Kovalev et al. 2022. APDG

Machine Learning Reading Group Summer 2022

Betty Shea

2022-06-15

University of British Columbia

Why this paper?

Relevancy/ interest

- link with VRRG and game theory
- applications in RL
- composition with a linear map problems

- Victor covered another primal-dual method (e.g. SDCA)
- Ties in with the discussion on Fenchel conjugates
- Find minimum of an objective $\Leftrightarrow$ find saddle-point in a minmax problem
- Solve optimization problem $\Leftrightarrow$ solve a two-player game

Reinforcement learning

Example of an application

- RL task of estimating the value function $V^\pi(s)$ of a policy π given state s
- Use linear approximation $\tilde{V}^\pi(s)$ with model parameters x
- Learn x by minimizing the mean squared error based on a norm defined by a matrix containing feature vectors of states visited
- Requires inverting a (potentially large) matrix
- Avoid this by solving an equivalent saddle-point problem

Composition with a linear map problems

- $\min_x f(Ax)$ where A is a linear map
- Special case of convex-concave saddle-point problem with bilinear coupling
- APDG is a variant of the forward-backward algorithm
- Solves objectives in the form of a sum of composite convex functions

What is this paper about?

THE UNIVERSITY OF BRITISH COLUMBIA

Title of the paper

- Accelerated Primal-Dual Gradient Method (APDG) *for*
- Smooth and Convex-Concave Saddle-Point Problems *with*
- Bilinear Coupling

accelerated

- convergence rate could be expressed in terms of condition number $\kappa = L/\mu$
- generally, non-accelerated $\implies O(\kappa)$, accelerated $\implies O(\sqrt{\kappa})$
- many ways to accelerate, paper's method is similar to Nesterov's

primal-dual gradient method

- takes steps using both primal and dual variables
- takes steps using the negative gradient

Objective

$$\min_{x \in \mathbb{R}^{d_x}} \max_{y \in \mathbb{R}^{d_y}} F(x, y) : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$$

A saddle point (x_*, y_*) of F satisfies

$$F(x_*, y) \leq F(x_*, y_*) \leq F(x, y_*)$$

for any (x, y)

Smooth, Convex-Concave

L_{xy}-smooth means ($L_{xy} > 0$)

$$\|\nabla_x F(x, y_1) - \nabla_x F(x, y_2)\| \leq L_{xy} \|y_1 - y_2\|$$

$$\|\nabla_y F(x_1, y) - \nabla_y F(x_2, y)\| \leq L_{xy} \|x_1 - x_2\|$$

Convex-concave means for any point (x_*, y_*)

$$x \mapsto F(x, y_*) \text{ is convex}$$

$$y \mapsto F(x_*, y) \text{ is concave}$$

Bilinear Coupling Problems

THE UNIVERSITY OF BRITISH COLUMBIA

$$\min_{x \in \mathbb{R}^{d_x}} \max_{y \in \mathbb{R}^{d_y}} F(x, y) = f(x) + y^\intercal Ax - g(y)$$

where $f(x) : \mathbb{R}^{d_x} \to \mathbb{R}$, $g(y) : \mathbb{R}^{d_y} \to \mathbb{R}$, $A \in \mathbb{R}^{d_x \times d_y}$

- A is a "coupling matrix" (that ties payoff of minimizer and maximizer)
- A is a matrix of the bilinear form
- paper has additional assumptions on A

Summary

THE UNIVERSITY OF BRITISH COLUMBIA

Contributions

- Two algorithms proposed
  - APDG for smooth, convex-concave, saddle-point problems with bilinear coupling
  - Gradient Descent-Ascent Method with Extrapolation (GDAE) for general smooth, convex-concave, saddle-point problems
- Algorithms allow for "direct" acceleration
- APDG convergence matches theoretical lower bound where known
- GDAE convergence nearly as good as SOTA

APDG (Algorithm 1)

Algorithm 1 APDG: Accelerated Primal-Dual Gradient Method

Input: $x^0 \in \mathrm{range}\mathbf{A}^\top$, $y^0 \in \mathrm{range}\mathbf{A}$, $\eta_x, \eta_y, \alpha_x, \alpha_y, \beta_x, \beta_y > 0$, $\tau_x, \tau_y, \sigma_x, \sigma_y \in (0,1]$, $\theta \in (0,1)$

$x_f^0 = x^0$

$y_f^0 = y^{-1} = y^0$

for $k = 0, 1, 2, \ldots$ **do**

  $y_m^k = y^k + \theta(y^k - y^{k-1})$

  $x_g^k = \tau_x x^k + (1-\tau_x) x_f^k$

  $y_g^k = \tau_y y^k + (1-\tau_y) y_f^k$

  $x^{k+1} = x^k + \eta_x \alpha_x (x_g^k - x^k) - \eta_x \beta_x \mathbf{A}^\top (\mathbf{A} x^k - \nabla g(y_g^k)) - \eta_x \left(\nabla f(x_g^k) + \mathbf{A}^\top y_m^k\right)$

  $y^{k+1} = y^k + \eta_y \alpha_y (y_g^k - y^k) - \eta_y \beta_y \mathbf{A}(\mathbf{A}^\top y^k + \nabla f(x_g^k)) - \eta_y (\nabla g(y_g^k) - \mathbf{A} x^{k+1})$

  $x_f^{k+1} = x_g^k + \sigma_x (x^{k+1} - x^k)$

  $y_f^{k+1} = y_g^k + \sigma_y (y^{k+1} - y^k)$

end for

APDG (Algorithm 1): Parameters

THE UNIVERSITY OF BRITISH COLUMBIA

Accelerated Primal-Dual Gradient Method for Smooth and Convex-Concave Saddle-Point Problems with Bilinear Coupling

Algorithm 1 APDG: Accelerated Primal-Dual Gradient Method

Input: $x^0 \in \mathrm{range}\,\mathbf{A}^\top$, $y^0 \in \mathrm{range}\,\mathbf{A}$, $\boxed{\eta_x, \eta_y}$, $\boxed{\alpha_x, \alpha_y}$, $\boxed{\beta_x, \beta_y} > 0$, $\boxed{\tau_x, \tau_y}$, $\boxed{\sigma_x, \sigma_y} \in (0, 1]$, $\boxed{\theta} \in (0, 1)$

$x_f^0 = x^0$ — overall learning rate — weight for accelerated component

$y_f^0 = y^{-1} = y^0$ — forward-backward parameter

for $k = 0, 1, 2, \dots$ **do**

$y_m^k = y^k + \boxed{\theta}(y^k - y^{k-1})$ — how much to extrapolate

$x_g^k = \boxed{\tau_x} x^k + (1 - \boxed{\tau_x}) x_f^k$ — how much acceleration

$y_g^k = \boxed{\tau_y} y^k + (1 - \boxed{\tau_y}) y_f^k$

$x^{k+1} = x^k + \boxed{\eta_x}\boxed{\alpha_x}(x_g^k - x^k) - \boxed{\eta_x}\boxed{\beta_x}\mathbf{A}^\top(\mathbf{A}x^k - \nabla g(y_g^k)) - \boxed{\eta_x}(\nabla f(x_g^k) + \mathbf{A}^\top y_m^k)$

$y^{k+1} = y^k + \boxed{\eta_y}\boxed{\alpha_y}(y_g^k - y^k) - \boxed{\eta_y}\boxed{\beta_y}\mathbf{A}(\mathbf{A}^\top y^k + \nabla f(x_g^k)) - \boxed{\eta_y}(\nabla g(y_g^k) - \mathbf{A}x^{k+1})$

$x_f^{k+1} = x_g^k + \boxed{\sigma_x}(x^{k+1} - x^k)$ — how much momentum

$y_f^{k+1} = y_g^k + \boxed{\sigma_y}(y^{k+1} - y^k)$

end for

APDG (Algorithm 1): Updates

Accelerated Primal-Dual Gradient Method for Smooth and Convex-Concave Saddle-Point Problems with Bilinear Coupling

Algorithm 1 APDG: Accelerated Primal-Dual Gradient Method

Input: $x^0 \in \mathrm{range}\,\mathbf{A}^\top, y^0 \in \mathrm{range}\,\mathbf{A}, \eta_x, \eta_y, \alpha_x, \alpha_y, \beta_x, \beta_y > 0, \tau_x, \tau_y, \sigma_x, \sigma_y \in (0,1], \theta \in (0,1)$

$x_f^0 = x^0$

$y_f^0 = y^{-1} = y^0$

for $k = 0, 1, 2, \ldots$ **do**

$\quad y_m^k = y^k + \theta(y^k - y^{k-1})$ Linear extrapolation step on newly introduced variable

$\quad x_g^k = \tau_x x^k + (1-\tau_x) x_f^k$

$\quad y_g^k = \tau_y y^k + (1-\tau_y) y_f^k$

Acceleration

$\quad x^{k+1} = x^k + \eta_x \alpha_x (x_g^k - x^k) - \eta_x \beta_x \mathbf{A}^\top (\mathbf{A} x^k - \nabla g(y_g^k)) - \eta_x \left(\nabla f(x_g^k) + \mathbf{A}^\top y_m^k\right)$

$\quad y^{k+1} = y^k + \eta_y \alpha_y (y_g^k - y^k) - \eta_y \beta_y \mathbf{A} (\mathbf{A}^\top y^k + \nabla f(x_g^k)) - \eta_y (\nabla g(y_g^k) - \mathbf{A} x^{k+1})$

Forward-Backward

$\quad x_f^{k+1} = x_g^k + \sigma_x (x^{k+1} - x^k)$

$\quad y_f^{k+1} = y_g^k + \sigma_y (y^{k+1} - y^k)$

Momentum

end for

APDG (Algorithm 1): What is it trying to do?

THE UNIVERSITY OF BRITISH COLUMBIA

Minmax problem

$$\min_{x \in \mathbb{R}^{d_x}} \max_{y \in \mathbb{R}^{d_y}} F(x,y) = f(x) + y^\intercal A x - g(y)$$

Finding a saddle point (x_*, y_*) means satisfying first order optimality conditions

$$\begin{cases} \nabla_x F(x_*, y_*) = \nabla f(x_*) + A^\intercal y_* = 0 \\ \nabla_y F(x_*, y_*) = -\nabla g(y^*) + A x_* = 0 \end{cases}$$

APDG (Algorithm 1): What is it trying to do?

Requires solving linear system

$$\begin{cases} x^{+} = x - A^{\intercal} y^{+} \\ y^{+} = y + A x^{+} \end{cases}$$

Closed form solution needs inverting a matrix in the form

$$(I + A^{\intercal} A) \text{ or } (I + A A^{\intercal})$$

APDG (Algorithm 1): What is it trying to do?

Instead, introduce a new variable y_m and solve iteratively

$$\begin{cases} x^+ = x - A^\intercal y_m \\ y^+ = y + Ax^+ \end{cases}$$

What to set y_m? Paper suggests linear extrapolation step

$$y_m = y + \theta(y - y^-)$$

where y^- is the value at the iteration previous to y

APDG

- Optimal for
  - strongly-convex-strongly-concave problems
  - affinely constrained minimization case (i.e. $\min_{Ax=b} f(x)$))
- Beats SOTA for
  - strongly-convex-concave case (unknown lower bound)
  - convex-concave case (unknown lower bound)
- Worse than SOTA for bilinear case

$$\min_{x\in\mathbb{R}^{d_x}} \max_{y\in\mathbb{R}^{d_y}} a^\intercal x + y^\intercal Ax - b^\intercal y$$

Results

Strongly-convex-strongly-concave case (Section 5.1)	
Algorithm 1	$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_x}{\mu_x}}, \sqrt{\frac{L_y}{\mu_y}}, \frac{L_{xy}}{\sqrt{\mu_x\mu_y}}\right\}\log\frac{1}{\epsilon}\right)$
Lower bound Zhang et al. (2021b)	$\mathcal{O}\left(\max\left\{\sqrt{\frac{L_x}{\mu_x}}, \sqrt{\frac{L_y}{\mu_y}}, \frac{L_{xy}}{\sqrt{\mu_x\mu_y}}\right\}\log\frac{1}{\epsilon}\right)$
DIPPA Xie et al. (2021)	$\tilde{\mathcal{O}}\left(\max\left\{\sqrt[4]{\frac{L_x^2 L_y}{\mu_x^2\mu_y}}, \sqrt[4]{\frac{L_x L_y^2}{\mu_x\mu_y^2}}, \frac{L_{xy}}{\sqrt{\mu_x\mu_y}}\right\}\log\frac{1}{\epsilon}\right)$
Proximal Best Response Wang & Li (2020)	$\tilde{\mathcal{O}}\left(\max\left\{\sqrt{\frac{L_x}{\mu_x}}, \sqrt{\frac{L_y}{\mu_y}}, \sqrt{\frac{L_{xy}L}{\mu_x\mu_y}}\right\}\log\frac{1}{\epsilon}\right)$
Affinely constrained minimization case (Section 5.2)	
Algorithm 1	$\mathcal{O}\left(\frac{L_{xy}}{\mu_{xy}}\sqrt{\frac{L_x}{\mu_x}}\log\frac{1}{\epsilon}\right)$
Lower bound Salim et al. (2021)	$\mathcal{O}\left(\frac{L_{xy}}{\mu_{xy}}\sqrt{\frac{L_x}{\mu_x}}\log\frac{1}{\epsilon}\right)$
OPAPC Kovalev et al. (2020)	$\mathcal{O}\left(\frac{L_{xy}}{\mu_{xy}}\sqrt{\frac{L_x}{\mu_x}}\log\frac{1}{\epsilon}\right)$
Strongly-convex-concave case (Section 5.3)	
Algorithm 1	$\mathcal{O}\left(\max\left\{\frac{\sqrt{L_x L_y}}{\mu_{xy}}, \frac{L_{xy}}{\mu_{xy}}\sqrt{\frac{L_x}{\mu_x}}, \frac{L_{xy}^2}{\mu_{xy}^2}\right\}\log\frac{1}{\epsilon}\right)$
Lower bound	N/A
Alt-GDA Zhang et al. (2021a)	$\mathcal{O}\left(\max\left\{\frac{L^2}{\mu_{xy}^2}, \frac{L}{\mu_x}\right\}\log\frac{1}{\epsilon}\right)$
Bilinear case (Section 5.4)	
Algorithm 1	$\mathcal{O}\left(\frac{L_{xy}^2}{\mu_{xy}^2}\log\frac{1}{\epsilon}\right)$
Lower bound Ibrahim et al. (2020)	$\mathcal{O}\left(\frac{L_{xy}}{\mu_{xy}}\log\frac{1}{\epsilon}\right)$
Azizian et al. (2020)	$\mathcal{O}\left(\frac{L_{xy}}{\mu_{xy}}\log\frac{1}{\epsilon}\right)$
Convex-concave case (Section 5.5)	
Algorithm 1	$\mathcal{O}\left(\max\left\{\frac{\sqrt{L_x L_y L_{xy}}}{\mu_{xy}^2}, \frac{L_{xy}^2}{\mu_{xy}^2}\right\}\log\frac{1}{\epsilon}\right)$
Lower bound	N/A

Related topics

Operator splitting

- Suppose objective involves smooth f and possibly nonsmooth g

$$\min_{x \in \mathbb{R}^n} f(x) + g(x)$$

- First-order optimality of x_* and introduce $\lambda > 0$

$$0 \in \lambda \nabla f(x_*) + \lambda \partial g(x_*)$$

- Can think of solution x_* as the fixed point of

$$x \mapsto prox_{\lambda g}\left(x - \lambda \nabla f(x)\right) \text{ for all } \lambda > 0$$

  which motivates the iterative approach

Related topics

Fenchel game

- Can rewrite objective using its Fenchel conjugate

$$\min_x f(x) = \min_x \max_y \langle x, y \rangle - f^*(y)$$

 if f is convex, proper and closed.
- All players playing no-regret algorithms $\implies$ converge to a Nash equilibrium (in 2-player general sum game) $\implies$ find saddle point
- Solve convex optimization problems using no-regret game dynamics

References

- Dmitry Kovalev, Alexander Gasnikov and Peter Richtárik. 2022 *Accelerated Primal-Dual Gradient Method for Smooth and Convex-Concave Saddle-Point Problems with Bilinear Coupling*
- P.L. Combettes, L.Condat, J.-C. Pesquet and B.C.Vũ. 2014. *A Forward-Backward View of some Primal-Dual Optimization Methods in Image Recovery*
- David G. Luenberger and Yinyu Ye. 2008. *Linear and Nonlinear Programming. 3rd ed.*
- Jun-Kun Wang, Jacob Abernethy and Kfir Y. Levy. 2021. *No-Regret Dynamics in the Fenchel Game: A Unified Framework for Algorithmic Convex Optimization*

Additional slides: application to RL

THE UNIVERSITY OF BRITISH COLUMBIA

- Estimate value function of a policy π

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, \pi\right]$$

with discount factor $\gamma \in (0,1)$, reward r, state s

- Use linear approximation of $V^\pi(s) = \phi(s)^\intercal x$ instead where $\phi(s)$ is a feature vector of state s and x is the model parameters
- Minimize mean squared projected Bellman error

$$\min_x \|Bx - b\|^2_{C^{-1}}$$

requires inverting $C = \sum_{t=1}^n \phi(s_t)\phi(s_t)^\intercal$

- Equivalently solve saddle-point problem

$$\min_x \max_y -2y^\intercal Bx - \|y\|^2_C + 2b^\intercal y$$

Additional slides: composition w/ linear map

- $\min_z f(Az)$ where A is a linear map
- Rewrite as min-max problem

$$\min_z f(Az) \equiv \min_{x=Az} f(x) \equiv \min_{A^{-1}x=z} f(x) \equiv \min_x \max_y f(x) + y^\intercal \left(A^{-1}x - z\right)$$

- Forward-backward algorithm for problems of the form

$$\min_{x\in\mathcal{H}} \sum_{i=1}^{m} g_i(L_i x)$$

where $\mathcal{H}$ and $(\mathcal{G})_{1\leq i\leq m}$ are Hilbert spaces, g_i is proper lower semi-continuous convex from $\mathcal{G}_i$ to $(-\infty,\infty]$ and L_i is a bounded linear operator from $\mathcal{H}$ to $\mathcal{G}_i$.

Additional slides: operator splitting notes

$$0 \in \lambda \nabla f(x^*) + \lambda \partial g(x^*) \text{ for all } \lambda > 0$$

$$0 \in (\lambda \nabla f(x^*) - x^*) + (x^* + \lambda \partial g(x^*))$$

$$(Id - \lambda \nabla f)(x^*) \in (Id + \lambda \partial g)(x^*)$$

$$x^* \in (Id + \lambda \partial g)^{-1}(Id - \lambda \nabla f)(x^*)$$

Define proximal operator

$$prox_{\lambda g}(x) \triangleq (Id + \lambda \partial g)^{-1}(x)$$

x^* is unique and so

$$x^* = \text{prox}_{\lambda g}(x^* - \lambda \nabla f(x^*)) \text{ for all } \lambda > 0$$

Hence x^* is a fixed point of

$$x \mapsto \text{prox}_{\lambda g}(x - \lambda \nabla f(x))$$