OPT2020: 12th Annual Workshop on Optimization for Machine Learning

Primal-Dual Sequential Subspace Optimization for Saddle-point Problems

Yoni Choukroun CHOUKROUN.YONI@GMAIL.COM
Huawei

Michael Zibulevsky MZIBUL@GMAIL.COM
Technion, IIT

Pavel Kisilev PAVEL.KISILEV@HUAWEI.COM
Huawei

Abstract

We introduce a new sequential subspace optimization method for large-scale saddle-point problems. It solves iteratively a sequence of auxiliary saddle-point problems in low-dimensional subspaces, spanned by directions derived from first-order information over the primal *and* dual variables. Proximal regularization is further deployed to stabilize the optimization process. Experimental results demonstrate significantly better convergence relative to popular first-order methods. We analyze the influence of the subspace on the convergence of the algorithm, and assess its performance in various deterministic optimization scenarios, such as bi-linear games, ADMM-based constrained optimization and generative adversarial networks.

1. Introduction

Saddle-point problems arise in many applications, such as game theory [16], constrained and robust optimization [1, 2] and generative adversarial networks (GANs) [14]. Important variational problems such as ℓ_∞ minimization, convex segmentation or compressed sensing [5, 7] have saddle-point formulations that are efficiently handled using primal-dual solvers.

In case of large scale optimization problems, there is a need for optimization algorithms whose storage requirement and computational cost per iteration grow at most linearly with the problem dimensions. In the context of minimization, this constraint has led to the development of a broad family of methods such as variable metric methods, and *subspace optimization*[8, 9, 12, 15, 20–22]. In this work, motivated by the inherent slowness of gradient based methods and the power of subspace optimization, we extend the idea of subspace optimization, until now limited to minimization, to saddle-point problems. Specifically, we solve sequentially low dimensional saddle-point problems in subspaces defined by first-order information. We propose to perform the subspace optimization over the primal *and* dual variables, allowing to search for a saddle-point in a richer subspace, wherein the function can increase and/or decrease in primal and dual variables respectively. Further, we propose to couple the saddle-point objective with proximal operators in order to ensure the existence of a stationary point in the subspace. We solve the subspace optimization via adapted second order optimization that can be implemented efficiently in the given low dimensional subspace. Finally, we perform backtracking line search over the gradient norm. This ensures faster convergence, and most importantly, prevents divergence in degenerative cases. Experimental results assess the power and usefulness of the proposed method.

2. Sequential Subspace Optimization for saddle-point Problems

We consider the unconstrained saddle-point problem

$$\min_{x\in\mathbb{R}^M}\max_{y\in\mathbb{R}^N} f(x,y), \tag{1}$$

where f is twice continuously differentiable and the first derivative is L Lipschitz continuous on $\mathbb{R}^M \times \mathbb{R}^N$. Finding a global saddle-point is computationally intractable. We therefore assume local convexity-concavity of the objective in which case saddle-point solution (x^*, y^*) holds in a *local* r neighborhood ball of (x^*, y^*), which we denote $B_2((x^*, y^*), r)$.

Let us define the subspace saddle-point problem as

$$\min_{\alpha\in\mathbb{R}^m}\max_{\beta\in\mathbb{R}^n} f(x_k + P_k\alpha, y_k + Q_k\beta), \tag{2}$$

where we assume $m \ll M$ and $n \ll N$. Matrices P_k and Q_k define the subspace structure at iteration k. The subspace optimization can be solved exactly *or* approximately. The new iterate is of the form

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \eta_k \begin{pmatrix} P_k\alpha \\ Q_k\beta \end{pmatrix}, \tag{3}$$

where η_k is the step size obtained via outer optimization (i.e. original problem), and the procedure stops if convergence tests are satisfied. This formulation allows flexibility in definition of the search space. The following simple but challenging example illustrates this property. The bi-linear game $f(x, y) = x^T y$ [29] diverges when search is performed over one dimensional anti-gradient/gradient direction. However, convergence can be reached if the optimization is performed separately over the primal and dual variables, as shown in the following theorem. Proofs are provided in the Appendix A.

Theorem 1 *Consider the saddle-point problem $f(x, y) = x^T y$ and the update from eq. (3), where α and β are obtained by solving eq.(2) with $P_k = \nabla_x f(x_k, y_k)$ and $Q_k = \nabla_y f(x_k, y_k)$. Then, $\forall\eta_k \in (0, 2f(x_k, y_k)^2/\|P_k\|^2\|Q_k\|^2)$ the procedure converges to optimum.*
Also, the gradient method (i.e. $-\alpha = \beta > 0$), diverges $\forall\eta_k > 0$.

We emphasize the fact that *joint* subspace optimization is convergent while independent (alternating) subspace optimization is divergent in this unstable case. Following the subspace minimization strategy to use more than current gradient, we seek a saddle-point in the subspace spanned by first order information. Namely, we use the *mandatory* (Nemirovskii [22]) current gradient, previous gradients, and the previous search steps in x and y, such that $\text{span}\{P_k\} = \text{span}\{S_k^x, G_k^x\}$ and $\text{span}\{Q_k\} = \text{span}\{S_k^y, G_k^y\}$, where $S_k^u = \{p_{k-l-1}^u, \ldots, p_{k-1}^u\}$ with $p_k^u = u_k - u_{k-1}$, and $G_k^u = \{\nabla_u f(x_{k-l}, y_{k-l}), \ldots, \nabla_u f(x, y)\}$. Other directions can be used or added to improve the convergence as well. Expanding the subspace with more directions can enrich the subspace but enables a subjective trade-off between computational cost and speed of convergence. Such subspace formulation generalizes popular methods, e.g. the gradient method [28] or Optimistic Mirror Descent [10, 27]. In order to improve the convergence, the proposed framework can be combined with other methods, such as weighted averaging of iterates as final solution [4], or consensus optimization [18] as modification of the objective.

2.1. General Subspace Convergence Analysis

In this section we analyse the convergence conditions of the subspace optimization method, in terms of the norm of gradient, i.e. convergence to stationary point. Consider $z_k = [x_k, y_k]$ where $[\cdot,\cdot]$ denotes vectors concatenation. We denote the direction $d_k = [P_k\alpha, Q_k\beta] = R_k[\alpha,\beta] := R_k\gamma$, where R_k is the block matrix populated with P_k and Q_k in the block diagonal and zero elsewhere. Here we assume R_k has linearly independent columns. From the first order expansion, there exists sufficiently small $\zeta > 0$ such that

$$f(z_{k+1}) = f(z_k + \zeta d_k) = f(z_k) + \zeta\langle\nabla f(z_k), d_k\rangle + o(\zeta^2\|d_k\|). \tag{4}$$

Thus, by taking derivative of eq.(4) we have $\nabla f(z_{k+1}) \approx \nabla f(z_k) + \zeta\nabla^2 f(z_k)d_k$. Since we are interested in decreasing the gradient norm to reach convergence, we have

$$\|\nabla f(z_{k+1})\|^2 = \|\nabla f(z_k)\|^2 + 2\zeta\nabla f(z_k)^T\nabla^2 f(z_k)d_k + o(\zeta^2\|\nabla^2 f(z_k)d_k\|). \tag{5}$$

Thereafter, the sufficient condition for convergence to local stationary point is $\nabla f(z_k)^T\nabla^2 f(z_k)d_k < 0$. For example, the steepest descent/ascent direction is convergent in the case of strongly convex-concave problem since in that case $\nabla f(z_k)^T\nabla^2 f(z_k)\nabla f(z_k) < 0$. We can reformulate the previous equation in terms of the subspace parameters such that, since $\nabla_\gamma f(z + R\gamma) = R^T\nabla_z f(z + R\gamma)$, we have

$$\begin{aligned}\|\nabla f(z_{k+1})\|^2 &\approx \|\nabla f(z_k)\|^2 + 2\zeta\nabla f(z_k)^T\nabla^2 f(z_k)R_k\gamma \\ =&\|\nabla f(z_k)\|^2 + 2\zeta\nabla_\gamma f(z_k + R_k\gamma)^T\big|_{\gamma=0}R_k^{+T}R_k^+\nabla_\gamma^2 f(z_k + R_k\gamma)\big|_{\gamma=0}\gamma,\end{aligned} \tag{6}$$

where, R_k^+ denote the Moore–Penrose pseudoinverse of matrix R_k^T. Thus, assuming single Newton step in subspace $\gamma = -\nu\nabla_\gamma^2 f(z_k + R_k\gamma)^{-1}\big|_{\gamma=0}\nabla_\gamma f(z_k + R_k\gamma)\big|_{\gamma=0}$, $\exists\nu > 0$ such that

$$\|\nabla f(z_{k+1})\|^2 = \|\nabla f(z_k)\|^2 - 2\nu\zeta\|R_k^+\nabla_\gamma f(z_k + R_k\gamma)\big|_{\gamma=0}\|^2. \tag{7}$$

According to eq (7), in the neighborhood of the current point z_k, Newton step in the subspace domain decreases gradient norm of the original problem, and thus induces global convergence to stationary point. However, *contrary to the minimization setup* [8] where subspace optimization are descent methods, exact convergence to stationary point in subspaces (i.e. $\nabla_\gamma f(z_k + R_k\gamma) = 0$) does not necessarily enforce convergence in the original problem space, as shown in Theorem 1. This is mainly due to the interaction term $\nabla_{xy} f(x, y)$ present in eq. (5) via the Hessian matrix. Thus, the extension of subspace optimization to saddle-point problems is then not straightforward.

2.2. Convergence Improvement Strategies

To ensure convergence of inner and outer optimization, we propose to solve the subspace optimization in a constrained local region. Also, we propose to correct the direction obtained from the subspace optimization by controlling the outer step size η_k as in eq. (3) via an adapted line-search procedure.

2.2.1. PROXIMAL REGULARIZATION

In the following, we extend the auxiliary subspace saddle problem (2) by adding proximal point regularization [17]. At each iteration we solve the subspace proximal problem over $\tilde{f}(x, y)$, namely

$$\begin{aligned}\min_{\alpha\in\mathbb{R}^m}\max_{\beta\in\mathbb{R}^n}\tilde{f}(x_k+P_k\alpha, y_k+Q_k\beta) :=& f(x_k+P_k\alpha, y_k+Q_k\beta)\\ &+\frac{\tau_k}{2}\|x_k+P_k\alpha-\bar{x}_k\|^2-\frac{\tau_k}{2}\|y_k+Q_k\beta-\bar{y}_k\|^2,\end{aligned} \tag{8}$$

where $\bar{x}_k$ and $\bar{y}_k$ denote the primal and dual prox-centers respectively (e.g. moving average or previous point). The proximal approach motivation is two-fold. First, it allows averaging over iterations, reducing the oscillation behavior typical to min-max games [29], and improves stability of the optimization procedure. Foremost, it ensures the existence of a saddle-point in potentially degenerate subspaces, avoiding divergence, in a trust-region fashion [30].

2.2.2. SADDLE-POINT BACKTRACKING LINE SEARCH

Common line-search backtracking methods [25] cannot be applied straightforwardly to the saddle-point problems, since implementing search over the function primal and dual values can diverge (e.g. bi-linear game). To tackle this problem, we perform backtracking line search over the gradient norm to both ensure faster convergence of the method, and, most importantly prevent potential divergence after the inner subspace optimization. The proposed procedure is described in Algorithm 1.

This step size search procedure is used in *both* inner (fast convergence in subspace) and outer (step correction) optimization. In our experiments we chose $c = 0$ for less computational overhead and set $\nu = 0.5$. We further limit the number of line-search iterations to 30. The following theorem, based on the analysis of Section 2.1, states the convergence of the proposed algorithm for the standard gradient method (memoryless subspaces).

Algorithm 1: Saddle Backtracking Line Search

Input : $f : \mathbb{R}^{M\times N} \to \mathbb{R}$, current point z_k, direction d_k, $c \in [0, 1), \nu \in (0, 1), \eta \leq 1$

Output: Step size η

while $\|\nabla f(z_k + \eta d_k)\|^2 \geq \|\nabla f(z_k)\|^2 + \eta c \nabla f(z_k)^T \nabla^2 f(z_k) d_k$ **do**

  $\eta = \eta * \nu$;

end

return η;

Theorem 2 *Consider function $f(x, y)$ with* stable *saddle-point (x^*, y^*). Assume the subspace is spanned by the anti-gradient and gradient directions for the primal and dual variables, respectively. Then, the procedure of Algorithm 1 converges to the optimum for every $(x_0, y_0) \in B_2((x^*, y^*), r)$.*

2.3. Efficient Second-Order Saddle-point Optimization in Subspace

Second order methods aim at finding roots of the gradient via solution of the second order expansion. Therefore, they can converge extremely fast to saddle-points, especially in the proximity of the solution where the problem has a good quadratic approximation. The major drawback is the prohibitive computational cost for both the computation and inversion of the Hessian. However, in our small subspace setting, second order methods, s.a. (Quasi-)Newton, can be handled efficiently. In particular, the computation of the Hessian in the subspace is performed via Hessian product with the direction vectors [26]. Nowadays, it can be handled efficiently via automatic differentiation tool,

Algorithm 2: Sequential Subspace Saddle-point Optimization

Input : $f : (\mathbb{R}^M \times \mathbb{R}^N) \to \mathbb{R}$, initial point $z_0 = (x_0, y_0)$, d the maximum subspace dimension, K the maximum number of iterations, ϵ the machine precision

Output: (x_{final}, y_{final})

$\tau \in \mathbb{R}_0^+, \nu \in (0, 1)$;
Initialize proximal centers $\bar{x}_0$, $\bar{y}_0$;
for $k = 0, 1, ..., K$ **do**
  if $\|\nabla f(x_k, y_k)\| < \epsilon$ **then**: **return** (x_k, y_k);
  if $\|\nabla \tilde{f}(x_k, y_k)\| < \epsilon$ **then**: $\tau = \nu\tau$;
  Update P_k and Q_k with current gradients ;
  Set $t = 0$ and $\gamma_t = 0$;
  while $\|\nabla_\gamma f(z_k + R_k\gamma_t)\| > \epsilon$ **do**
    $\bar{\gamma} = -\tilde{H}_\gamma^{-1}(z_k + R_k\gamma_t)\nabla_\gamma \tilde{f}(z_k + R_k\gamma_t)$, eq. (9);
    Find inner step size η_{in} following Algorithm 1 over subspace objective;
    Set $\gamma_{t+1} = \gamma_t + \eta_{in}\bar{\gamma}$ and $t = t + 1$;
  end
  Find outer step size η_{out} following Algorithm 1;
  Update $z_{k+1} = z_k + \eta_{out} R_k \gamma_t$;
  if $dim(P_k) > d - 1$ **then**: Remove the oldest direction from P_k and Q_k;
  Update P_{k+1} and Q_{k+1} with search steps and/or gradients;
  Update proximal centers $\bar{x}_k$, $\bar{y}_k$ (e.g with x_k and y_k);
end
return (x_K, y_K);

since $(\partial^2 f) \cdot v = \partial(\partial f \cdot v)$. Hessian inversion is computationally negligible in *low dimensional* subspace (generally up to ten dimensions). The method can be further accelerated using frozen or truncated Hessian strategies, especially when the Hessian remains almost unchanged in the vicinity of the solution.

The second order proximal subspace optimization is performed iteratively until the convergence (or maximum number of iterations) is reached, as follows:

$$\begin{aligned}
\gamma_{k+1} &= \gamma_k - \eta_k \tilde{H}_\gamma^{-1}(z_k + R_k\gamma_k)\nabla_\gamma \tilde{f}(z_k + R_k\gamma_k) \\
&= \gamma_k - \eta_k \big(R_k^T(\tilde{H}_z(z_k + R_k\gamma_k))R_k\big)^{-1} R_k^T \nabla_z \tilde{f}(z_k + R_k\gamma_k) \\
&= \gamma_k - \eta_k \big(R_k^T(H_z(z_k + R_k\gamma_k) + \mathrm{T})R_k\big)^{-1} R_k^T \nabla_z \tilde{f}(z_k + R_k\gamma_k),
\end{aligned} \tag{9}$$

where the last two equations illustrate the computational complexity of the method, through the Hessian-vector product over the subspaces matrix R_k, and the low dimensional subspace Hessian inversion. Here, the matrices $H_u(v)$ and $\tilde{H}_u(v)$ denote the variable metric matrices reduced to $\nabla_u^2 f(v)$ and $\nabla_u^2 \tilde{f}(v)$ respectively in the Newton scheme. Also, T denote the dampening matrix that ensures stability of the former saddle-point system, such that $\mathrm{T} = \tau \left(\begin{array}{c|c} I & 0 \\ \hline 0 & -I \end{array}\right)$. Here η_k is the step size commonly obtained via the line search procedure. In the case of the Newton optimization in the subspace being computationally intensive (e.g. high dimensional subspace or prohibitive derivatives computation), a Quasi-Newton method can be deployed instead. In the saddle-point set-

ting, Quasi-Newton alternatives that do *not* enforce positive definiteness of the Hessian can be used, such as symmetric rank-one (SR1) with usual handling of the update factors [24]. We summarize the proposed sequential subspace optimization framework for saddle-point problems in Algorithm 2.

3. Experimental Results

To assess the performance of the proposed method, and to demonstrate its efficacy, we performed several experiments. GDA refers to the gradient method [28], DAS refers to the dual averaging scheme of [23], OGDA is the optimistic gradient method [11], CP refers to Chambolle-Pock algorithm [6], and EGDA refers to the extrinsic gradient method [31]. Detailed description of *all* the experiments settings is provided in the Appendix C. Results are obtained with a *three* dimensional subspace for each variable. Unless stated otherwise all the methods use the same oracle.

3.1. Quadratic saddle-point Problem

We consider the following quadratic saddle-point problem

$$\min_x \max_y \frac{1}{2}(x^T A_x x + y^T A_y y) + x^T C y + b_x^T x + b_y^T y, \tag{10}$$

where the matrices A_x, A_y, C are generated from the normal distribution and have pre-defined condition numbers (see Appendix C.1). We plot in Figure 1 the distance to optimum (leftmost plot), the norm of the gradient (second plot from left), the effect of the subspace dimension on the convergence of the proposed method (third from left), and the impact of the condition number κ of the block-matrices on the mean convergence rate (rightmost plot).

In the first experiment presented in Figure 1 *top row*, we consider a separable problem with $A_x \succ 0$, $A_y \prec 0$ and $C = 0$. We show that the proposed approach keeps its manifold expansion property throughout the last search direction ($m = n \geq 3$), and therefore converges extremely fast to the solution. In contrast, the convergence of other methods is very slow. This is due to the difficulty of the gradient method to converge in ill-conditioned scenarios, and due to the unified step size for the primal and the dual directions. In the second experiment (*middle row*), we consider the stable quadratic saddle point problem with $A_x \succ 0$, $A_y \prec 0$ and C to be full rank matrices. We observe the superiority of the proposed method, while the advantage of using more directions is clear in handling the interaction matrix C, as compared to gradient based methods. In the last experiment (*bottom row*), we deploy the bi-linear game problem, where $A_x = 0$, $A_y = 0$ and C is a full rank matrix. The proposed method performs significantly better, as compared to other first-order approaches. In the bi-linear case, we can see that increasing the size of the subspace is not necessarily beneficial. We show the superiority of the method in term of computational time for the different settings in Table 3.1.

3.2. Constrained Optimization: ADMM

In this experiment, we consider the smooth Lasso problem that can be reformulated (see Appendix C.2) as a saddle-point problem of the augmented Lagrangian

$$\min_{x,w} \max_y \left\{ \frac{1}{2}\|Ax - b\|^2 + \lambda \sum_j \varphi_s(w_j) + y^T(x - w) + \frac{\rho}{2}\|x - w\|^2 \right\}, \tag{11}$$

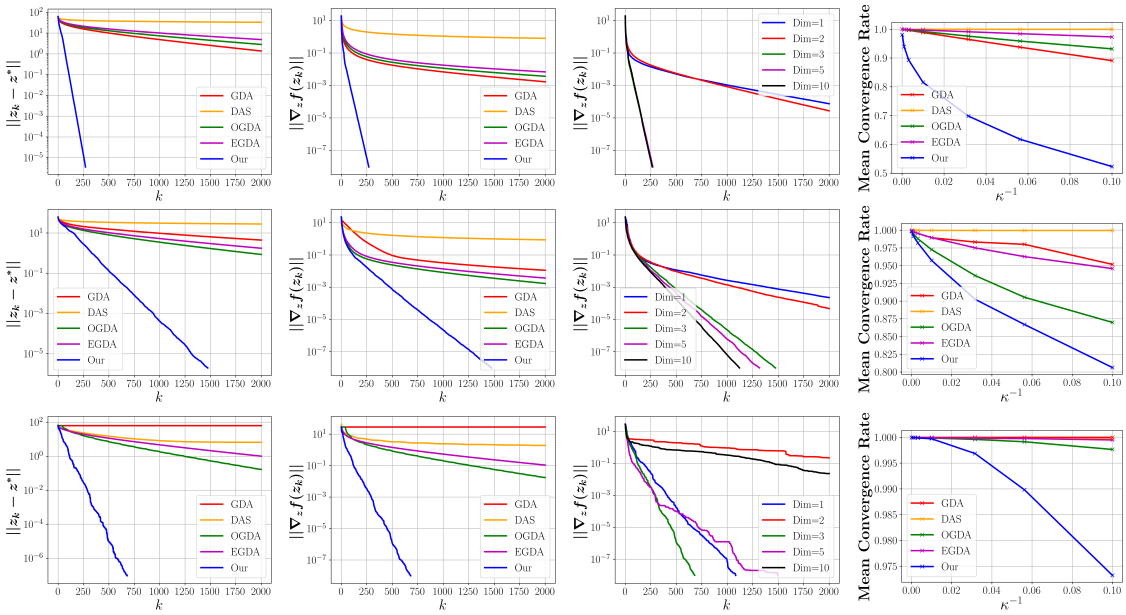

Figure 1: Quadratic saddle point problems

Setting	GDA	DAS	OGDA	EGDA	CP	**Our**
Separable	42.5	-	73.1	60.7	15.9	**5.9**
Stable	144.1	-	49.3	55.4	-	**38.1**
Unstable	∞	-	34.3	33.6	-	**14.7**

Table 1: Mean computation time in seconds of the presented methods until convergence threshold is reached, for the different quadratic settings. ∞ denotes non-convergence and '-' denotes slower convergence than our method by at least factor 30.

with $\lambda \in \mathbb{R}^+$, ρ denotes the penalty parameter and $\varphi_s(t)$ denotes the scalar smooth approximation of the L_1 norm [13]. In Figure 2 we present the convergence results of the ADMM method with smoothing constant $s = 10^{-3}$, versus the proposed subspace method boosted by the ADMM directions populating the subspace matrices. The boosting obtained by the proposed approach is significant both in speed and accuracy.

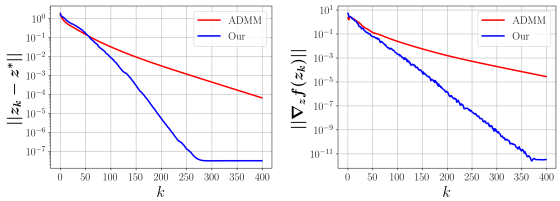

Figure 2: Subspace Optimization boosting via ADMM Directions

3.3. Generative Adversarial Networks

We test the proposed method in deterministic setting of the Dirac GAN scheme proposed in [19] by expanding the dimensions of the problem such that

$$f(x, y) = \phi(-x^T y) + \phi(y^T c), \tag{12}$$

for some scalar function ϕ (see Appendix C.3). Since the objective is concave-concave, the competing methods fail to converge to saddle-point, and diverge to saturation regions [19]. The proximal operator prevents the method from diverging, and allows faster convergence to optimum as depicted in the rightmost plot.

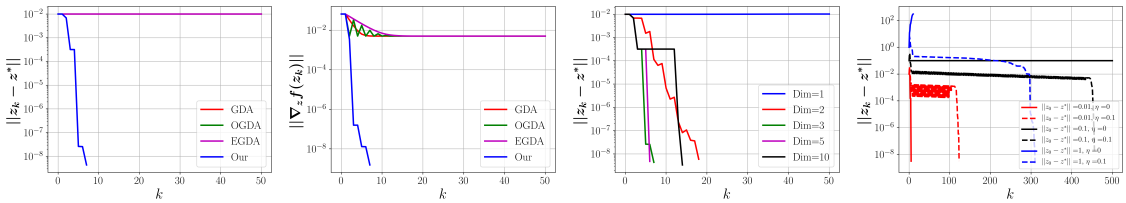

Figure 3: Dirac GAN. Other methods similarly converge to saturation region (leftmost).

4. Conclusions

In this paper we introduced a sequential subspace optimization approach to saddle-point problems. We improve convergence of first-order methods via efficient secondary subspace optimization. We evaluated the proposed framework on several saddle-point problems, demonstrating its efficacy and superior performance relative to popular optimization techniques. Further theoretical investigation of the influence of the subspace directions and dimensions may provide better understanding and enable development of both faster and more efficient saddle-point optimization methods.

References

[1] Kenneth J Arrow, Leonid Hurwicz, and Hirofumi Uzawa. Studies in linear and non-linear programming. 1958.

[2] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton University Press, 2009.

[3] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[4] Ronald E Bruck Jr. On the weak convergence of an ergodic iteration for the solution of variational inequalities for monotone operators in hilbert space. *Journal of Mathematical Analysis and Applications*, 61(1):159–164, 1977.

[5] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006.

[6] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

[7] Tony F Chan, Selim Esedoglu, and Mila Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM journal on applied mathematics*, 66(5): 1632–1648, 2006.

[8] AR Conn, Nick Gould, A Sartenaer, and Ph L Toint. On iterated-subspace minimization methods for nonlinear optimization. *Linear and Nonlinear Conjugate Gradient-Related Methods*, pages 50–78, 1996.

[9] EE Cragg and AV Levy. Study on a supermemory gradient method for the minimization of functions. *Journal of Optimization Theory and Applications*, 4(3):191–205, 1969.

[10] Constantinos Daskalakis and Ioannis Panageas. The limit points of (optimistic) gradient descent in min-max optimization. In *Advances in Neural Information Processing Systems*, pages 9236–9246, 2018.

[11] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. *arXiv preprint arXiv:1711.00141*, 2017.

[12] John E Dennis Jr and Kathryn Turner. Generalized conjugate directions. *Linear Algebra and its Applications*, 88:187–209, 1987.

[13] Michael Elad, Boaz Matalon, and Michael Zibulevsky. Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization. *Applied and Computational Harmonic Analysis*, 23(3):346–367, 2007.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[15] Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS Washington, DC, 1952.

[16] Kevin Leyton-Brown and Yoav Shoham. Essentials of game theory: A concise multidisciplinary introduction. *Synthesis lectures on artificial intelligence and machine learning*, 2(1): 1–88, 2008.

[17] Bernard Martinet. Brève communication. régularisation d'inéquations variationnelles par approximations successives. *Revue française d'informatique et de recherche opérationnelle. Série rouge*, 4(R3):154–158, 1970.

[18] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. In *Advances in Neural Information Processing Systems*, pages 1825–1835, 2017.

[19] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? *arXiv preprint arXiv:1801.04406*, 2018.

[20] A Miele and JW Cantrell. Study on a memory gradient method for the minimization of functions. *Journal of Optimization Theory and Applications*, 3(6):459–470, 1969.

[21] Guy Narkiss and Michael Zibulevsky. Sequential subspace optimization method for large-scale unconstrained problems. Technical Report CCIT 559, Technion – Israel Institute of Technology, Faculty of Electrical Engineering, 2005.

[22] Arkadi Nemirovski. Orth-method for smooth convex optimization. *Izvestia AN SSSR, Transl.: Eng. Cybern. Soviet J. Comput. Syst. Sci*, 2:937–947, 1982.

[23] Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.

[24] Jorge Nocedal and S Wright. *Numerical optimization*. Series in operations research and financial engineering, Springer, New York,, 2006.

[25] Jorge Nocedal and S Wright. *Numerical Optimization - Line Search Methods*, pages 30–65. Springer New York, New York, NY, 2006. ISBN 978-0-387-40065-5. doi: 10.1007/978-0-387-40065-5_3. URL https://doi.org/10.1007/978-0-387-40065-5_3.

[26] Barak A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6:147–160, 1994.

[27] Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013.

[28] R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

[29] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[30] Danny C Sorensen. Newton's method with a model trust region modification. *SIAM Journal on Numerical Analysis*, 19(2):409–426, 1982.

[31] Abhay Yadav, Sohil Shah, Zheng Xu, David Jacobs, and Tom Goldstein. Stabilizing adversarial nets with prediction methods. *arXiv preprint arXiv:1705.07364*, 2017.

Appendix A. Proof of Theorem 3.1

Let us first consider the bi-linear setting $f(x,y) = x^T C y$, where C is a full rank-matrix. We first show that the gradient method is diverging in the above case. The gradient method update is defined as

$$\begin{aligned}\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} &= \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \eta_k \begin{pmatrix} -\nabla_x f(x_k, y_k) \\ \nabla_y f(x_k, y_k) \end{pmatrix} \\ &= \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \eta_k \begin{pmatrix} 0 & -C \\ C^T & 0 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} \\ &= \begin{pmatrix} I & -\eta_k C \\ \eta_k C^T & I \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} = A \begin{pmatrix} x_k \\ y_k \end{pmatrix}.\end{aligned} \tag{13}$$

Thus we have $\forall C$ and $\forall \eta_k$

$$\left\| \begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} \right\|^2 \geq \lambda_{\min}(A^T A) \left\| \begin{pmatrix} x_k \\ y_k \end{pmatrix} \right\|^2 = \left(1 + \lambda_{\min}(\eta_k^2 C C^T)\right) \left\| \begin{pmatrix} x_k \\ y_k \end{pmatrix} \right\|^2. \tag{14}$$

We now proceed to the proof of convergence of the one dimensional subspace method, assuming $C = I$. Optimal solution of the subspace optimization satisfies

$$\begin{aligned}&\begin{cases} P_k^T \nabla_x f(x_k + P_k\alpha, y_k + Q_k\beta) = 0 \\ Q_k^T \nabla_y f(x_k + P_k\alpha, y_k + Q_k\beta) = 0 \end{cases} \\ \Leftrightarrow &\begin{cases} \alpha = -(Q_k^T C^T P_k)^{-1} Q_k^T C^T x_k \\ \beta = -(P_k^T C Q_k)^{-1} P_k^T C y_k \end{cases}\end{aligned} \tag{15}$$

Thereafter, the update of the variable x is written as

$$x_{k+1} = x_k - \eta_k P_k (Q_k^T C^T P_k)^{-1} Q_k^T C^T x_k = x_k - \eta_k P_k (Q_k^T C^T P_k)^{-1} Q_k^T Q_k, \tag{16}$$

with $\eta_k > 0$. Then, we can show that

$$\begin{aligned}\|x_{k+1}\|^2 &= \|x_k\|^2 - 2\frac{\eta_k}{Q_k^T C^T P_k} \langle x_k, P_k \|Q_k\|^2 \rangle + \frac{\eta_k^2 \|P_k\|^2 \|Q_k\|^4}{(Q_k^T C^T P_k)^2} \\ &= \|x_k\|^2 + \frac{\|Q_k\|^2}{(Q_k^T C^T P_k)^2} \Big(- 2\eta_k Q_k^T C^T P_k \langle x_k, P_k \rangle + \eta_k^2 \|P_k\|^2 \|Q_k\|^2 \Big).\end{aligned} \tag{17}$$

Denoting $\delta_k = -2\eta_k Q_k^T C^T P_k \langle x, P_k \rangle + \eta_k^2 \|P_k\|^2 \|Q_k\|^2$
and since $C = I$ we have

$$\begin{aligned}\delta_k &= -2\eta_k \|Q_k^T P_k\|^2 + \eta_k^2 \|P_k\|^2 \|Q_k\|^2 \\ &= -2\eta_k f(x_k, y_k)^2 + \eta_k^2 \|\nabla_x f(x_k, y_k)\|^2 \|\nabla_y f(x_k, y_k)\|^2\end{aligned} \tag{18}$$

Thus, $\forall \eta_k \in (0, 2f(x_k, y_k)^2 / \|\nabla_x f(x_k, y_k)\|^2 \|\nabla_y f(x_k, y_k)\|^2)$ we have $\|x_{k+1}\|^2 < \|x_k\|^2$. By following similar arguments, we get $\|y_{k+1}\|^2 < \|y_k\|^2$.
□

Appendix B. Proof of Theorem 3.2

We are looking for $\eta > 0$ such that $z_{k+1} = z_k + \eta d_k$ is a better stationary point than z_k, i.e. $\|\nabla f(z_{k+1})\| \leq \|\nabla f(z_k)\|$. Here d_k designate the anti-gradient d_x and gradient direction d_y according to the primal and dual variable respectively. From first order expansion we have

$$\begin{aligned} f(z_{k+1}) &= f(z_k) + \eta\langle \nabla f(z_k), d_k\rangle + o(\eta^2\|d_k\|) \\ \iff \nabla f(z_{k+1}) &= \nabla f(z_k) + \eta\nabla^2 f(z_k) d_k \end{aligned} \tag{19}$$

Thus we have

$$\begin{aligned} \|\nabla f(z_{k+1})\|^2 &= \|\nabla f(z_k)\|^2 + 2\eta\nabla f(z_k)^T\nabla^2 f(z_k)d_k + \eta^2 d_k^T\nabla^2 f(z_k)^T\nabla^2 f(z_k)d_k \\ \underset{\eta\to 0}{\Rightarrow} \|\nabla f(z_{k+1})\|^2 &= \|\nabla f(z_k)\|^2 + 2\eta\nabla f(z_k)^T\nabla^2 f(z_k)d_k, \end{aligned} \tag{20}$$

Since we have

$$\begin{aligned} \nabla f(z_k)^T\nabla^2 f(z_k)d_k &= -d_x^T\nabla_{xx} f(z_k)d_x + d_y^T\nabla_{yy} f(z_k)d_y - d_x^T\nabla_{xy} f(z_k)d_y + d_y^T\nabla_{yx} f(z_k)d_x \\ &= -d_x^T\nabla_{xx} f(z_k)d_x + d_y^T\nabla_{yy} f(z_k)d_y < 0, \end{aligned} \tag{21}$$

where the last inequality arises from the positive/negative definiteness of the second order partial derivatives in $B_2((x^*, y^*), r)$. □

Notice the line search procedure cannot diverge for non-strongly convex-concave problems where the block diagonal Hessian can vanish.

Appendix C. Experiments Settings

GDA refers to the gradient method [28], DAS refers to the dual averaging scheme of [23], OGDA is the optimistic gradient method [11], and EGDA refers to the extrinsic gradient method [31]. All the methods but DAS are implemented using the proposed backtracking line search for improved convergence. When line search did not converge for EGDA, we searched for optimal step size. In the following, the proximal centers of the proposed method are set to previous point (i.e. x_{k-1} and y_{k-1}). In all the presented experiments, the subspace is populated by the current gradient ($m = n = 1$), by previous gradient ($m = n = 2$), and by previous search directions ($m = n \geq 3$). The machine precision ϵ is set to single precision 10^{-8} and the maximum number of inner optimization iterations is limited to 10. In all the figures k denote the iteration number. Unless stated otherwise all the methods use the same oracle.

C.1. Quadratic saddle-point Problem

We recall the quadratic saddle-point problem

$$\min_x \max_y \frac{1}{2}(x^T A_x x + y^T A_y y) + x^T C y + b_x^T x + b_y^T y, \tag{22}$$

where the matrices A_x, A_y, C are generated from the normal distribution and have pre-defined condition numbers. Namely, we generate a standard Gaussian matrix with i.i.d. entries, perform its SVD and substitute diagonal singular values with an array of log-uniform random values in predefined range. The dimension of the optimization problem is set to $M = 1500, N = 500$. We plot

the distance to optimum (leftmost plot) and the norm of the gradient (second plot from left). Also, we show the impact of the condition number of the block-matrices in A on the mean convergence rate $K^{-1}\sum_k^K \|z_{k+1}-z^*\|/\|z_k-z^*\|$ (rightmost plot). Here x-axis represents the inverse condition number κ^{-1}. These first three plots are obtained with a *three* dimensional subspace for each variable. Finally, we show the effect of the subspace dimension on the convergence of the proposed method (third from left). For this experiment we add time comparison with the Chambolle-Pock algorithm (CP) with step size set according to the interaction term matrix to ensure proven convergence [6]. For fairness, we do not assume any closed form solution is given, all the derivatives are computed at each iteration using the same automatic differentiation tool for all the methods.

In the first experiment presented, we consider a separable problem with $A_x \succ 0$, $A_y \prec 0$ and $C = 0$. The two matrices are conditioned with the condition numbers $\kappa(A_x) = 10^3, \kappa(A_y) = 10^2$.

In the second experiment, we consider the stable quadratic saddle point problem with $A_x \succ 0$, $A_y \prec 0$ and C to be full rank matrices. Here, all the block-matrices are conditioned with condition number $\kappa(A_x) = 10^3, \kappa(A_y) = 10^2, \kappa(C) = 10^3$.

In the last experiment, we deploy the bi-linear game problem, where $A_x = 0$, $A_y = 0$ and C is a full rank matrix, such that $\kappa(C) = 10^2$. Here $M = N = 1000$ so there exist only one solution to the equivalent system of linear equations.

C.2. Constrained Optimization: ADMM

We recall the original problem

$$\min_x \quad \frac{1}{2}\|Ax-b\|^2 + \lambda \sum_j \varphi_s(x_j), \tag{23}$$

with $\lambda \in \mathbb{R}^+$. Here x_j denotes the j^{th} component of vector x, and $\varphi_s(t)$ denotes the scalar non-linearity that implements the smooth convex approximation $\sum_j \varphi_s(x_j)$ of the ℓ_1 norm such that $\varphi_s(t) = |t| - s\ln(1+|t|/s), s \in (0,\infty)$, where the scaling factor s defines the degree of smoothness. This choice of $\varphi_s(t)$ yields well defined shrinkage [13]. The original ADMM algorithm can be summarized in the following three steps: minimization in former primal variable x, minimization in separable variable w, and update of the dual variable y [3]. We can reformulate the Lasso setting as a saddle-point problem of the augmented Lagrangian

$$\min_{x,w}\max_y \left\{\frac{1}{2}\|Ax-b\|^2 + \lambda\sum_j \varphi_s(w_j) + y^T(x-w) + \frac{\rho}{2}\|x-w\|^2\right\} \tag{24}$$

where ρ denotes the penalty parameter. In Figure 2 we present the convergence results of the ADMM method with smoothing constant $s = 10^{-3}$, versus the proposed subspace method boosted by the ADMM directions populating the subspace matrices. The data setting is the same as in [3], Section (11.1).

C.3. Generative Adversarial Networks

Generative Adversarial Networks became recently one of the most popular applications of the minimax approach [14]. We test the proposed method in deterministic setting of the Dirac GAN scheme proposed in [19] by expanding the dimensions of the problem such that

$$f(x,y) = \phi(-x^T y) + \phi(y^T c), \tag{25}$$

for some scalar function ϕ. Here, c denotes the high dimensional Dirac distribution value that we sample from the Normal distribution. The primal and dual variables represent the data generator and discriminator, respectively. Figure 3 depicts the distance to optimum (leftmost), the gradient norm of the generator and discriminator (second from left), the influence of the subspace dimension (third from left), and the influence of the proximal factor on convergence for different initialization (rightmost). Therein, we use the common sigmoid cross-entropy loss $\phi(t) = -\ln(1 + e^{-t})$ [14].