


# Global Convergence of Stochastic Gradient Descent for Some Non-convex Matrix Problems

Christopher De Sa   Kunle Olukotun   Christopher Ré  
{cdesa,kunle,chrismre}@stanford.edu

Stanford

July 7, 2015

# Stochastic Gradient Descent for Matrix Completion

- ▶ Matrix completion: ubiquitous problem
  - ▶ recover a low rank matrix from a series of samples
- ▶ SGD for matrix completion: commonly used in industry
  - ▶ IBM, Oracle, Twitter<sup>1</sup> , Jellyfish<sup>2</sup>


---

<sup>1</sup>Gupta et al., “WTF: The Who to Follow Service at Twitter”.

<sup>2</sup>Recht and Ré, “Parallel stochastic gradient algorithms for large-scale matrix completion”.

<sup>3</sup>E. Candès, Li, and Soltanolkotabi, “Phase Retrieval via Wirtinger Flow: Theory and Algorithms”; Jain, Netrapalli, and Sanghavi, “Low-rank Matrix Completion Using Alternating Minimization”.

# Stochastic Gradient Descent for Matrix Completion

- ▶ Matrix completion: ubiquitous problem
  - ▶ recover a low rank matrix from a series of samples
- ▶ SGD for matrix completion: commonly used in industry
  - ▶ IBM, Oracle, Twitter<sup>1</sup> , Jellyfish<sup>2</sup>
- ▶ Previous work: great local convergence results<sup>3</sup>
  - ▶ require initialization phase like SVD


---

<sup>1</sup>Gupta et al., “WTF: The Who to Follow Service at Twitter”.

<sup>2</sup>Recht and Ré, “Parallel stochastic gradient algorithms for large-scale matrix completion”.

<sup>3</sup>E. Candès, Li, and Soltanolkotabi, “Phase Retrieval via Wirtinger Flow: Theory and Algorithms”; Jain, Netrapalli, and Sanghavi, “Low-rank Matrix Completion Using Alternating Minimization”.

# Stochastic Gradient Descent for Matrix Completion

- ▶ Matrix completion: ubiquitous problem
  - ▶ recover a low rank matrix from a series of samples
- ▶ SGD for matrix completion: commonly used in industry
  - ▶ IBM, Oracle, Twitter<sup>1</sup> , Jellyfish<sup>2</sup>
- ▶ Previous work: great local convergence results<sup>3</sup>
  - ▶ require initialization phase like SVD

## Gap between theory and practice


- ▶ in theory: requires initialization step and/or other conditions
- ▶ in practice: works basically all the time

<sup>1</sup>Gupta et al., “WTF: The Who to Follow Service at Twitter”.

<sup>2</sup>Recht and Ré, “Parallel stochastic gradient algorithms for large-scale matrix completion”.

<sup>3</sup>E. Candès, Li, and Soltanolkotabi, “Phase Retrieval via Wirtinger Flow: Theory and Algorithms”; Jain, Netrapalli, and Sanghavi, “Low-rank Matrix Completion Using Alternating Minimization”.

# Stochastic Gradient Descent for Matrix Completion

- ▶ Matrix completion: ubiquitous problem
  - ▶ recover a low rank matrix from a series of samples
- ▶ SGD for matrix completion: commonly used in industry
  - ▶ IBM, Oracle, Twitter , Jellyfish
- ▶ Previous work: great local convergence results
  - ▶ require initialization phase like SVD

## Gap between theory and practice

- ▶ in theory: requires initialization step and/or other conditions
- ▶ in practice: works basically all the time

## Our Contribution

We show that this algorithm converges globally and give a rate!

- ▶ using random initialization

# Matrix Completion Problem

We take samples  $\tilde{A} \in \mathbb{R}^{n \times n}$  of a matrix  $A \in \mathbb{R}^{n \times n}$ .

Goal is to fit low-rank matrix  $X$  to samples:

$$\begin{aligned} \text{minimize} \quad & \mathbf{E} \left[ \left\| \tilde{A} - X \right\|_F^2 \right] \\ \text{subject to} \quad & X \in \mathbb{R}^{n \times n}, \mathbf{rank}(X) \leq 1, X \succeq 0. \end{aligned}$$

Apply *quadratic substitution*  $X = yy^T$  (Burer-Monteiro):

$$\begin{aligned} \text{minimize} \quad & \mathbf{E} \left[ \left\| \tilde{A} - yy^T \right\|_F^2 \right] \\ \text{subject to} \quad & y \in \mathbb{R}^n. \end{aligned}$$

# Multiple Applications of Matrix Completion

- ▶ standard matrix completion<sup>1</sup>
- ▶ matrix sensing<sup>2</sup>
- ▶ subspace tracking<sup>3</sup>

---

<sup>1</sup>E. J. Candès and Recht, “Exact Matrix Completion via Convex Optimization”.

<sup>2</sup>Jain, Netrapalli, and Sanghavi, “Low-rank Matrix Completion Using Alternating Minimization”; E. Candès, Li, and Soltanolkotabi, “Phase Retrieval via Wirtinger Flow: Theory and Algorithms”.

<sup>3</sup>Balzano, Nowak, and Recht, “Online identification and tracking of subspaces from highly incomplete information”.

# Multiple Applications of Matrix Completion

- ▶ standard matrix completion<sup>1</sup>  $\Leftrightarrow$  entrywise sampling
- ▶ matrix sensing<sup>2</sup>  $\Leftrightarrow$  trace sampling
- ▶ subspace tracking<sup>3</sup>  $\Leftrightarrow$  subspace sampling

## How to represent many applications?

different application  $\Leftrightarrow$  different noise model

- ▶ same optimization problem
- ▶ different distribution for  $\tilde{A}$

---

<sup>1</sup>E. J. Candès and Recht, “Exact Matrix Completion via Convex Optimization”.

<sup>2</sup>Jain, Netrapalli, and Sanghavi, “Low-rank Matrix Completion Using Alternating Minimization”; E. Candès, Li, and Soltanolkotabi, “Phase Retrieval via Wirtinger Flow: Theory and Algorithms”.

<sup>3</sup>Balzano, Nowak, and Recht, “Online identification and tracking of subspaces from highly incomplete information”.



## How to handle many applications?

different application  $\iff$  different noise model

Use only weak assumptions about the samples  $\tilde{A}$ :

- ▶  $\tilde{A}$  is an *unbiased estimator* for  $A$ .
- ▶ Bound only the variance of  $\tilde{A}$ .

Also: easily handle additional noise.

- 1 Analyzing the Problem
- 2 Our Version of SGD
- 3 Proving Convergence
- 4 Experiments

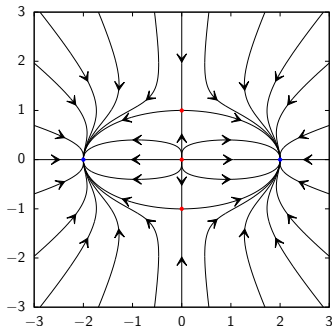
- 1** Analyzing the Problem
- 2 Our Version of SGD
- 3 Proving Convergence
- 4 Experiments

# Gradient Flow for 2D Case

Simple gradient flow for 2D case:

$$\dot{x} = 4x - (x^2 + y^2)x$$

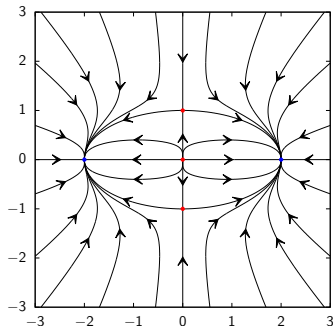
$$\dot{y} = y - (x^2 + y^2)y$$



# Non-Convexity in Gradient Flow

Consequences of non-convexity:

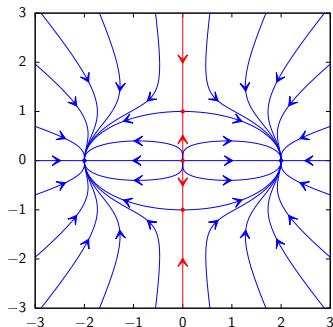
- ▶ we get pushed in different directions
- ▶ there are multiple unstable fixed points



# Bad Trajectory

If we initialize on a bad (red) trajectory, we may be unable to escape.

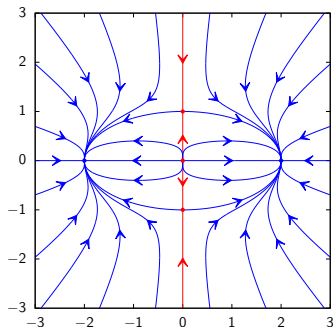
- ▶ even near bad trajectory, may take a long time to converge
- ▶ consequence of unstable fixed points



# Consequences for SGD

Using weak noise model:

- ▶ can't show convergence from everywhere in reasonable time
  - ▶ can't show convergence from initial points near bad trajectory
- ▶ can't show almost sure convergence from anywhere
  - ▶ algorithm can always “jump” onto bad trajectory
  - ▶ then stay there for an arbitrarily long time



# Outline

- 1 Analyzing the Problem
- 2 Our Version of SGD**
- 3 Proving Convergence
- 4 Experiments



Standard SGD gives the update rule

$$y_{k+1} = y_k - 4\alpha_k \left( y_k y_k^T y_k - \tilde{A}_k y_k \right).$$

By using a variable step size scheme, update rule becomes:

$$y_{k+1} = \left( 1 + \eta \tilde{A}_k \right) y_k \left( 1 + \eta \|y_k\|^2 \right)^{-1}.$$

## Alecton Update Rule

$$y_{k+1} = \left( 1 + \eta \tilde{A}_k \right) y_k.$$

# Algorithm Alepton

## Algorithm Alepton Overview

- ▶ (Initialization) Do uniform random initialization such that  $\|y_0\| = 1$ .
- ▶ (Angular Phase) Run SGD with the Alepton update rule to recover the angular component.

$$y_{k+1} = \left(1 + \eta \tilde{A}_k\right) y_k.$$

- ▶ (Radial Phase) Use averaging to recover the radial component.

All steps of algorithm simple; easy to compute.

# Outline

- 1 Analyzing the Problem
- 2 Our Version of SGD
- 3 Proving Convergence**
- 4 Experiments

# Analyze Non-Convex SGD Using Martingales

Using a standard Lyapunov-function approach won't work.

- ▶ this approach shows convergence from everywhere, which we've shown doesn't happen

Martingale approach:

- ▶ handles processes which can fail with some probability
- ▶ bounds the probability of failure

## Success condition

$$\rho_k = \frac{(u_1^T y_k)^2}{\|y_k\|^2} \geq 1 - \epsilon,$$

where  $u_1$  is the dominant eigenvector of  $A$ .

We let  $F_t$ , the *failure event*, denote the event that success has not occurred by iteration  $t$ .

# Only Constrain Second Moment of Samples

## Second Moment Constraint

$$\mathbf{E} \left[ \left( y^T \tilde{\mathbf{A}} z \right)^2 \right] \leq \sigma^2 \|y\|^2 \|z\|^2.$$

# Convergence Result

## Theorem

For any  $\chi > 0$ , if we run for  $t$  iterations where

$$t \geq \frac{n \log n}{\epsilon} F(\sigma, A) G(\chi),$$

then the probability of failure is bounded by

$$\mathbf{P}(F_t) \leq \chi.$$

## Takeaway point

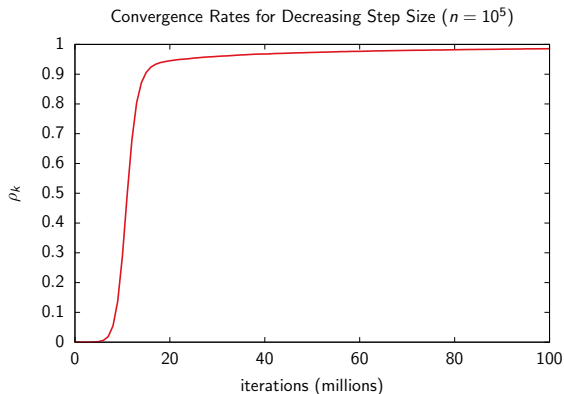
- ▶ The  $n \log n$  part is standard for matrix completion algorithms.
- ▶ The  $\epsilon^{-1}$  is typical for even convex SGD.
- ▶ For all applications we looked at,  $F(\cdot)$  is independent of  $n$ .

# Outline

- 1 Analyzing the Problem
- 2 Our Version of SGD
- 3 Proving Convergence
- 4 Experiments**



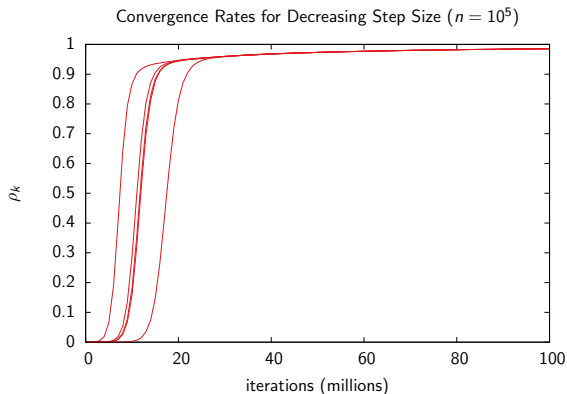
# Alecton Converges for Large Datasets



Convergence trajectory of Alecton for entrywise sampling.

- ▶ 1.5 GB sparse dataset ( $A \in \mathbb{R}^{10^5 \times 10^5}$ )

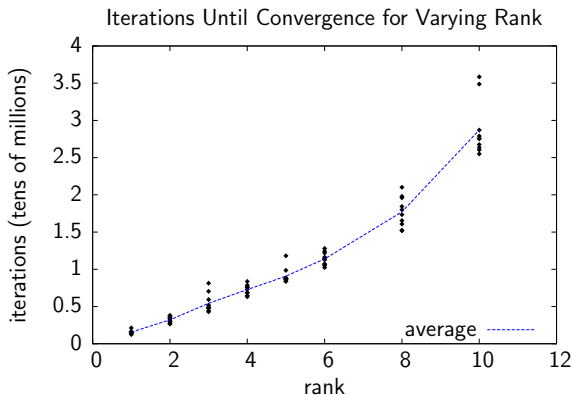
# Alecton Converges for Large Datasets



Convergence trajectory of Alecton for entrywise sampling.

- ▶ 1.5 GB sparse dataset ( $A \in \mathbb{R}^{10^5 \times 10^5}$ )
- ▶ convergence time varies for different initializations

# Alecton Also Works for Higher-Rank Recovery



Iterations until convergence for recovering higher-rank estimates.

- ▶ same 1.5 GB sparse entrywise sampling dataset
- ▶ good practical scaling with rank

# Conclusion

- ▶ SGD for Matrix completion is a ubiquitous algorithm.
  - ▶ we provide a global convergence result
  - ▶ typically requires  $t = O(n\epsilon^{-1} \log n)$  timesteps
- ▶ Result applies to many applications
  - ▶ collaborative filtering, subspace tracking, matrix sensing, etc.
- ▶ Scales well to big data!

# Conclusion

- ▶ SGD for Matrix completion is a ubiquitous algorithm.
  - ▶ we provide a global convergence result
  - ▶ typically requires  $t = O(n\epsilon^{-1} \log n)$  timesteps
- ▶ Result applies to many applications
  - ▶ collaborative filtering, subspace tracking, matrix sensing, etc.
- ▶ Scales well to big data!

**Thank you!**

**Questions?**

contact: [cdesa@stanford.edu](mailto:cdesa@stanford.edu)