$$\min_{L,S} \quad \text{rank}(L) + \lambda \text{card}(S)$$

subject to $\|M - L - S\|_F \leq \Delta$.

**Problem:** Computationally intractable!

**Solution:** Convex relaxation

$$\min_{L,S} \quad \|L\|_* + \lambda \|S\|_1$$

subject to $\|M - L - S\|_F \leq \Delta$.

where $\|S\|_1 = \sum_{ij} S_{ij}$ is the $\ell_1$ entrywise norm of $S$.

**Question:** Does it work?

- Will noisy *Principal Component Pursuit (PCP)* recover $L_0$?

**Question:** Is it efficient?

- Can noisy PCP scale to large RMF problems?
Noisy Principal Component Pursuit: Does it work?

Yes, with high probability.

**Theorem** (Zhou, Li, Wright, Candès, and Ma, 2010)

If \( \mathbf{L}_0 \) with rank \( r \) is incoherent, and \( \mathbf{S}_0 \in \mathbb{R}^{m \times n} \) contains \( s \) non-zero entries with uniformly distributed locations, then if

\[
r = O\left(\frac{m}{\log^2 n}\right) \quad \text{and} \quad s \leq c \cdot mn,
\]

the minimizer to the problem

\[
\text{minimize}_{\mathbf{L}, \mathbf{S}} \quad \| \mathbf{L} \|_* + \lambda \| \mathbf{S} \|_1
\]

subject to \( \| \mathbf{M} - \mathbf{L} - \mathbf{S} \|_F \leq \Delta \).

with \( \lambda = 1/\sqrt{n} \) satisfies

\[
\| \hat{\mathbf{L}} - \mathbf{L}_0 \|_F \leq f(m, n)\Delta
\]

with high probability when \( \| \mathbf{M} - \mathbf{L}_0 - \mathbf{S}_0 \|_F \leq \Delta \).

See also Agarwal, Negahban, and Wainwright (2011)
Not quite...

- Standard interior point methods: $O(n^6)$ (Chandrasekaran, Sanghavi, Parrilo, and Willsky, 2009)

- More efficient, tailored algorithms:
  - Accelerated Proximal Gradient (APG) (Lin, Ganesh, Wright, Wu, Chen, and Ma, 2009b)
  - Augmented Lagrange Multiplier (ALM) (Lin, Chen, Wu, and Ma, 2009a)
  - Require rank-$k$ truncated SVD on every iteration
  - Best case $\text{SVD}(m, n, k) = O(mnk)$

Idea: Leverage the divide-and-conquer techniques developed for MC in the RMF setting
DFC: Does it work?

Yes, with high probability.

**Theorem** (Mackey, Talwalkar, and Jordan, 2014b)

If $L_0$ with rank $r$ is incoherent, and $S_0 \in \mathbb{R}^{m \times n}$ contains $s \leq c \cdot mn$ non-zero entries with uniformly distributed locations, then

$$l = O\left(\frac{r^2 \log^2(n)}{\epsilon^2}\right)$$

random columns suffice to have

$$\|\hat{L}^{proj} - L_0\|_F \leq (2 + \epsilon)f(m, n)\Delta$$

with high probability when $\|M - L_0 - S_0\|_F \leq \Delta$ and noisy principal component pursuit is used as the base algorithm.

- Can sample polylogarithmic number of columns
- Implies exact recovery for noiseless ($\Delta = 0$) setting
Figure: Estimation error of DFC and base algorithm (APG) with $m = 1K$ and $r = 10$. 
Figure: Speed-up over base algorithm (APG) for random matrices with $r = 0.01m$ and 10% of entries corrupted.
Application: Video background modeling

Task
- Each video frame forms one column of matrix $\mathbf{M}$
- Decompose $\mathbf{M}$ into stationary background $\mathbf{L}_0$ and moving foreground objects $\mathbf{S}_0$

Challenges
- Video is noisy
- Foreground corruption is often clustered, not uniform
Application: Video background modeling

**Example:** Significant foreground variation

**Specs**
- 1 minute of airport surveillance (Li, Huang, Gu, and Tian, 2004)
- 1000 frames, 25344 pixels
- Base algorithm: half an hour
- DFC: 7 minutes
**Example:** Changes in illumination

**Specs**
- 1.5 minutes of lobby surveillance (Li, Huang, Gu, and Tian, 2004)
- 1546 frames, 20480 pixels
- Base algorithm: 1.5 hours
- DFC: 8 minutes
Future Directions

New Applications and Datasets

- Practical problems with large-scale or real-time requirements
Example: Large-scale Affinity Estimation

**Goal:** Estimate semantic similarity between pairs of datapoints
- **Motivation:** Assign class labels to datapoints based on similarity

**Examples from computer vision**
- **Image tagging:** tree vs. firefighter vs. Tony Blair
- **Video / multimedia content detection:** wedding vs. concert

- Face clustering:

**Application:** Content detection, 9K YouTube videos, 20 classes
- **Baseline:** Low Rank Representation (Liu, Lin, and Yu, 2010)
  - **Strong guarantees** but 1.5 days to run
- **Divide and conquer** (Talwalkar, Mackey, Mu, Chang, and Jordan, 2013)
  - Comparable guarantees
  - Comparable performance in 1 hour (5 subproblems)
Future Directions

New Applications and Datasets
- Practical problems with large-scale or real-time requirements

New Divide-and-Conquer Strategies
- Other ways to reduce computation while preserving accuracy
DFC-Nys: Generalized Nyström Decomposition

1. Choose a random column submatrix $C \in \mathbb{R}^{m \times l}$ and a random row submatrix $R \in \mathbb{R}^{d \times n}$ from $M$. Call their intersection $W$.

   $$M = \begin{bmatrix} W & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, \quad C = \begin{bmatrix} W \\ M_{21} \end{bmatrix}, \quad R = \begin{bmatrix} W & M_{12} \end{bmatrix}$$

2. Recover the low rank components of $C$ and $R$ in parallel to obtain $\hat{C}$ and $\hat{R}$.

3. Recover $L_0$ from $\hat{C}$, $\hat{R}$, and their intersection $\hat{W}$

   $$\hat{L}^{nys} = \hat{C}\hat{W} + \hat{R}$$

   - Generalized Nyström method (Goreinov, Tyryshnikov, and Zamarashkin, 1997)
   - Minimal cost: $O(mk^2 + lk^2 + dk^2)$ where $k = \text{rank}(\hat{L}^{nys})$

4. Ensemble: Run $p$ times in parallel and average estimates.
Future Directions

New Applications and Datasets
- Practical problems with large-scale or real-time requirements

New Divide-and-Conquer Strategies
- Other ways to reduce computation while preserving accuracy
- More extensive use of ensembling

New Theory
- Analyze statistical implications of divide and conquer algorithms
  - Trade-off between statistical and computational efficiency
  - Impact of ensembling
- Developing suite of matrix concentration inequalities to aid in the analysis of randomized algorithms with matrix data
Concentration Inequalities

Matrix concentration

\[ \mathbb{P}\{\|X - \mathbb{E}X\| \geq t\} \leq \delta \]
\[ \mathbb{P}\{\lambda_{\text{max}}(X - \mathbb{E}X) \geq t\} \leq \delta \]

- Non-asymptotic control of random matrices with complex distributions

Applications

- Matrix completion from sparse random measurements
  (Gross, 2011; Recht, 2011; Negahban and Wainwright, 2010; Mackey, Talwalkar, and Jordan, 2014b)
- Randomized matrix multiplication and factorization
  (Drineas, Mahoney, and Muthukrishnan, 2008; Hsu, Kakade, and Zhang, 2011)
- Convex relaxation of robust or chance-constrained optimization
  (Nemirovski, 2007; So, 2011; Cheung, So, and Wang, 2011)
- Random graph analysis (Christofides and Markström, 2008; Oliveira, 2009)
Concentration Inequalities

Matrix concentration

\[ \mathbb{P}\{ \lambda_{\text{max}}(X - \mathbb{E}X) \geq t \} \leq \delta \]

Difficulty: Matrix multiplication is not commutative

\[ e^{X+Y} \neq e^X e^Y \neq e^Y e^X \]

Past approaches (Ahlswede and Winter, 2002; Oliveira, 2009; Tropp, 2011)

- Rely on deep results from matrix analysis
- Apply to sums of independent matrices and matrix martingales

Our work (Mackey, Jordan, Chen, Farrell, and Tropp, 2014a; Paulin, Mackey, and Tropp, 2015)

- Stein’s method of exchangeable pairs (1972), as advanced by Chatterjee (2007) for scalar concentration
  - Improved exponential tail inequalities
    (Hoeffding, Bernstein, Bounded differences)
  - Polynomial moment inequalities (Khintchine, Rosenthal)
  - Dependent sums and more general matrix functionals
**Example: Matrix Bounded Differences Inequality**

**Corollary (Paulin, Mackey, and Tropp, 2015)**

Suppose $Z = (Z_1, \ldots, Z_n)$ has independent coordinates, and 

$$
(H(z_1, \ldots, z_j, \ldots, z_n) - H(z_1, \ldots, z_j', \ldots, z_n))^2 \leq A_j^2
$$

for all $j$ and values $z_1, \ldots, z_n, z'_j$. Define the boundedness parameter

$$
\sigma^2 := \left\| \sum_{j=1}^{n} A_j^2 \right\|
$$

If each $A_j$ is $d \times d$, then, for all $t \geq 0$,

$$
P\{\lambda_{\text{max}}(H(Z) - \mathbb{E} H(Z)) \geq t\} \leq d \cdot e^{-t^2/(2\sigma^2)}.
$$

- Improves prior results in the literature (e.g., Tropp, 2011)
- Useful for analyzing
  - Multiclass classifier performance (Machart and Ralaivola, 2012)
  - Crowdsourcing accuracy (Dalvi, Dasgupta, Kumar, and Rastogi, 2013)
  - Convergence in non-differentiable optimization (Zhou and Hu, 2014)
Future Directions

New Applications and Datasets
- Practical problems with large-scale or real-time requirements

New Divide-and-Conquer Strategies
- Other ways to reduce computation while preserving accuracy
- More extensive use of ensembling

New Theory
- Analyze statistical implications of divide and conquer algorithms
  - Trade-off between statistical and computational efficiency
  - Impact of ensembling
- Developing suite of **matrix concentration inequalities** to aid in the analysis of randomized algorithms with matrix data
Future Directions

The End

Thanks!

\[ P_\Omega(M) \xrightarrow{\text{Divide}} P_\Omega(C_1) P_\Omega(C_2) \cdots P_\Omega(C_t) \xrightarrow{\text{Factor}} \hat{C}_1 \hat{C}_2 \cdots \hat{C}_t \xrightarrow{\text{Combine (Project)}} \hat{L}^{proj} \]

\[ P_\Omega(M) \xrightarrow{\text{Divide}} P_\Omega(C) P_\Omega(R) \xrightarrow{\text{Factor}} \hat{C} \hat{R} \xrightarrow{\text{Combine (Nyström)}} \hat{L}^{nys} \]
References


Talwalkar, Ameet, Mackey, Lester, Mu, Yadong, Chang, Shih-Fu, and Jordan, Michael I. Distributed low-rank subspace segmentation. December 2013.


