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

Christopher De Sa CDESA@STANFORD.EDU
Department of Electrical Engineering, Stanford University, Stanford, CA 94309

Kunle Olukotun KUNLE@STANFORD.EDU
Department of Electrical Engineering, Stanford University, Stanford, CA 94309

Christopher Ré CHRISMRE@STANFORD.EDU
Department of Computer Science, Stanford University, Stanford, CA 94309

Abstract

Stochastic gradient descent (SGD) on a low-rank factorization (Burer & Monteiro, 2003) is commonly employed to speed up matrix problems including matrix completion, subspace tracking, and SDP relaxation. In this paper, we exhibit a step size scheme for SGD on a low-rank least-squares problem, and we prove that, under broad sampling conditions, our method converges globally from a random starting point within $O(\epsilon^{-1} n \log n)$ steps with constant probability for constant-rank problems. Our modification of SGD relates it to stochastic power iteration. We also show experiments to illustrate the runtime and convergence of the algorithm.

1. Introduction

We analyze an algorithm to solve the stochastic optimization problem

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

where p is an integer and $\tilde{A}$ is a symmetric matrix drawn from some distribution with bounded covariance. The solution to this problem is the matrix formed by zeroing out all but the largest p positive eigenvalues of the matrix $\mathbf{E}[\tilde{A}]$. This problem, or problems that can be transformed to this problem, appears in a variety of machine learning applications including matrix completion (Jain et al., 2013; Teflioudi et al., 2012; Chen et al., 2011), general data analysis (Zou et al., 2004), subspace tracking (Balzano et al., 2010), principle component analysis (Arora et al., 2012), optimization (Burer & Monteiro, 2005; Journée et al., 2010; Mishra et al., 2013; Horstmeyer et al., 2014), and recommendation systems (Gupta et al., 2013; Oscar Boykin, 2013-2014).

Sometimes, (1) arises under conditions in which the samples $\tilde{A}$ are sparse, but the matrix X would be too large to store and operate on efficiently; a standard heuristic to use in this case is a low-rank factorization (Burer & Monteiro, 2003). The idea is to substitute $X = YY^T$ and solve the problem

$$\begin{array}{ll} \text{minimize} & \mathbf{E}\left[\left\|\tilde{A} - YY^T\right\|_F^2\right] \\ \text{subject to} & Y \in \mathbb{R}^{n\times p}. \end{array} \tag{2}$$

By construction, if we set $X = YY^T$, then $X \in \mathbb{R}^{n\times n}$, $\mathbf{rank}\left(X\right) \leq p$, and $X \succeq 0$; this allows us to drop these constraints. Instead of having to store the matrix X (of size n^2), we only need to store the matrix Y (of size np).

In practice, many people use stochastic gradient descent (SGD) to solve (2). Efficient SGD implementations can scale to very large datasets (Recht & Ré, 2013; Niu et al., 2011; Teflioudi et al., 2012; Agarwal et al., 2011; Bottou, 2010; Duchi et al., 2011; Bottou & Bousquet, 2008; Hu et al., 2009). However, standard stochastic gradient descent on (2) does not converge globally, in the sense that there will always be some initial values for which the norm of the iterate will diverge .

People have attempted to compensate for this with sophisticated methods like geodesic step rules (Journée et al., 2010) and manifold projections (Absil et al., 2008); however, even these methods cannot guarantee global convergence. Motivated by this, we describe Alecton, an algo-

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

rithm for solving (2), and analyze its convergence. Alecton is an SGD-like algorithm that has a simple update rule with a step size that is a simple function of the norm of the iterate Y_k. We show that Alecton converges globally. We make the following contributions:

- We establish the convergence rate to a global optimum of Alecton using a random initialization; in contrast, prior analyses (Candès et al., 2014; Jain et al., 2013) have required more expensive initialization methods, such as the singular value decomposition of an empirical average of the data.
- In contrast to previous work that uses bounds on the magnitude of the noise (Hardt & Price, 2014; Hardt, 2014), our analysis depends only on the variance of the samples. As a result, we are able to be robust to different noise models, and we apply our technique to these problems, which did not previously have global convergence rates:
  - *matrix completion*, in which we observe entries of A one at a time (Jain et al., 2013; Keshavan et al., 2010) (Section 4.1),
  - *phase retrieval*, in which we observe $\mathrm{tr}(u^T A v)$ for randomly selected u, v (Candès et al., 2014; Candès & Li, 2014) (Section 4.3), and
  - *subspace tracking*, in which A is a projection matrix and we observe random entries of a random vector in its column space (Balzano et al., 2010) (Section 4.4).

  Our result is also robust to different noise models.
- We describe a martingale-based analysis technique that is novel in the space of non-convex optimization. We are able to generalize this technique to some simple regularized problems, and we are optimistic that it has more applications.

1.1. Related Work

Much related work exists in the space of solving low-rank factorized optimization problems. Foundational work in this space was done by Burer and Monteiro (Burer & Monteiro, 2003; 2005), who analyzed the low-rank factorization of general semidefinite programs. Their results focus on the classification of the local minima of such problems, and on conditions under which no non-global minima exist. They do not analyze the convergence rate of SGD.

Another general analysis in Journée et al. (2010) exhibits a second-order algorithm that converges to a local solution. Their results use manifold optimization techniques to optimize over the manifold of low-rank matrices. These approaches have attempted to correct for falling off the manifold using Riemannian retractions (Journée et al., 2010), geodesic steps (Balzano et al., 2010), or projections back onto the manifold. General non-convex manifold optimization techniques (Absil et al., 2008) tell us that first-order methods, such as SGD, will converge to a fixed point, but they provide no convergence rate to the global optimum. Our algorithm only involves a simple rescaling, and we are able to provide global convergence results.

Our work follows others who have studied individual problems that we consider. Jain et al. (2013) study matrix completion and provides a convergence rate for an exact recovery algorithm, alternating minimization; subsequent work (Jain & Netrapalli, 2014) gives fast rates for projected gradient descent. Candès et al. (2014) provide a similar result for phase retrieval. Sun & Luo (2014) give general conditions under which various algorithms work for exact matrix recovery. In contrast to these results, which require expensive SVD-like operations to initialize, our results allow random initialization. Our provided convergence rates apply to additional problems and SGD algorithms that are used in practice (but are not covered by previous analysis). However, our convergence rates are slower in their respective settings. This is likely unavoidable in our setting, as we show that our convergence rate is optimal in this more general setting .

A related class of algorithms that are similar to Alecton is stochastic power iteration (Arora et al., 2012). These algorithms reconsider (1) as an eigenvalue problem, and uses the familiar power iteration algorithm, adapted to a stochastic setting. Stochastic power iteration has been applied to a wide variety of problems (Arora et al., 2012; John Goes & Lerman, 2014). Oja (1985) show convergence of this algorithm, but provides no rate. Arora et al. (2013) analyze this problem, and state that "obtaining a theoretical understanding of the stochastic power method, or of how the step size should be set, has proved elusive." Our paper addresses this by providing a method for selecting the step size, although our analysis shows convergence for any sufficiently small step size.

Shamir (2014) provide exponential-rate local convergence results for a stochastic power iteration algorithm for PCA. As they note, it can be used in practice to improve the accuracy of an estimate returned by another, globally-convergent algorithm such as Alecton.

Also recently, Balsubramani et al. (2013) and Hardt & Price (2014) provide a global convergence rate for the stochastic power iteration algorithm. Our result only depends on the variance of the samples, while both their results require absolute bounds on the magnitude of the noise. This allows us to analyze a different class of noise models, which enables us to do matrix completion, phase retrieval, and subspace tracking in the same model.

2. Algorithmic Derivation

We focus on the low-rank factorized stochastic optimization problem (2). We can rewrite the objective as $\mathbf{E}\left[\tilde{f}(Y)\right]$, with sampled objective function

$$\tilde{f}(Y) = \mathbf{tr}\left(YY^TYY^T\right) - 2\mathbf{tr}\left(Y\tilde{A}Y^T\right) + \left\|\tilde{A}\right\|_F^2.$$

In the analysis that follows, we let $A = \mathbf{E}\left[\tilde{A}\right]$, and let its eigenvalues be $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ with corresponding orthonormal eigenvectors $u_1, u_2, \ldots, u_n$ (such a decomposition is guaranteed since A is symmetric). The standard stochastic gradient descent update rule for this problem is, for some step size α_k,

$$\begin{aligned} Y_{k+1} &= Y_k - \alpha_k \nabla \tilde{f}_k(Y) \\ &= Y_k - 4\alpha_k\left(Y_kY_k^TY_k - \tilde{A}_kY_k\right), \end{aligned}$$

where $\tilde{A}_k$ is the sample we use at timestep k.

The low-rank factorization introduces symmetry into the problem. If we let

$$\mathcal{O}_p = \left\{U \in \mathbb{R}^{p\times p} \mid U^TU = I_p\right\}$$

denote the set of orthogonal matrices in $\mathbb{R}^{p\times p}$, then $\tilde{f}(Y) = \tilde{f}(YU)$ for any $U \in \mathcal{O}_p$. Previous work has used manifold optimization techniques to solve such symmetric problems (Journée et al., 2010). Absil et al. (2008) state that stochastic gradient descent on a manifold has the general form

$$x_{k+1} = x_k - \alpha_k G_{x_k}^{-1} \nabla \tilde{f}_k(x_k),$$

where G_x is the matrix such that for all u and v,

$$u^TG_xv = \langle u, v\rangle_x,$$

where the right side of this equation denotes the *Riemannian metric* (do Carmo, 1992) of the manifold at x. For (2), the manifold in question is

$$\mathcal{M} = \mathbb{R}^{n\times p}/\mathcal{O}_p,$$

which is the quotient manifold of $\mathbb{R}^{n\times p}$ under the orthogonal group action. According to Absil et al. (2008), this manifold has induced Riemannian metric

$$\langle U, V\rangle_Y = \mathbf{tr}\left(UY^TYV^T\right). \tag{3}$$

For Alecton, we are free to pick any Riemannian metric and step size. Inspired by (3), we pick a new step size parameter η, and let $\alpha_k = \frac{1}{4}\eta$ and set

$$\langle U, V\rangle_Y = \mathbf{tr}\left(U(I + \eta Y^TY)V^T\right).$$

(We can think of this as an interpolation between the flat metric and the quotient metric.) With this, the SGD update rule becomes

$$\begin{aligned} Y_{k+1} &= Y_k - \eta\left(Y_kY_k^TY_k - \tilde{A}_kY_k\right)\left(I + \eta Y_k^TY_k\right)^{-1} \\ &= \left(Y_k\left(I + \eta Y_k^TY_k\right) - \eta\left(Y_kY_k^TY_k - \tilde{A}_kY_k\right)\right) \\ &\qquad \cdot \left(I + \eta Y_k^TY_k\right)^{-1} \\ &= \left(I + \eta\tilde{A}_k\right)Y_k\left(I + \eta Y_k^TY_k\right)^{-1}. \end{aligned}$$

For $p = 1$, choosing a Riemannian metric to use with SGD results in the same algorithm as choosing an SGD step size that depends on the iterate Y_k. The same update rule would result if we substituted

$$\alpha_k = \frac{1}{4}\eta\left(1 + \eta Y_k^TY_k\right)^{-1}$$

into the standard SGD update formula. We can think of this as the manifold results giving us intuition on how to set our step size.

The reason why selecting this particular step size/metric is useful in practice is that we can run the simpler update rule

$$\bar{Y}_{k+1} = \left(I + \eta\tilde{A}_k\right)\bar{Y}_k. \tag{4}$$

If $\bar{Y}_0 = Y_0$, the iteration will satisfy the property that the column space of Y_k will always be equal to the column space of $\bar{Y}_k$, (since $C(XY) = C(X)$ for any invertible matrix Y, where $C(X)$ denotes the column space of X). That is, if we just care about computing the column space of Y_k, we can do it using the much simpler update rule (4). Intuitively, we have transformed an optimization problem operating in the whole space $\mathbb{R}^n$ to one operating on the Grassmannian manifold; one benefit of Alecton is that we don't have to work on the actual Grassmannian, but get some of the same benefits from a rescaling of the Y_k space. In this specific case, the Alecton update rule is akin to stochastic power iteration, since it involves a repeated multiplication by the sample; this would not hold for optimization on other manifolds.

We can use (4) to compute the column space (or "angular component") of the solution, before then recovering the rest of the solution (the "radial component") using averaging. Doing this corresponds to Algorithm 1, Alecton. Notice that, unlike most iterative algorithms for matrix recovery, Alecton does not require any special initialization phase and can be initialized randomly.

Analysis Analyzing this algorithm is challenging, as the low-rank decomposition also introduces symmetrical families of fixed points. Not all these points are globally optimal: in fact, a fixed point will occur whenever

$$YY^T = \sum_{i\in C}\lambda_iu_iu_i^T$$

Algorithm 1 Alecton: Solve stochastic matrix problem

Require: $\eta \in \mathbb{R}$, $K \in \mathbb{N}$, $L \in \mathbb{N}$, and a sampling distribution $\mathcal{A}$

▷ ***Angular component (eigenvector) estimation phase***

Select Y_0 uniformly in $\mathbb{R}^{n\times m}$ s.t. $Y_0^T Y_0 = I$.

for $k = 0$ **to** $K - 1$ **do**

Select $\tilde{A}_k$ uniformly and independently at random from the sampling distribution $\mathcal{A}$.

$Y_{k+1} \leftarrow Y_k + \eta \tilde{A}_k Y_k$

end for

$\hat{Y} \leftarrow Y_K \left(Y_K^T Y_K\right)^{-\frac{1}{2}}$

▷ ***Radial component (eigenvalue) estimation phase***

$R_0 \leftarrow 0$

for $l = 0$ **to** $L - 1$ **do**

Select $\tilde{A}_l$ uniformly and independently at random from the sampling distribution $\mathcal{A}$.

$R_{l+1} \leftarrow R_l + \hat{Y}^T \tilde{A}_l \hat{Y}$

end for

$\bar{R} \leftarrow R_L / L$

return $\hat{Y}\bar{R}^{\frac{1}{2}}$

for any set C of size less than p.

One consequence of the non-optimal fixed points is that the standard proof of SGD's convergence, in which we choose a Lyapunov function and show that this function's expectation decreases with time, cannot work. If such a Lyapunov function were to exist, it would show that no matter where we initialize the iteration, convergence to a global optimum will still occur rapidly; this cannot be possible due to the presence of the non-optimal fixed points. Thus, a standard statement of global convergence, that convergence occurs uniformly regardless of initial condition, cannot hold.

We therefore use martingale-based methods to show convergence. Specifically, our attack involves defining a process x_k with respect to the natural filtration $\mathcal{F}_k$ of the iteration, such that x_k is a supermartingale, that is $\mathbf{E}\left[x_{k+1}|\mathcal{F}_k\right] \leq x_k$. We then use the *optional stopping theorem* (Fleming & Harrington, 1991) to bound both the probability and rate of convergence of x_k, from which we derive convergence of the original algorithm. We describe this analysis in the next section.

3. Convergence Analysis

First, we need a way to define convergence for the angular phase. For most problems, we want $C(Y_k)$ to be as close as possible to the span of $u_1, u_2, \ldots, u_p$. However, for some cases, this is not what we want. For example, consider the case where $p = 1$ but $\lambda_1 = \lambda_2$. In this case, the algorithm could not recover u_1, since it is indistinguishable from u_2. Instead, it is reasonable to expect $C(Y_k)$ to converge to the span of u_1 and u_2. To handle this case, we instead want to measure convergence to the subspace spanned by some number, $q \geq p$, of the most significant eigenvectors (in most cases, $q = p$). For a particular q, let U be the projection matrix onto the subspace spanned by $u_1, u_2, \ldots, u_q$, and define Δ, the *eigengap*, as $\Delta = \lambda_q - \lambda_{q+1}$. We now let $\epsilon > 0$ be an arbitrary tolerance, and define an angular success condition for Alecton.

Definition 1. When running the angular phase of Alecton, we define a quantity ρ_k to measure success, and say that *success has occurred* at timestep k if

$$\rho_k = \min_{z\in\mathbb{R}^p} \frac{\|UY_k z\|^2}{\|Y_k z\|^2} \geq 1 - \epsilon.$$

This condition requires that all members of the column space of Y_k are close to the desired subspace. We say that *success has occurred by time* t if success has occurred for some timestep $k < t$. Otherwise, we say the algorithm has *failed*, and we let F_t denote this failure event.

To prove convergence, we need to put some restrictions on the problem. Our theorem requires the following three conditions.

Condition 1 (Alecton Variance)**.** A sampling distribution $\mathcal{A}$ with expected value A satisfies the *Alecton Variance Condition* (AVC) with parameters (σ_a, σ_r) if for any $y \in \mathbb{R}^n$ and for any symmetric matrix $W \succeq 0$ that commutes with A, if $\tilde{A}$ is sampled from $\mathcal{A}$, the following bounds hold:

$$\mathbf{E}\left[y^T \tilde{A}^T W \tilde{A} y\right] \leq \sigma_a^2 \mathbf{tr}\left(W\right) \|y\|^2$$

and

$$\mathbf{E}\left[\left(y^T \tilde{A} y\right)^2\right] \leq \sigma_r^2 \|y\|^4.$$

In Section 4, we show several models that satisfy AVC.

Condition 2 (Alecton Rank)**.** An instance of Alecton satisfies the *Alecton Rank Condition* if either $p = 1$ (rank-1 recovery), or each sample $\tilde{A}$ from $\mathcal{A}$ is rank-1 (rank-1 sampling).

Most of the noise models we analyze have rank-1 samples, and so satisfy the rank condition.

Condition 3 (Alecton Step Size)**.** Define γ as

$$\gamma = \frac{2n\sigma_a^2 p^2 (p + \epsilon)}{\Delta\epsilon}\eta.$$

This represents a constant step size parameter that is independent of problem scaling. An instance of Alecton satisfies the *Alecton Step Size Condition* if $\gamma \leq 1$.

Note that the step size condition is only an upper bound on the step size. This means that, even if we do not know the problem parameters exactly, we can still choose a feasible

step size as long as we can bound them. (However, smaller step sizes imply slower convergence, so it is a good idea to choose η as large as possible.)

We will now define a useful function, then state our main theorem that bounds the probability of failure.

Definition 2. For some p, let $R \in \mathbb{R}^{p\times p}$ be a random matrix the entries of which are independent standard normal random variables. Define function Z_p as

$$Z_p(\gamma) = 2\left(1 - \mathbf{E}\left[\left|I + \gamma p^{-1}(R^T R)^{-1}\right|^{-1}\right]\right).$$

Theorem 1. *Assume that we run an instance of Alecton that satisfies the variance, rank, and step size conditions. Then for any $\chi > 0$, if we run for t timesteps where*

$$t \geq \frac{4n\sigma_a^2 p^2(p+\epsilon)}{\Delta^2\gamma\epsilon(\chi - Z_p(\gamma))}\log\left(\frac{np^2}{\gamma q\epsilon}\right), \tag{5}$$

then the probability that the angular phase has not succeeded is $P(F_t) \leq \chi$. Also, after running for L steps in the radial phase, for any constant ψ it holds that

$$P\left(\left\|\bar{R} - \hat{Y}^T A\hat{Y}\right\|_F^2 \geq \psi\right) \leq \frac{p^2\sigma_r^2}{L\psi}.$$

In particular, if $\sigma_a\Delta^{-1}$ does not vary with n, this theorem implies convergence of the angular phase with constant probability after $O(\epsilon^{-1}np^3\log n)$ iterations and in the same amount of time. Note that since we do not reuse samples in Alecton, our rates do not differentiate between sampling and computational complexity, unlike many other algorithms . We also do not consider numerical error or overflow: periodically re-normalizing the iterate may be necessary to prevent these in an implementation of Alecton. Note that if we initialized with the SVD instead of randomly, we could afford to pick a larger value of γ since we start nearer to the optimum; the algorithm will therefore converge quicker.

Since the upper bound expression uses Z_p, which is obscure, we plot it here (Figure 1). We also can make a more precise statement about the failure rate for $p = 1$.

Lemma 1. *For the case of rank-1 recovery,*

$$Z_1(\gamma) = \sqrt{2\pi\gamma}\exp\left(\frac{\gamma}{2}\right)\operatorname{erfc}\left(\sqrt{\frac{\gamma}{2}}\right) \leq \sqrt{2\pi\gamma}.$$

3.1. Martingale Technique

A proof for Theorem 1 and full formal definitions will appear in the appendix of this document, but since the method is nonstandard for non-convex optimization (although it has been used in Shamir (2011) to show convergence for convex problems), we will outline it here. First, we define a

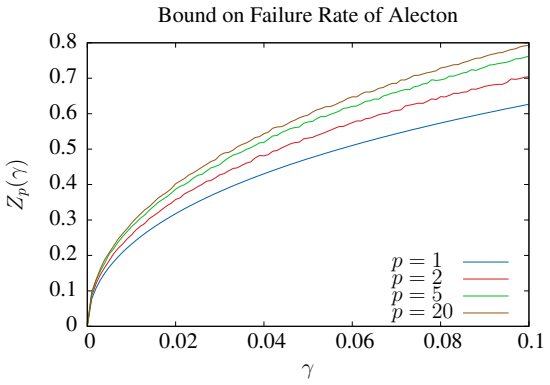


Figure 1. Value of Z_p computed as average of 10^5 samples.

failure event f_k at each timestep, that occurs if the iterate gets "too close" to the unstable fixed points. Next, we define a sequence τ_k, where

$$\tau_k = \frac{\left|Y_k^T U Y_k\right|}{\left|Y_k^T\left(\gamma n^{-1}p^{-2}qI + (1 - \gamma n^{-1}p^{-2}q)U\right)Y_k\right|}$$

(where $|X|$ denotes the determinant of X); the intuition here is that τ_k is close to 1 if and only if success occurs, and close to 0 when failure occurs. We show that, for some constant R, if neither success nor failure occurs at time k,

$$\mathbf{E}\left[\tau_{k+1}|\mathcal{F}_k\right] \geq \tau_k\left(1 + R\left(1 - \tau_k\right)\right); \tag{6}$$

here, $\mathcal{F}_k$ denotes the *filtration* at time k, which contains all the events that have occurred up to time k (Fleming & Harrington, 1991). If we let T denote the first time at which either success or failure occurs, then this implies that τ_k is a submartingale for $k < T$. We use the optional stopping Theorem (Fleming & Harrington, 1991) (here we state a discrete-time version).

Definition 3 (Stopping Time)**.** A random variable T is a stopping time with respect to a filtration $\mathcal{F}_k$ if $\{T \leq k\} \in \mathcal{F}_k$ for all k. That is, we can tell whether $T \leq k$ using only events that have occurred up to time k.

Theorem 2 (Optional Stopping Theorem)**.** *If x_k is a martingale (or submartingale) with respect to a filtration $\mathcal{F}_k$, and T is a stopping time with respect to the same filtration, then $x_{k\wedge T}$ is also a martingale (resp. submartingale) with respect to the same filtration, where $k \wedge T$ denotes the minimum of k and T. In particular, for bounded submartingales, this implies that $\mathbf{E}[x_0] \leq \mathbf{E}[x_T]$.*

Applying this to the submartingale τ_k and time T results in

$$\begin{aligned}\mathbf{E}[\tau_0] &\leq \mathbf{E}[\tau_T] \\ &= \mathbf{E}[\tau_T|F_T]\,P(f_T) + \mathbf{E}[\tau_T|\neg F_T]\,(1 - P(f_T)) \\ &\leq \delta P(f_T) + (1 - P(f_T)).\end{aligned}$$

This isolates the probability of the failure event occurring. Next, we return to (6); subtracting 1 from both sides and taking the logarithm results in

$$\begin{aligned}\mathbf{E}\left[\log\left(1-\tau_{k+1}\right)|\mathcal{F}_k\right] &\leq \log(1-\tau_k)+\log\left(1-R\tau_k\right)\\ &\leq \log(1-\tau_k)-R\delta.\end{aligned}$$

So, if we let $W_k = \log(1-\tau_k)+R\delta k$, then W_k is a supermartingale. We again apply the optional stopping theorem to produce

$$\mathbf{E}\left[W_0\right] \geq \mathbf{E}\left[W_T\right] = \mathbf{E}\left[\log(1-\tau_T)\right] + R\delta\mathbf{E}\left[T\right].$$

This isolates the expected value of the stopping time. Finally, we notice that success occurs before time t if $T \leq t$ and f_T does not occur. By the union bound, and Markov's inequality, this implies that

$$P_{\text{failure}} \leq P\left(f_T\right) + t^{-1}\mathbf{E}\left[T\right].$$

Substituting the isolated values for $P\left(f_T\right)$ and $\mathbf{E}\left[T\right]$ produces the result of Theorem 1.

The radial part of the theorem follows from an application of Chebychev's inequality to the average of L samples of $\hat{y}^T\tilde{A}\hat{y}$ — we do not devote any discussion to it since averages are already well understood.

4. Application Examples

4.1. Entrywise Sampling

One sampling distribution that arises in many applications (most importantly, matrix completion (Candès & Recht, 2009)) is *entrywise sampling*. This occurs when the samples are independently chosen from the entries of A. Specifically,

$$\tilde{A} = n^2 e_i e_i^T A e_j e_j^T,$$

where i and j are each independently drawn from $1,\ldots,n$. It is standard for these types of problems to introduce a *matrix coherence bound* (Jain et al., 2013).

Definition 4. A matrix $A \in \mathbb{R}^{n\times n}$ is incoherent with parameter μ if and only if for every unit eigenvector u_i of the matrix, and for all standard basis vectors e_j,

$$\left|e_j^T u_i\right| \leq \mu n^{-\frac{1}{2}}.$$

Under an incoherence assumption, we can provide a bound on the second moment of $\tilde{A}$, which is all that we need to apply Theorem 1 to this problem.

Lemma 2. *If A is incoherent with parameter μ, and $\tilde{A}$ is sampled uniformly from the entries of A, then the distribution of $\tilde{A}$ satisfies the Alecton variance condition with parameters $\sigma_a^2 = \mu^4\left\|A\right\|_F^2$ and $\sigma_r^2 = \mu^4\mathbf{tr}\left(A\right)^2$.*

For problems in which the matrix A is of constant rank, and its eigenvalues do not vary with n, neither $\left\|A\right\|_F$ nor $\mathbf{tr}\left(A\right)$ will vary with n. In this case, σ_a^2, σ_r^2, and Δ will be constants, and the $O(\epsilon^{-1}n\log n)$ bound on convergence time will hold.

4.2. Rectangular Entrywise Sampling

Entrywise sampling also commonly appear in rectangular matrix recovery problems. In these cases, we are trying to solve something like

$$\begin{array}{ll}\text{minimize} & \left\|M-X\right\|_F^2\\ \text{subject to} & X\in\mathbb{R}^{m\times n}, \mathbf{rank}\left(X\right)\leq p.\end{array}$$

To solve this problem using Alecton, we first convert it into a symmetric matrix problem by constructing the block matrix

$$A = \begin{bmatrix} 0 & M \\ M^T & 0\end{bmatrix};$$

it is known that recovering the dominant eigenvectors of A is equivalent to recovering the dominant singular vectors of M.

Entrywise sampling on M corresponds to choosing a random $i\in 1,\ldots,m$ and $j\in 1,\ldots,n$, and then sampling $\tilde{A}$ as

$$\tilde{A} = mnM_{ij}(e_i e_{m+j}^T + e_{m+j}e_i^T).$$

In the case where we can bound the entries of M (this is natural for recommender systems), we can prove the following.

Lemma 3. *If $M\in\mathbb{R}^{m\times n}$ satisfies the entry bound*

$$M_{ij}^2 \leq \xi m^{-1}n^{-1}\left\|M\right\|_F^2$$

for all i and j, then the rectangular entrywise sampling distribution on M satisfies the Alecton variance condition with parameters $\sigma_a^2 = \sigma_r^2 = 2\xi\left\|M\right\|_F^2$.

As above, for problems in which the magnitude of the entries of M is bounded and does not vary with problem size, our big-O convergence time bound will still hold.

4.3. Trace Sampling

Another common sampling distribution arises from the *matrix sensing* problem (Jain et al., 2013). In this problem, we are given the value of v^TAw for unit vectors v and w selected uniformly at random. (Candès et al. (2014) handle this problem for the more general complex case using Wirtinger flow.) Using this, we can construct an unbiased sample $\tilde{A} = n^2vv^TAww^T$; this lets us bound the variance.

Lemma 4. *If $n > 50$, and v and w are sampled uniformly from the unit sphere in $\mathbb{R}^n$, then for any positive semidefinite matrix A, if we let $\tilde{A} = n^2vv^TAww^T$, then the distribution of $\tilde{A}$ satisfies the Alecton variance condition with parameters $\sigma_a^2 = 16\left\|A\right\|_F^2$ and $\sigma_r^2 = 16\mathbf{tr}\left(A\right)^2$.*

If the eigenvalues of A do not vary with problem size, our big-O convergence time bound will be the same.

In some cases of the trace sampling problem, instead of being given samples of the form $u^T Av$, we know $u^T Au$. In this case, we need to use two independent samples $u_1^T Au_1$ and $u_2^T Au_2$, and let $u \propto u_1 + u_2$ and $v \propto u_1 - u_2$ be two unit vectors which we will use in the above sampling scheme. Notice that since u_1 and u_2 are independent and uniformly distributed, u and v will also be independent and uniformly distributed (by the spherical symmetry of the underlying distribution). Furthermore, we can compute

$$u^T Av = (u_1 + u_2)^T A(u_1 - u_2) = u_1^T Au_1 - u_2^T Au_2.$$

This allows us to use our above trace sampling scheme even with samples of the form $u^T Au$.

4.4. Subspace Sampling

We now analyze the following more complicated distribution, which arises in subspace tracking (Balzano et al., 2010). Our matrix A is a rank-r projection matrix, and each sample consists of some randomly-selected entries from a randomly-selected vector in its column space. Specifically, we are given Qv and Rv, where v is selected uniformly at random from $C(A)$, and Q and R are independent random diagonal projection matrices with expected value $mn^{-1}I$. With this, we can construct the unbiased sample

$$\tilde{A} = rn^2 m^{-2} Qvv^T R.$$

As in the entrywise case, we need to introduce a coherence constraint to bound the second moment.

Definition 5. A subspace of $\mathbb{R}^n$ of dimension q with associated projection matrix U is incoherent with parameter μ if for all standard basis vectors e_i, $\|Ue_i\|^2 \leq \mu r n^{-1}$.

Using this, we can prove the following facts about the second moment of this distribution.

Lemma 5. *The subspace sampling distribution, when sampled from a subspace that is incoherent with parameter μ, satisfies the Alecton variance condition with parameters $\sigma_a^2 = \sigma_r^2 = r^2(1 + \mu r m^{-1})^2$.*

Sometimes we are given just one random diagonal projection matrix S, and the product Sv. We can use this to construct a sample of the above form by randomly splitting the given entries among Q and R in such a way that $Q = QS$ and $R = RS$, and Q and R are independent. We can then construct an unbiased sample $\tilde{A} = rn^2 m^{-2} QSvv^T SR$, which uses only the entries of v that we are given.

4.5. Noisy Sampling

Since our analysis depends only on a variance bound, it extends naturally to the case in which the values of our samples themselves are noisy. Using the additive property of the variance for independent random variables, we can show that additive noise only increases the variance of the sampling distribution by a constant amount proportional to the variance of the noise. Similarly, using the multiplicative property of the variance for independent random variables, multiplicative noise only multiplies the variance of the sampling distribution by a constant factor proportional to the variance of the noise. In either case, we can show that the noisy sampling distribution satisfies AVC. Numerical imprecision can also be modeled in the same way.

Algorithm 2 Alecton One-at-a-time

Require: A sampling distribution $\mathcal{A}_1$
  for $i = 1$ **to** p **do**
    ▷ Run rank-1 Alecton to produce output y_i.
    $y_i \leftarrow \text{Alecton}_{p=1}(\mathcal{A}_i)$
    Generate sampling distribution $\mathcal{A}_{i+1}$ such that, if $\tilde{A}'$ is sampled from $\mathcal{A}_{i+1}$ and $\tilde{A}$ is sampled from $\mathcal{A}_i$, $\mathbf{E}\left[\tilde{A}'\right] = \mathbf{E}\left[\tilde{A}\right] - y_i y_i^T$.
  end for
  return $\sum_{i=1}^{p} y_i y_i^T$

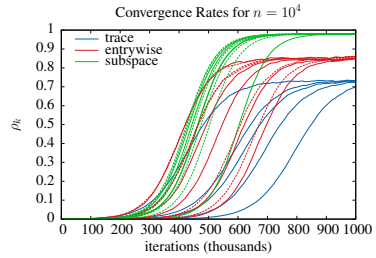
4.6. Extension to Higher Ranks

It is possible to use multiple iterations of the rank-1 version of Alecton to recover additional eigenvalue/eigenvector pairs of the data matrix A one-at-a-time. This is a standard technique for using power iteration algorithms to recover multiple eigenvalues. Sometimes, this may be preferable to using a single higher-rank invocation of Alecton (for example, we may not know a priori how many eigenvectors we want). We outline this technique as Algorithm 2. If the eigenvalues of A are independent of n and p, it will converge in $O(\epsilon^{-1} pn \log n)$ total SGD update steps.

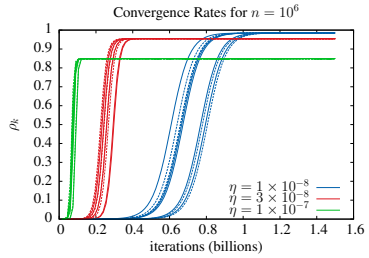
5. Experiments

We experimentally verify our main claim, that Alecton does converge quickly for practical datasets. No data was collected for the radial phase of Alecton, since the performance of averaging is already well understood.

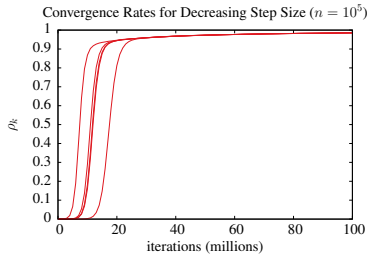
The first experiments were run on symmetric synthetic data matrices $A \in \mathbb{R}^{n \times n}$ each with ten random eigenvalues $\lambda_i > 0$. Figure 2(a) illustrates the convergence of Alecton with $p = q = 1$ using three sampling distributions on datasets with $n = 10^4$. We ran Alecton starting from five random initial values; the different plotted trajectories illustrate how convergence time can depend on the initial value. Note that, due to the underlying symmetry of the quadratic substitution, the multiple runs of the algorithm do not converge to the same value of Y but rather $X = YY^T$.



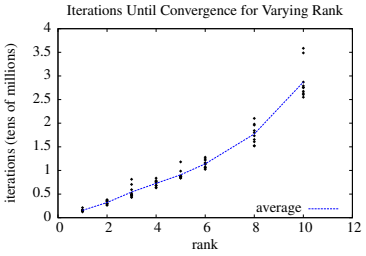
(a) Angular convergence of three distributions on a synthetic dataset with $\eta = 10^{-5}$.



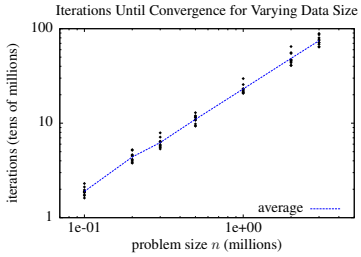
(b) Angular convergence of entrywise sampling on a large synthetic dataset for different step sizes.



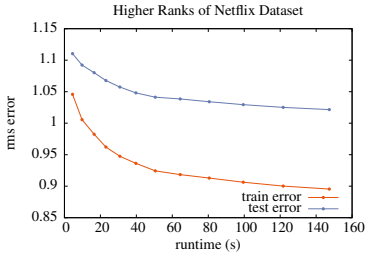
(c) Angular convergence of entrywise sampling on a synthetic dataset for a decreasing step size scheme.



(d) Iterations until convergence of angular phase for recovering higher-rank estimates on large synthetic dataset.



(e) Iterations until convergence of angular phase for larger synthetic dataset sizes.



(f) RMS errors for higher-rank Alecton one-at-a-time algorithm.

Figure 2. Experiments ran on a single twelve-core machine (Intel Xeon E5-2697, 2.70GHz) with 256 GB of shared memory.

Figure 2(b) illustrates the performance of Alecton on a larger dataset with $n = 10^6$ as the step size parameter η is varied: a smaller value of η yields slower, but more accurate convergence. Also, the smaller the value of η, the more the initial value seems to affect convergence time.

Figure 2(c) shows convergence of a modified version of Alecton in which the step size η is decreased over time (proportional to $1/k$): we converge to the global optimum, rather than to a noise floor as in the constant-η case. Figure 2(d) shows the angular convergence time of Alecton on a dataset with $n = 10^4$ as the rank of the model changes: the convergence time increases as the rank increases. Figure 2(e) gives the angular convergence time of Alecton as the dataset size changes. It illustrates the near-linear relationship between dataset size and convergence time.

Figure 2(f) demonstrates convergence results on real data from the Netflix Prize problem (Funk, 2006). This problem involves recovering a matrix with 480,189 columns and 17,770 rows from a training dataset containing 110,198,805 revealed entries. We used the rectangular entrywise distribution described above, and ran Alecton One-at-a-time to recover the twelve most significant singular vectors of the matrix, using 10^7 iterations for each run of Alecton. Each point in Figure 2(f) represents the absolute runtime and RMS errors after the recovery of some number of eigenvectors. This plot illustrates that the runtime of this algorithm does not increase disastrously as the number of recovered eigenvectors expands.

5.1. Future Work

The Hogwild! algorithm (Niu et al., 2011) is a parallel, lock-free version of SGD that performs similarly to sequential SGD on convex problems. It is an open question whether a Hogwild! version of Alecton converges with a good rate, but we are optimistic that it will.

6. Conclusion

This paper exhibited Alecton, a stochastic gradient descent algorithm applied to a non-convex low-rank factorized problem; it is similar to the algorithms used in practice to solve a wide variety of problems. We prove that Alecton converges globally, and provide a rate of convergence. We do not require any special initialization step but rather initialize randomly. Furthermore, our result depends only on the variance of the samples, and therefore holds under broad sampling conditions that include both matrix completion and matrix sensing, and is also able to take noisy samples into account. We show these results using a martingale-based technique that is novel in the space of non-convex optimization, and we are optimistic that this technique can be applied to other problems in the future.

Acknowledgments

Thanks to Ben Recht, Mahdi Soltanolkotabi, Joel Tropp, Kelvin Gu, and Madeleine Udell for helpful conversations. Thanks also to Ben Recht and Laura Waller for datasets.

The authors acknowledge the support of: DARPA Contract-Air Force, Xgraphs; Language and Algorithms for Heterogeneous Graph Streams, FA8750-12-2-0335; NSF Grant, BIGDATA: Mid-Scale: DA: Collaborative Research: Genomes Galore - Core Techniques, Libraries, and Domain Specific Languages for High-Throughput DNA Sequencing, IIS-1247701; NSF Grant, SHF: Large: Domain Specific Language Infrastructure for Biological Simulation Software, CCF-1111943; Dept. of Energy- Pacific Northwest National Lab (PNNL)- Integrated Compiler and Runtime Autotuning Infrastructure for Power, Energy and Resilience-Subcontract 108845; NSF Grant EAGER-XPS:DSD:Synthesizing Domain Specific Systems-CCF-1337375; Stanford PPL affiliates program, Pervasive Parallelism Lab: Oracle, NVIDIA, Huawei, SAP Labs; DARPA XDATA Program under No. FA8750-12-2-0335 and DEFT Program under No. FA8750-13-2-0039; DARPAs MEMEX program and SIMPLEX program; the National Science Foundation (NSF) CAREER Award under No. IIS-1353606; the Office of Naval Research (ONR) under awards No. N000141210041 and No. N000141310129; the National Institutes of Health Grant U54EB020405 awarded by the National Institute of Biomedical Imaging and Bioengineering (NIBIB) through funds provided by the trans-NIH Big Data to Knowledge (BD2K, http://www.bd2k.nih.gov) initiative; the Sloan Research Fellowship; the Moore Foundation; American Family Insurance; Google; and Toshiba.

"The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, AFRL, NSF, ONR, NIH, or the U.S. Government."

References

Absil, P.-A., Mahony, R., and Sepulchre, R. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008. ISBN 978-0-691-13298-3.

Agarwal, Alekh, Chapelle, Olivier, Dudík, Miroslav, and Langford, John. A reliable effective terascale linear learning system. *CoRR*, abs/1110.4198, 2011.

Arora, R., Cotter, A., Livescu, K., and Srebro, N. Stochastic optimization for PCA and PLS. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pp. 861–868, Oct 2012.

Arora, Raman, Cotter, Andy, and Srebro, Nati. Stochastic optimization of PCA with capped MSG. In Burges, C.j.c., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K.q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 1815–1823. 2013.

Balsubramani, Akshay, Dasgupta, Sanjoy, and Freund, Yoav. The fast convergence of incremental PCA. In *NIPS*, pp. 3174–3182, 2013.

Balzano, Laura, Nowak, Robert, and Recht, Benjamin. Online identification and tracking of subspaces from highly incomplete information. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pp. 704–711. IEEE, 2010.

Bottou, Lon. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. 2010.

Bottou, Lon and Bousquet, Olivier. The tradeoffs of large scale learning. In *IN: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 20*, pp. 161–168, 2008.

Burer, Samuel and Monteiro, Renato DC. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

Burer, Samuel and Monteiro, Renato DC. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005.

Candès, Emmanuel, Li, Xiaodong, and Soltanolkotabi, Mahdi. Phase retrieval via wirtinger flow: Theory and algorithms. *arXiv preprint arXiv:1407.1065*, 2014.

Candès, Emmanuel J. and Recht, Benjamin. Exact matrix completion via convex optimization. *FoCM*, 9(6):717–772, 2009. ISSN 1615-3375.

Candès, EmmanuelJ. and Li, Xiaodong. Solving quadratic equations via phaselift when there are about as many equations as unknowns. *FoCM*, 14(5):1017–1026, 2014.

Chen, Caihua, He, Bingsheng, and Yuan, Xiaoming. Matrix completion via an alternating direction method. *IMAJNA*, 2011.

do Carmo, M.P. *Riemannian Geometry*. Mathematics (Birkhäuser) theory. Birkhäuser Boston, 1992. ISBN 9780817634902.

Duchi, John, Hazan, Elad, and Singer, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July 2011.

Fleming, Thomas R and Harrington, David P. Counting processes and survival analysis. volume 169, pp. 56–57. John Wiley & Sons, 1991.

Funk, Simon. Netflix Update: Try this at Home. 2006.

Gupta, Pankaj, Goel, Ashish, Lin, Jimmy, Sharma, Aneesh, Wang, Dong, and Zadeh, Reza. WTF: The who to follow service at twitter. WWW '13, pp. 505–514, 2013.

Hardt, Moritz. Understanding alternating minimization for matrix completion. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pp. 651–660. IEEE, 2014.

Hardt, Moritz and Price, Eric. The noisy power method: A meta algorithm with applications. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.d., and Weinberger, K.q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2861–2869. Curran Associates, Inc., 2014.

Horstmeyer, R., Chen, R. Y., Ou, X., Ames, B., Tropp, J. A., and Yang, C. Solving ptychography with a convex relaxation. *ArXiv e-prints*, dec 2014.

Hu, Chonghai, Kwok, James T., and Pan, Weike. Accelerated gradient methods for stochastic optimization and online learning. In *Advances in Neural Information Processing Systems 22*, pp. 781–789, 2009.

Jain, Prateek and Netrapalli, Praneeth. Fast exact matrix completion with finite samples. *arXiv preprint arXiv:1411.1087*, 2014.

Jain, Prateek, Netrapalli, Praneeth, and Sanghavi, Sujay. Low-rank matrix completion using alternating minimization. In *Proceedings of the Forty-fifth Annual ACM STOC*, pp. 665–674. ACM, 2013.

John Goes, Teng Zhang, Raman Arora and Lerman, Gilad. Robust stochastic principal component analysis. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, pp. 266–274, 2014.

Journée, M., Bach, F., Absil, P.-A., and Sepulchre, R. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM J. on Optimization*, 20(5):2327–2351, May 2010.

Keshavan, R.H., Montanari, A., and Oh, Sewoong. Matrix completion from a few entries. *Information Theory, IEEE Transactions on*, 56(6):2980–2998, June 2010.

Mishra, Bamdev, Meyer, Gilles, Bach, Francis, and Sepulchre, Rodolphe. Low-rank optimization with trace norm penalty. *SIAM Journal on Optimization*, 23(4):2124–2149, 2013.

Niu, Feng, Recht, Benjamin, Ré, Christopher, and Wright, Stephen J. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *In NIPS*, 2011.

Oja, Erkki. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of Mathematical Analysis and Applications*, 106, 1985.

Oscar Boykin, Sam Ritchie, Ian O'Connell Jimmy Lin. Summingbird: A framework for integrating batch and online mapreduce computations. In *Proceedings of the VLDB Endowment*, volume 7, pp. 1441–1451, 2013-2014.

Recht, Benjamin and Ré, Christopher. Parallel stochastic gradient algorithms for large-scale matrix completion. *Mathematical Programming Computation*, 5(2): 201–226, 2013. ISSN 1867-2949.

Shamir, Ohad. Making gradient descent optimal for strongly convex stochastic optimization. *CoRR*, abs/1109.5647, 2011.

Shamir, Ohad. A stochastic PCA algorithm with an exponential convergence rate. *CoRR*, abs/1409.2848, 2014.

Sun, Ruoyu and Luo, Zhi-Quan. Guaranteed matrix completion via non-convex factorization. *arXiv preprint arXiv:1411.8003*, 2014.

Teflioudi, Christina, Makari, Faraz, and Gemulla, Rainer. Distributed matrix completion. *2013 IEEE 13th ICDM*, 0:655–664, 2012. ISSN 1550-4786.

Zou, Hui, Hastie, Trevor, and Tibshirani, Robert. Sparse principal component analysis. *J. Comp. Graph. Stat.*, 15:2006, 2004.