Extension of a saddle point mirror descent algorithm
with application to robust PageRank

Andrey Tremba and Alexander Nazin

Abstract—The paper is devoted to designing an efficient recursive algorithm for solving the robust PageRank problem recently proposed by Juditsky and Polyak [7]. To this end, we reformulate the problem to a specific convex-concave saddle point problem \( \min_{x \in X} \max_{y \in Y} q(x, y) \) with simple convex sets \( X \subset \mathbb{R}^N \) and \( Y \subset \mathbb{R}^N \), i.e., standard simplex and Euclidean unit ball, respectively. Aiming this goal we develop an extension of saddle point mirror descent algorithm where additional parameter sequence is introduced, thus providing more degree of freedom and the refined error bounds. Detailed complexity results of this method applied to the robust PageRank problem are given and discussed. Numerical example illustrates the theoretical results proved.

I. INTRODUCTION

The ultimate goal is to solve robust page ranking problem on unit simplex \( S_N = \{ x : \sum_{i=1}^{N} x_i = 1, \forall i, x_i \geq 0 \} \) that is

\[
x^* = \arg\min_{x \in S_N} (\|Ax - x\|_2 + \varepsilon \|x\|_2),
\]

(1)

with a column-stochastic matrix \( A \) and a relevant parameter \( \varepsilon > 0 \). This problem was introduced in [4], motivated that its solution is less-sensitive to disturbances than the original PageRank solution vector. We use game approach proposed in [7] for the original PageRank problem. This approach is very suitable for robust PageRank problem as well due to its nonlinearity. There are good traits of (1) that solution is unique, and it may be reformulated as a minmax (game) problem \( \min_{x \in S_N} \max_{y \in B_N} q(x, y) \) with unit Euclidean ball \( B_N = \{ y \in \mathbb{R}^N : \|y\|_2 \leq 1 \} \) and game function

\[
q(x, y) = y^T (Ax - x) + \varepsilon \|x\|_2.
\]

Main issue of solving (1) is that cost of calculation of a (sub)gradient of objective function for huge-dimensional vector \( x \) is high, and normally involve laborious matrix-vector multiplications. We exploit and improve mirror descent algorithms, both deterministic and stochastic ones, which only require subgradients (or stochastic subgradients).

In the latter method the matrix-vector multiplications are avoided at all.

Notice, that \( \varepsilon = 0 \) reduces the function in (2) to that of [7], and the algorithms therein may be treated as the consequences of those presented here in the paper. In other words, we extend the approach of [7] to the robust PageRank problem. Let’s remind that including parameter \( \varepsilon > 0 \) in (1) and, therefore, (2), is not simply treated as regularization, or penalty, or whatever. In fact, it is a novel approach to solving ranking problem, which has been called robust PageRank [4].

The contribution is two-fold: first, we derive an extension of the stochastic mirror descent algorithm for saddle point problem (as well as deterministic mirror descent algorithm), [9], with extra parameters, and show its non-asymptotic convergence rate. Second, for the specific robust PageRank problem we compare these algorithms, with separate complexity in number of steps and matrix-vector multiplication complexity, thus help choosing appropriate algorithm with given problem size. The results numerically tested on robust PageRank calculation.

The paper organized as follows: in next section preliminary notions and problem formulation are given, then goes a section with deterministic and stochastic (randomized) algorithms themselves. In IV-th section these algorithms are applied to the stated robust PageRank problem, followed by complexity analysis and example. Sections with proofs and future works suggestions conclude the paper.

II. PRELIMINARIES

Let us introduce convex compacts \( X \subset \mathbb{R}^N, Y \subset \mathbb{R}^M \) equipped with norms \( \|x\|_x, \|y\|_y \) and function \( q(x, y) \). We assume \( q(x, y) \) is continuous, \( x \in X, y \in Y \), and \( q(x, y) \geq 0 \). Moreover, we assume \( q(x, y) \) is convex in \( x \) and \( y \) and \( C \)-Lipschitz continuous in \( x \) and \( y \), i.e.,

\[
q(x, y) \leq q(x', y) \leq q(x, y') \leq q(x', y') \leq q(x, y) + C \|x - x'\|_x + C \|y - y'\|_y.
\]

A. Generic convex-concave saddle point problem

Let function \( q : X \times Y \to \mathbb{R} \) be continuous, defined on convex compact sets \( X, Y \), and let it be convex in \( x \in X \) and concave in \( y \in Y \). Then it has a saddle point \( (x^*, y^*) \in X \times Y \), i.e., for all \( x \in X \) and \( y \in Y \), the following inequalities hold:

\[
q(x^*, y) \leq q(x^*, y^*) \leq q(x, y^*).
\]

(3)

In other words, the saddle point is a solution to minimax problem \( q^* = q(x^*, y^*) = \min_{x \in X} \max_{y \in Y} q(x, y) = \min_{y \in Y} \max_{x \in X} q(x, y) \).
\[
\max_{y \in Y} \min_{x \in X} q(x, y). \quad \text{Quality of a candidate point } (\bar{x}, \bar{y}) \in X \times Y \text{ can be naturally expressed in terms of the error (or gap):
}\]
\[
\Delta(\bar{x}, \bar{y}) \doteq (\max_{y \in Y} q(\bar{x}, y) - q^*) + (q^* - \min_{x \in X} q(x, \bar{y})) = \max_{y \in Y} q(\bar{x}, y) - \min_{x \in X} q(x, \bar{y}).
\]
Note that inequality \(q(\bar{x}, y) - q(x, \bar{y}) \leq c, \forall (x, y) \in X \times Y, \) implies the bound \(\Delta(\bar{x}, \bar{y}) \leq c, \) which leads to the inequality \(|q(\bar{x}, y) - q^*| \leq c.\)

\section{B. Proxy functions}

Here we exploit the properties of convex compacts \(X\) and \(Y.\) Let’s consider strongly convex modulus \(\alpha_x\) and \(\alpha_y\) proxy functions \(V : X \to \mathbb{R}_+\) and \(T : Y \to \mathbb{R}_+\) (with respect to the norms \(\| \cdot \|_x\) and \(\| \cdot \|_y\)). For example, let \(X\) be a unit simplex \(S_N,\) and \(Y\) be a unit ball \(B_M,\) equipped with \(\ell_2\)-norm and Euclidean one, respectively. Then proxy functions may be chosen as
\[
\hat{V}(x) = \ln N + \sum_{i=1}^{N} x^{(i)} \ln x^{(i)}, \quad x \in S_N, \\
\hat{T}(y) = \frac{1}{2} \| y \|^2, \quad y \in B_M,
\]
with the modulus’ \(\alpha_x = 1\) and \(\alpha_y = 1.\) We remind that the choice of these proxy-functions made in purpose to exploit simple (and explicit) calculation of Legendre-Fenchel transform according to \(X\) and \(Y\) (following functions \(W\) and \(U\)), and acquiring its maximizers:
\[
W_\beta(\zeta) = \max_{x \in X} (-\zeta^T x - \beta V(x)), \quad \zeta \in \mathbb{R}^N, \\
U_\delta(\eta) = \max_{y \in Y} (-\eta^T y - \delta T(y)), \quad \eta \in \mathbb{R}^M,
\]
with parameters \(\beta, \delta > 0\), and properties of their gradients, including Lipschitz ones [13]:
\[
-\nabla W_\beta(\zeta) = \arg \min_{x \in X} (\zeta^T x + \beta V(x)) \in X, \quad \frac{d}{d\zeta} W_\beta(\zeta) = \nabla \zeta, \\
-\nabla U_\delta(\eta) = \arg \min_{y \in Y} (\eta^T y + \delta T(y)), \quad \frac{d}{d\eta} U_\delta(\eta) = \nabla \eta, \\
\| \nabla W_\beta(\zeta) \|_x \leq \frac{1}{\beta}, \| \zeta \|_x, \| \nabla U_\delta(\eta) \|_y \leq \frac{1}{\delta}, \| \eta \|_y.
\]

Used dual norms are defined as usual through the primal ones, i.e., \(\| \zeta \|_x = \max_{\| x \|_x \leq 1} x^T \zeta.\)

For \(\hat{V}\) on \(S_N\) and \(\hat{T}\) on \(B_M,\) we have componentwise
\[
\frac{\partial \hat{W}_\beta(\zeta)}{\partial \zeta^{(i)}} = -e^{-\zeta^{(i)} / \beta} \left( \sum_{k=1}^{N} e^{-\zeta^{(k)} / \beta} \right)^{-1}, \quad i = 1, \ldots, N, \\
\frac{\partial \hat{U}_\delta(\eta)}{\partial \eta^{(j)}} = \begin{cases} \frac{1}{\eta^{(j)}} & \text{if } \| \eta \| \leq \delta, \\ 0 & \text{otherwise}. \end{cases}
\]

It is a matter of choice to consider these gradient maps as non-linear projections onto \(S_N\) and \(B_N (X \times Y \text{ in general case}) [1].\) Finally, let’s define extremal values
\[
V_* = \min_{x \in X} V(x), \quad V^* = \max_{x \in X} V(x), \\
T_* = \min_{y \in Y} T(y), \quad T^* = \max_{y \in Y} T(y).
\]

For the particular choice \(\hat{V}, \hat{T}\) for \(X = S_N, Y = B_M\) we have \(\hat{V}_* = \hat{T}_* = 0, \quad \hat{V}_* = \ln N, \quad \hat{T}_* = 1/2.\)

\section{C. Subgradients}

Let function \(q(x, y)\) be convex on \(x \in X\) (for a given \(y).\) Its (partial) subdifferential at point \((x_1, y)\) is a set of all vectors \(q_x \in \mathbb{R}^N\) satisfying
\[
q(x_1, y) \geq q(x_1, y) + (x_2 - x_1)^T q_x, \quad \forall x_2 \in X, \forall y \in Y.
\]
This set is nonempty for convex functions and contain exactly one element coinciding with partial derivative for the smooth convex functions. Sometimes it is denoted as \(\partial^\delta q(x, y).\) We denote as partial subgradient \(q_x(x, y)\) one somehow deterministically distinguished element of the subdifferential set at the point \((x, y).\) The “partial” prefix thereafter is omitted for shortcut.

Similarly we denote subgradient \(g_y \in \mathbb{R}^M\) at point \((x, y_1)\) for concave function, e.g. for \(q(x, y_1)\) being concave on \(y,\) the following property holds
\[
q(x, y_2) \leq q(x, y_1) + (y_2 - y_1)^T g_y(x, y_1), \quad x \in X, \forall y_2 \in Y.
\]

\section{III. Mirror descent algorithm (MDA)}

The mirror descent method goes back to [10]. There is a connection of MDA with nonlinear projected subgradient method [1]. Deterministic algorithm with averaging was presented in early 2000-th as primal-dual subgradient method [11], including variation for saddle point problems. There were two parameter sequences introduced. The advanced stochastic MDA versions were developed in [3]. For game problem the stochastic MDA is also known as saddle point mirror stochastic approximation algorithm [9].

In this section we enhance the saddle point MDA with one more parameter sequence \(\delta_k\) and show that the algorithm’s core is the same for both deterministic and stochastic algorithms producing similar bounds. General properties of the algorithms are followed by specific choice of parameters.

\subsection{A. Deterministic MDA: Algorithm 1}

Algorithm 1 (which is much similar to that of described in [7]) has two different parameter sequences \(\beta_k\) and \(\delta_k\) on the second (projection) step instead of the same sequence. Including weighting parameter sequence \(\gamma_k\) there are as many as three control sequences in total.

1. Specify positive sequence \(\gamma_k, k \geq 1\) and two non-decreasing positive sequences \(\beta_k, \delta_k, k \geq 0.\) Fix initial value as zero vector for joint variables \(\zeta_0, \eta_0 \in \mathbb{R}^{M+N}\) and calculate initial values \(x_0 = -\nabla W_{\beta_0}(\zeta_0), y_0 = -\nabla U_{\delta_0}(\eta_0) \in X \times Y\) for the primal vector variables.

2. For each \(k = 1, \ldots, n,\) given primal pair \((x_{k-1}, y_{k-1})\) and dual one \((\zeta_{k-1}, \eta_{k-1}),\) update variables
\[
\begin{bmatrix}
\zeta_k \\
\eta_k
\end{bmatrix} = \begin{bmatrix} \zeta_{k-1} \\
\eta_{k-1}
\end{bmatrix} + \gamma_k \begin{bmatrix} q_x(x_{k-1}, y_{k-1}) \\
-q_y(x_{k-1}, y_{k-1})
\end{bmatrix}, \\
x_k = -\nabla W_{\beta_k}(\zeta_k), \quad y_k = -\nabla U_{\delta_k}(\eta_k).
\]

using subgradients \(q_x(\cdot, \cdot), q_y(\cdot, \cdot).\)

3. These values should not be chosen arbitrary as supposed in [3].
3) Output the averaged vectors

\[ x_n = \frac{1}{\sum_{k=1}^{n} \gamma_k} \sum_{k=1}^{n} \gamma_k x_{k-1}, \]
\[ y_n = \frac{1}{\sum_{k=1}^{n} \gamma_k} \sum_{k=1}^{n} \gamma_k y_{k-1}. \]  

Theorem 1 (deterministic MDA): Given averaged estimates (6) generated by Algorithm 1 at \( n \)-th step, the error is bounded as follows:

\[ \Delta(x_n, y_n) \leq \frac{1}{\sum_{k=1}^{n} \gamma_k} \left( R_n + \sum_{k=1}^{n} \gamma_k \left( \frac{L^2_{x,x}}{\alpha_x \beta_k} + \frac{L^2_{y,y}}{\alpha_y \delta_k} \right) \right), \]

where \( L_{x,x} = \max_{(x,y) \in X \times Y} \|q_x(x,y)\|_{\star,x}, \)
\( L_{x,y} = \max_{(x,y) \in X \times Y} \|q_y(x,y)\|_{\star,y}, \)
\( R_n = \beta_0 V^* - \beta_0 V_n + \delta_0 T^* - \delta_0 T_n. \)

All proofs are given in section V.

B. Stochastic (or randomized) MDA: Algorithm 2

Algorithm 2 is formally the same as deterministic one, but on the second step in (5) instead of exact subgradients \( q_x(x_k, y_k) \) and \( q_y(x_k, y_k) \) the stochastic subgradients \( \phi_k, \psi_k \) are used with the following properties:

\[ \mathbb{E}\{\phi_k(x_t, y_t) \leq k \} = q_x(x_{k-1}, y_{k-1}), \]
\[ \mathbb{E}\{\psi_k(x_t, y_t) \leq k \} = q_y(x_{k-1}, y_{k-1}), \]
\[ \mathbb{E}\{\|\phi_k\|_{\star,x}^2 \leq \hat{L}^2_{x,x}, \}
\[ \mathbb{E}\{\|\psi_k\|_{\star,y}^2 \leq \hat{L}^2_{y,y}. \}

Theorem 2 (stochastic MDA): Under assumptions of Algorithm 2 and properties (7), (8), the bound holds

\[ \mathbb{E}\Delta(x_n, y_n) \leq \frac{1}{\sum_{k=1}^{n} \gamma_k} \left( R_n + \sum_{k=1}^{n} \gamma_k \left( \frac{L^2_{x,x}}{\alpha_x \beta_k} + \frac{L^2_{y,y}}{\alpha_y \delta_k} \right) \right), \]

\( R_n = \beta_0 V^* - \beta_0 V_n + \delta_0 T^* - \delta_0 T_n. \)

In application to PageRank (Section IV), we’ll introduce auxiliary random variables and define stochastic gradients based on these variables. To express artificial nature of randomization such MDAs may be called as randomized MDAs.

C. Parameter’s choice

As we see, three parameter sequences \( \gamma_k, \beta_k, \delta_k \) are to be prefixed in both algorithms. While \( \gamma_k \) have sense of weighting factors for data in given points, and they actually cannot be chosen different in advance\(^6\), it is being simply put as \( \gamma_k = 1, k = 1, \ldots, n, \) with recurrent calculation of \( x_n, y_n: \)

\[ x_n = \left( 1 - \frac{1}{\gamma_n} \right) x_{n-1} + \frac{1}{\gamma_n} x_{n-1}, \]
\[ y_n = \left( 1 - \frac{1}{\gamma_n} \right) y_{n-1} + \frac{1}{\gamma_n} y_{n-1}, \]
\( x_0 = 0, \)
\( y_0 = 0. \)

So we are left only with \( \beta_k, \delta_k \) for minimization of the upper bounds. It may be shown (cf. [3]) that good choice is

\[ \beta_k = \beta_0 \sqrt{k+1}, \]
\[ \delta_k = \delta_0 \sqrt{k+1}, \]
\[ \beta_0 = \frac{L_{x,x}}{\sqrt{\alpha_x}}, \]
\[ \delta_0 = \frac{L_{y,y}}{\sqrt{\alpha_y}}. \]

\(^6\)An online choice still possible and may be studied as well, cf. [11].

which leads to the error bound as small as

\[ \Delta(x_n, y_n) \leq \frac{\sqrt{n+1}}{n} \left( L_{x,x} \sqrt{\frac{V^*}{\alpha_x}} + L_{x,y} \sqrt{\frac{T^*}{\alpha_y}} \right). \]

This bound has the same order on number of iterations \( n \) but is better in constant than the bound for two-sequences case with \( \delta_k = \beta_k, \) cf. [7]. For stochastic MDA, from Theorem 2 the same bound with constants \( L_{x,x} \) and \( L_{x,y} \) follows instead of \( L_{x,x} \) and \( L_{x,y}, \) respectively. Another advantage of having three control sequences is that the adaptive step-size may be chosen for each of \( \beta_k \) and \( \delta_k \) separately, though it would need proof modification like in [8].

IV. APPLYING THE ROBUST PAGE-RANK

In this section we apply both algorithms for the specific problem (1), for minimizing function

\[ f(x) = \|Ax - x\|_2 + \varepsilon \|x\|_2 \]

on unit simplex \( S_N \) with column-stochastic\(^8\) matrix \( A. \) The solution \( x^* \) may be thought as \( x \)-part of saddle point problem for function \( q(x, y) \) (2) and sets \( X = S_N, Y = B_N. \)

In application of MDAs for this problem it is naturally to use proxy-functions \( \hat{V}, \tilde{T}, \ell_1 \)-norm as \( \| \cdot \|_1 \) norm and Euclidean \( \| \cdot \|_2 \) norm as \( \| \cdot \|_2 \), with conjugate \( \ell_{\infty} \)-norm as \( \| \cdot \|_{\infty} \) norm and Euclidean \( \| \cdot \|_2 \) norm as \( \| \cdot \|_2 \).

Note that since saddle point problem for robust PageRank came from minimization problem, it is possible to use error bound on \( \Delta(x_n, y_n) \) for estimating function residual

\[ f(x_n) - f(x^*) \leq \Delta(x_n, y_n). \]

Partial subgradients for (2) are gradients

\[ q_x(x, y) = A^T y - y + \varepsilon \frac{x}{\|x\|_2}, \]
\[ q_y(x, y) = A x - x, \]

and, on sets \( X = S_N \) and \( Y = B_N, \) they lead to the evident uniform bounds on column-stochastic matrices \( A: \)

\[ \max_{x \in S_N} \|Ax - x\|_2 \leq \max_{y \in B_N} \|A^T y - y\|_2 + \varepsilon \max_{x \in S_N} \||x\|_\infty/\|x\|_2 \]
\[ \leq 2 + \varepsilon = L_{x,x}, \]
\[ \max_{x \in S_N} \|Ax - x\|_2 \leq 2 = L_{y,y}. \]

Final bound for deterministic MDA follows from (10) with appropriate constants

\[ f(x_n) - f(x^*) \leq \frac{\sqrt{n+1}}{n} \left( (2 + \varepsilon) \sqrt{\ln N} + \sqrt{2} \right) \]

\[^7\]It is interesting that almost the same bound with a factor around 2 was obtained in [11], c.f. Section 4, for two-sequences case and entropy proxy-function \( T \) by using norm variability and different \( \beta_k \) choice.

\[^8\]All matrix elements \( a(i,j) \) are nonnegative, and sums in each column \( \sum_{i=1}^N a(i,j), j = 1, \ldots, N \) are equal to 1.

3693
A. Randomization technique

Here we briefly describe randomization, which simplifies calculation for huge-dimensional problems, see [7]. The key point is that we introduce randomization artificially, generating some random variables, which allow to fallback from matrix-vector multiplications to sole vectors. Using such technique allows solving problems with dense matrices A, being out of reach of sparse optimization problem statement.

In each k-th step of stochastic (or randomized) MDA we are to obtain stochastic gradients. Let’s introduce two discrete independently distributed random variables (indices) \( \omega_k, \chi_k \in \{1, \ldots, N\} \), which are generated by the following conditional distributions:

\[
\mathbb{P}(\chi_k = j|(x_t, y_t)_{t \leq k}) = \frac{1}{N}, \quad j = 1, \ldots, N,
\]

\[
\mathbb{P}(\omega_k = i|(x_t, y_t)_{t \leq k}) = x^{(i)}_k, \quad i = 1, \ldots, N,
\]

and choose (random as well) vectors

\[
\phi_{k+1} = N(A^T)_{(\chi_k)y_k}(x_k) - y_k + \varepsilon \frac{x_k}{\|x_k\|_2},
\]

\[
\psi_{k+1} = A(\omega_k) - x_k,
\]

where \( A^{(i)} \) represents \( i \)-th column of matrix A. Comparison with (11) reveals that for these vectors property (7) holds so we can use them as stochastic subgradients. This randomization is slightly different from the one in [7], here used vectors themselves when possible.

Let us shortly denote conditional expectation \( \mathbb{E}_k \{ \cdot \} \equiv \mathbb{E}\{ \cdot |(x_t, y_t), t \leq k \} \). Bounds \( \mathbb{L}^2_{x,y}, \mathbb{L}^2_{x,y} \) are as follows:

\[
\mathbb{E}_k \{ \| \phi_k \|_2^2 \} = \sum_{j=1}^N \left\| N(A^T)_{(j)y_k}(j) - y_k + \varepsilon \frac{x_k}{\|x_k\|_2} \right\|_\infty^2 \frac{1}{N} \leq 2N \sum_{i=1}^N \left\| (A^T)_{(i)y_k}(i) \right\|_2^2 + 2 \left\| y_k - \varepsilon \frac{x_k}{\|x_k\|_2} \right\|_\infty^2 \leq 2N + 4 + 4\varepsilon^2 = \mathbb{L}^2_{x,y}.
\]

\[
\mathbb{E}_k \{ \| \psi_k \|_2^2 \} = \sum_{i=1}^N \left\| A^{(i)} - x_k \right\|_2^2 \leq \max_{i=1, \ldots, N} \| A^{(i)} - x_k \|_2^2 \leq 4 = \mathbb{L}^2_{x,y}.
\]

(12)

Though the first bound is not very satisfying due to \( N \) appearance, it gives final estimate

\[
\mathbb{E} f(\tau_n) - f(x^*) \leq \frac{N+1}{n} \left( \sqrt{2(N + 2 + 2\varepsilon^2) \ln N + \varepsilon^2} \right) \leq \sqrt{2 \frac{N \ln N}{n}}
\]

where \( \varepsilon^2 \) is supposed to be much less than \( N \). This bound says that at least approximately \( N \) iterations should be made to achieve \( O(1) \) error. It appears that in practice this algorithm performs faster, because any specific matrix \( A \) has smaller value \( \mathbb{L}_{x,x} \) and seems that bound is overrated for the problem as it will be shown in the example. There was different approach of eliminating \( N \) factor in (12) by using matrix \( H \) instead of \( A \), where \( H \) is Hadamard matrix and \( \hat{A} \) is extended up to needed dimension matrix \( A \) [7]. Unfortunately, there high complexity is implicit either in the premultilpuation, either on runtime complexity step. Namely, when using column randomization, getting random column of matrix \( (H \hat{A})^T \) in the expression of vector \( \phi \) in (12) still requires multiplication of matrix \( A \) to a vector.

B. Calculation complexity

Here we introduce calculation complexity in terms of the two components: matrix-vector multiplication (MVM) should be thought separately from a vector-vector operation (VVO) like sum of two vectors, vector to scalar multiplication or calculation of vector norm, etc. There is intuitive meaning of VVO as “(sequential) access to all elements of a vector”.

For deterministic algorithm, partial derivatives (11) require 2 MVMs and 5 VVOs (including 4 VVOs in gradients of \( W, U \)), plus averaging (9) takes 6 VVOs. For whole algorithm up to \( n \)-th step there are \( 2n \) MVMs and \( 19n \) VVOs in total.

For the randomized algorithm, it takes 5 VVO to calculate vectors \( \phi_k, \psi_k \), additionally there are \( 2N + 2 \log_2 N \) elementary operations for random value generation of index \( \omega_k \). The algorithm up to \( n \)-th step uses \( 19n \) VVOs and \( 2n(N + \log_2 N) \) elementary operations in total.

To make comparison between deterministic and stochastic bounds we are to define operation cost of each VVO and MVM. Cost of a VVO is a constant number of \( N \) elementary operations, but a MVM cost depends on how sparse matrix \( A \) is. Let’s put a parameter \( \nu \) as sparsity coefficient, with meaning that matrix \( A \) contains \( \nu N \) nonzero elements. Then one MVM cost is \( \nu N \) elementary operations.

Ergo for fixed \( n \) deterministic algorithm have complexity of \( (19 + 2\nu) n \) and stochastic algorithm have complexity of \( (21 + 2 \log_2 N) n \) elementary operations. Given that on error estimates are the same (for the matrix \( A \) of interest), the prior algorithm should be chosen with knowledge of matrix \( A \) structure, namely its sparsity.

C. Example

We take same graph of movies-related web sites as in [4] with \( N = 4754 \) vertices and put \( \varepsilon = 1 \). Reference solution was computed by MOSEK toolbox in Matlab [6]. In Fig. 1 there are \( q(\tau_k, y_k) \) values of deterministic and randomized MDAs for \( n = 10N \) iterations. Also plotted averaged (10 runs) iterations of randomized MDA. The error well inside derived bounds, these are even out of plot. It is seen that even with worse error bound the randomized MDA outperforms the deterministic one.

9We’ll later use the fact that normally sparse matrices \( A \) are used.

10These numbers may be decreased in small factor due to optimization, i.e. operation \( x_1 + ax_2 \) may be done in 1 VVO.

11Such as accessing one element of a vector, multiplying or adding two scalars, etc.
V. PROOFS

First there is given technical lemma’s proof with bounds on weighted sums of some scalar products for vectors produced by algorithm (5). This lemma uses Bregman distance properties. Then Theorems 1 and 2 with general bounds for subgradient and stochastic subgradients cases are proved.

Lemma 1 (cf. [3], with initial conditions refined): Let set $X \subset \mathbb{R}^N$ be compact and convex. Given arbitrary sequence $\phi_k \in \mathbb{R}^N, k = 1, \ldots, n$, positive sequence $\gamma_k, k = 1, \ldots, n$, positive non-decreasing sequence $\beta_k, k = 0, 1, \ldots, n-1$, proxy-function $V(\cdot)$ and its $\beta$-conjugate $W_\beta(\cdot)$, initial points $\zeta_0 = 0 \in \mathbb{R}^N, x_0 = -\nabla W_\beta(\zeta_0) \in X$, and process

$$
\zeta_k = \zeta_{k-1} + \gamma_k \phi_k, \quad x_k = -\nabla W_\beta(\zeta_k), \quad k = 1, \ldots, n,
$$

the following inequality holds

$$
\sum_{k=1}^n \gamma_k (x_{k-1} - x)^T \phi_k \leq \beta_n V^* - \beta_0 V_s + \sum_{k=1}^n \frac{\gamma_k^2}{2 \alpha_k \beta_{k-1}} \| \phi_k \|^2_{x_k},
$$

for all $x \in X$.

Proof of Lemma 1: Function $W_{\beta_k-1}(\cdot)$ is continuously differentiable and thus may be written in integral form

$$
W_{\beta_k-1}(\zeta_k) = W_{\beta_k-1}(\zeta_{k-1}) + \gamma_k \phi_k^T \nabla W_{\beta_k-1}(\zeta_{k-1}) + \gamma_k \int_0^1 \phi_k^T \left[ \nabla^2 W_{\beta_k-1}(\tau \zeta_k + (1 - \tau) \zeta_{k-1}) - \nabla W_{\beta_k-1}(\zeta_{k-1}) \right] d\tau
$$

$$
\leq W_{\beta_k-1}(\zeta_{k-1}) - \gamma_k \phi_k^T x_{k-1} + \frac{\gamma_k^2}{2 \alpha_k \beta_{k-1}} \| \phi_k \|^2_{x_{k-1}},
$$

where we used definition of $x_{k-1}$, equality $\zeta_k - \zeta_{k-1} = \gamma_k \phi_k$, inequality based on definition of dual norm $a^T b \leq \|a\|_{x_k} \|b\|_{x_k}$, and Lipschitz property on $\nabla W_\beta$. Noting that $W_{\beta_k}(\zeta_k) \leq W_{\beta_k-1}(\zeta_k)$ due to monotonic decrease of $W_\beta$ by $\beta$, and $\beta_i \geq \beta_{i-1}$, we can derive

$$
\gamma_k x_{k-1}^T \phi_k \leq W_{\beta_k-1}(\zeta_{k-1}) - W_{\beta_k}(\zeta_k) + \frac{\gamma_k^2}{2 \alpha_k \beta_{k-1}} \| \phi_k \|^2_{x_k}.\n$$

Summing over $k = 1, \ldots, n$ and subtracting $x^T \zeta_n = x^T \sum_{k=1}^n \gamma_k \phi_k$ from both sides we get

$$
\sum_{k=1}^n \gamma_k (x_{k-1} - x)^T \phi_k \leq W_{\beta_n}(\zeta_0) - W_{\beta_n}(\zeta_n) - x^T \zeta_n + \sum_{k=1}^n \frac{\gamma_k^2}{2 \alpha_k \beta_{k-1}} \| \phi_k \|^2_{x_k}.
$$

The final lemma statement follows from the fact that $W_{\beta_n}(\zeta_0) = W_{\beta_n}(0) = -\beta_n V_s$, and $-W_{\beta_n}(\zeta_n) - x^T \zeta_n \leq \max_{\zeta \in \mathbb{R}^N}(\beta_n V(\zeta) - x^T \zeta) = \beta_n V(x) \leq \beta_n V^*$ by duality of $W_\beta(\cdot)$ and $\beta V(\cdot)$.

Corollary 1: For the process

$$
\eta_k = \eta_{k-1} - \gamma_k \psi_k, \quad k = 1, \ldots, n,
$$

holds

$$
\sum_{k=1}^n \gamma_k (y - y_{k-1})^T \psi_k \leq \delta_n T_s - \delta_0 T_s^* + \frac{\gamma^2}{\alpha_0 \delta_{k-1}} \| \psi_k \|^2_{x_k}.\n$$

The corollary proof is the same as that of Lemma 1 with $y, \eta, \psi, T, U, \alpha, \beta, \| \cdot \|_{x_k}$ instead of $x, \zeta, \phi, V, W, \alpha, \| \cdot \|_{x_k}$. The only difference is that minus sign in formulae for $\eta_k$ results in different negation order of the left side expression.

Proof of Theorem 1: For brevity, let’s define

$$
P_n = R_n = \frac{n}{\frac{1}{\gamma_1} r_{1k} \sum_{k=1}^n \gamma_k^2}{\frac{1}{2} \left( \frac{L^2_{k,x}}{\alpha_k \beta_{k-1}} + \frac{L^2_{k,y}}{\alpha_k \beta_{k-1}} \right)}.\n$$

From convexity-concavity of function $q$ and the property of subgradients we have for all $(x, y) \in X \times Y, k = 0, \ldots, n-1,$

$$
q(x_k, y_k) - q(x_k, y_{k-1}) \leq (x_k - x_{k-1})^T q_x(x_{k-1}, y_k),
$$

$$
q(x_k, y_{k-1}) - q(x_k, y_k) \leq (y_{k-1} - y_k)^T q_y(x_k, y_k), (13)
$$

By summing both hand sides in (13), multiplying by $\gamma_k$, and applying the bounds obtained in Lemma 1 and Corollary 1 for the case $\phi_k = q_x(x_{k-1}, y_{k-1}), \psi_k = q_y(x_{k-1}, y_{k-1})$ we have

$$
\sum_{k=1}^n \gamma_k (q(x_k, y_{k-1}) - q(x_k, y_k)) \leq P_n.
$$

Using Jensen’s inequality with convexity (concavity) over arguments with nonnegative $\gamma_k$ we get

$$
q(x_n, y_n) = q \left( x_n \sum_{k=1}^n \gamma_k x_k, y_n \right) \leq \sum_{k=1}^n \gamma_k q(x_k, y),
$$

$$
-q(x_n, y_{n-1}) = -q \left( x_n \sum_{k=1}^n \gamma_k y_k \right) \leq \sum_{k=1}^n \gamma_k q(x, y_k),
$$

and conclude that $q(x_n, y_n) - q(x, y_n) \leq \frac{1}{\gamma_1} P_n$.

Taking a maximizer in $Y$ and minimizer in $X$ on the left-side difference, we arrive at the theorem’s stated bound on error $\Delta(x_n, y_n)$ (4).

Now we are to prove stochastic case. Its proof follows the one of Theorem 1 but steps are taken in different order.

Proof of Theorem 2: Similarly, denote

$$
Q_n = R_n = \frac{n}{\frac{1}{\gamma_1} r_{1k} \sum_{k=1}^n \gamma_k^2}{\frac{1}{2} \left( \frac{L^2_{k,x}}{\alpha_k \beta_{k-1}} + \frac{L^2_{k,y}}{\alpha_k \beta_{k-1}} \right)}.\n$$

3695
Sum the bounds proved in Lemma 1 and Corollary 1, then divide on \( \sum_{k=1}^{n} \gamma_k \), apply mathematical expectation with bounds (8), and get
\[
\mathbb{E} \left( \frac{1}{\sum_{k=1}^{n} \gamma_k} \sum_{k=1}^{n} \gamma_k (x_{k-1} - x)^T \phi_k \right) + \mathbb{E} \left( \frac{1}{\sum_{k=1}^{n} \gamma_k} \sum_{k=1}^{n} \gamma_k (y - y_{k-1})^T \psi_k \right) \leq \frac{Q^2}{\sum_{k=1}^{n} \gamma_k}.
\]
Consider only \( x \)-related terms like \( (x_{k-1} - x)^T \phi_k \). As soon each next point \( (x_k, y_k) \) depends on previous stochastic gradients \( \phi_1, \psi_1, t < k \), for each term we should sequentially take conditional expectation. I.e. for \( \mathbb{E}(\phi_1 | x_0, y_0) = q_2(x_0, y_0) \) it is evident as initial conditions are fixed. Next, \( \mathbb{E}(x_1 | x_0, y_0) = x_1 \) became deterministic\(^{12} \) function only on \( x_0, y_0 \), and do not depend on random \( \phi_1, \psi_1 \). Same applied to \( y \)-related terms \( (y - y_{k-1})^T \psi_k \). Step-by-step following timeline further with property (7) we get
\[
\frac{1}{\sum_{k=1}^{n} \gamma_k} \sum_{k=1}^{n} \gamma_k \mathbb{E}((x_{k-1} - x)^T q_2(x_{k-1}, y_{k-1})) \\
+ \frac{1}{\sum_{k=1}^{n} \gamma_k} \sum_{k=1}^{n} \gamma_k \mathbb{E}((y - y_{k-1})^T q_2(x_{k-1}, y_{k-1})) \leq \frac{1}{\sum_{k=1}^{n} \gamma_k} Q_n,
\]
where expectation is over variables \( x_k, y_k, k = 1, \ldots, n - 1 \) only. Next following subgradient inequalities (13) with Jensen’s ones (14), and taking expectation we get
\[
\mathbb{E}(q(x_n, y) - q(x, \bar{y}_n)) \leq \frac{1}{\sum_{k=1}^{n} \gamma_k} \sum_{k=1}^{n} \gamma_k \mathbb{E}((x_{k-1} - x)^T q_2(x_{k-1}, y_{k-1})) \\
+ \frac{1}{\sum_{k=1}^{n} \gamma_k} \sum_{k=1}^{n} \gamma_k \mathbb{E}((y - y_{k-1})^T q_2(x_{k-1}, y_{k-1}))
\]
which is valid for all \( x \in X, y \in Y \). Combining two previous inequalities (for any fixed \( n \) and taking maximum on \( y \in Y \) and minimum on \( x \in X \) leads to the theorem’s result.

VI. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In the paper extension of mirror descent algorithm for convex-concave saddle point problems is derived by introducing extra parameter sequence. This approach allows to obtain refined error bounds, explained proofs provided. The explored mirror descent algorithms (both deterministic and randomized) were applied to robust PageRank problem, followed by complexity analysis and comparison of these algorithms. Also minor fix in initial condition setup made for the algorithms’ grounding lemma.

B. Future Works

There are quite a number of possible improvements for this direction. First, in complexity analysis for robust PageRank calculation above no storage related and access-time related issues were touched, which would also affect complexity bounds, especially cost of MVM operations.

Some bound improvement could be reached by using adaptive alteration of algorithm’s parameters \( \gamma_k, \beta_k, \delta_k \) through iterations versus using a priori setup.

Another thought is that in contrast with deterministic algorithm stochastic one guarantees only estimate on expectation, with sampled random output and its quality (e.g. standard deviation) should be studied in future works as well. Still should be kept in mind that for huge dimensional problems small numeric calculation errors tend to accumulate, so deterministic algorithm may behave like stochastic one. For sparse problems a comparison with specific sparse optimization methods can be studied.

Finally, it is interesting to apply MDA for slightly different version of robust PageRank, using another norms, e.g. minimizing
\[
f_1(x) = \|A x - x\|_\infty + \varepsilon \|x\|_\infty.
\]

While optimization with such norms could hardly be done directly under high dimension, corresponding saddle point problem may be solved efficiently. The authors assume that constants (12) in this case won’t depend on \( N \) at all.

VII. ACKNOWLEDGMENTS

The authors thank Boris Polyak for encouraging discussion of robust PageRank, and anonymous referees for valuable notes and suggestions.

REFERENCES


