Solution Structure Utilization for Efficient Optimization and Large-scale Machine Learning

LEE Ching-pei

Optimization in the Big Data Era

Fast Optimization Algorithms in the Big Data Era

Dec 8, 2022

Regularized Optimization Problem

Consider the following regularized optimization problem:

$$\min_{x \in \mathbb{R}^n} F(x) \coloneqq f(x) + \Psi(x), \tag{REG}$$

- Assume solution set $\Omega \neq \emptyset$
- Loss function f: differentiable (other assumptions depend on algorithms)
- Regularizer Ψ: convex and lower semicontinuous
- Regularizers for inducing structures in the solution, like
  - (Group) sparsity: feature selection, model compression, faster prediction
  - Low rank (for matrix/tensor): robustness, denoising, ground truth recoverability

Structure-inducing in the Exact Solution

- Viewing from the first-order optimality condition, such regularizers induce desired structures exactly at stationary points
- But iterative algorithms we use can only generate approximate solutions to an optimization problem
- Convergence guarantees we get are often that the limit points of the iterates are stationary
- There is usually no guarantee for the approximate solutions to possess the same structure as the limit points

Fast Convergence Does not Imply Structure

Example 1

$$\min_{x \in \mathbb{R}^2} (x_1 - 2.5)^2 + (x_2 - 0.3)^2 + \|x\|_1,$$

- $\Psi(\cdot) = \|\cdot\|_1$, the only solution is $x^* = (2, 0)$: with sparsity
- Consider $\{x^t\}$ with $x_1^t = 2 + f(t), x_2^t = f(t)$, for some $f(t) > 0$ with $f(t) \downarrow 0$
- f is arbitrary, both the iterates and the corresponding objective values converge to the optimum arbitrarily fast, but no x^t has the same sparsity pattern as x^*

Structure, Manifold, and Partial Smoothness

- Points sharing the same structure as a stationary point (sparse pattern, matrices of the same rank, ...) can often be locally represented by a smooth manifold
- Promotion of a structure is often achieved through a special kind of nonsmoothness, called partial smoothness, of the regularizer
- A function is partly smooth at a point x^* if it is locally smooth in a neighborhood within an active manifold containing x^*, but nonsmooth along directions leaving the manifold (normal space)
- The active manifold is locally the optimal structure (the one with the lowest possible dimension) we can get for any sequence converging to the same limit point
- Select regularizer for its corresponding active manifold

Manifold Identification

- If we are able to find/identify such an optimal manifold, even with approximate solutions, we can get the desired target structure
- By identification, we mean for an algorithm to generate iterates that eventually stay within this optimal manifold
- We will utilize tools from manifold identification (pioneered by Wright (1993), later further extended by Lewis (2002); Hare and Lewis (2004)) to design suitable algorithms

Solution Structure Utilization

- Those optimal manifolds are usually of a dimensionality much lower than the original dimension of the optimization problem
- Could we utilize the solution structure, namely the optimal manifold, to devise algorithms that
  1. are guaranteed to find the optimal structure,
  2. converge fast, and
  3. require lower per-iteration cost?
- The original goal of adding a regularizer is finding the right structure, and thus just discussing convergence guarantees or rates for regularized problems without considering the structure seems (to me) insufficient
- Although in many cases of machine learning, for generalization error we don't need a highly precise solution, it might be a different story for finding the structure

Outline

Partial Smoothness

Definition 2 (Partly smooth (Hare and Lewis, 2004))

A convex function Ψ is partly smooth at x^* relative to a set $\mathcal{M}_{x^*} \ni x^*$ if $\partial\Psi(x^*) \neq \emptyset$ and:

1. Around x^*, $\mathcal{M}_{x^*}$ is a $\mathcal{C}^2$-manifold and $\Psi|_{\mathcal{M}_{x^*}}$ is $\mathcal{C}^2$.
2. The affine span of $\partial\Psi(x^*)$ is a translate of the normal space to $\mathcal{M}_{x^*}$ at x^*.
3. $\partial\Psi$ is continuous at x^* relative to $\mathcal{M}_{x^*}$.

Roughly speaking: function value changes

- smoothly along the active manifold $\mathcal{M}_{x^*}$ around x^*,
- but sharply along directions leaving to the manifold

Partly Smooth Regularizers

Some popular regularizers we will use in this talk:

- ℓ_1-norm, ℓ_0-norm: $\mathcal{M}_{x^*} = \{y \mid y_i = 0, \forall i : x_i^* = 0\}$
- Group-LASSO norm: $\mathcal{M}_{x^*} = \{y \mid y_I = 0, \forall I : x_I^* = 0\}$, I are blocks/groups
- Nuclear norm: $\mathcal{M}_{X^*} = \{Y \mid \mathrm{rank}(Y) = \mathrm{rank}(X)\}$

Identification

Lemma 3 (L. (2020), extended from Lewis and Zhang (2013))

Assume $f \in \mathcal{C}^1$ and Ψ convex and partly smooth at a point x^* relative to $\mathcal{M}_{x^*}$. If

1. *The nondegenerate condition holds*

$$0 \in \operatorname{relint}(\partial F(x^*)) = \nabla f(x^*) + \operatorname{relint}(\partial \Psi(x^*)) \qquad \text{(NOD)}$$

2. $x^t \to x^*$, *and*
3. $F(x^t) \to F(x^*)$,

then

$$\operatorname{dist}\left(0, \partial F(x^t)\right) \to 0 \quad \Leftrightarrow \quad x^t \in \mathcal{M}_{x^*} \text{ for all t large.}$$

Outline

Motivating Example: Structured Sparsity

- Structured neural networks: condense gigantic deep learning models, for smaller memory footprint, deploying on mobile devices, faster prediction
- Especially important because overparameterization is widely used for easier training
- Main approach: sparsify to get fewer model variables to store, and hopefully fewer computation
- Want to trim out neurons (in fully-connected layers) or a whole convolutional kernel, but not just individual weights, to really reduce model size and accelerate prediction (GPUs are fast only when handling dense matrices/tensors)
- Such patterned sparsity is called structured sparsity

Variance Reduction

- Variance reduction can help to achieve the zero minimum-norm subgradient condition
- But variance reduction that utilizes the finite-sum structure of ERM does not work for deep learning (Defazio and Bottou, 2019) because of data augmentation, which transform the objective to the expectation of a loss function over distributions
- Most algorithms with variance reduction in infinite-sum settings require more computation (for better convergence rates) – not ideal for time-consuming deep learning problems
- Our proposal (Huang and L., 2022): regularized modernized dual averaging (RMDA), inspired by RDA (Xiao, 2010; Lee and Wright, 2012) and MDA (Jelassi and Defazio, 2020)

Problem Formulation and Algorithm Idea

Consider the following regularized optimization problem:

$$\min_{x} \quad F(x) \coloneqq \mathbb{E}_{\xi \sim \mathcal{D}} \left[f_{\xi}(x) \right] + \psi(x)$$

- $\mathcal{D}$ is a distribution
- f_{ξ} is differentiable with Lipschitz gradient almost everywhere for all $\xi \in \Omega$
- $f \coloneqq \mathbb{E}_{\xi \sim \mathcal{D}} \left[f_{\xi}(x) \right]$ is the expected loss (over all possible data augmentations)

Algorithm

Algorithm 1: RMDA $(x^0, T, \eta(\cdot), c(\cdot))$

input : Initial point x^0, step size schedule $\eta(\cdot)$, momentum schedule function $c(\cdot)$, number of epochs T

$V_0 \leftarrow 0, \quad \alpha_0 \leftarrow 0$

for $t = 1, \ldots, T$ **do**

$\beta_t \leftarrow \sqrt{t}, \quad s_t \leftarrow \eta(t)\beta_t, \quad \alpha_t \leftarrow \alpha_{t-1} + s_t$

Sample $\xi_t \sim \mathcal{D}$ and compute $G^t \leftarrow \nabla f_{\xi_t}(x^{t-1})$

$V^t \leftarrow V^{t-1} + s_t G^t$ `// dual (weighted) averaging`

$\tilde{x}^t \leftarrow \operatorname{argmin}_x \langle \frac{V^t}{\alpha_t}, x \rangle + \frac{\beta_t}{2\alpha_t} \|x - x^0\|^2 + \psi(x)$ `// proximal operation`

$x^t \leftarrow (1 - c(t))x^{t-1} + c(t)\tilde{x}^t$ `// momentum`

output : The final model x^T

- Multi-stage step sizes; restart V_t and α_t whenever the stepsize changes
- increase $c(t)$ by the same proportion we decrease $\eta(t)$, but capped by $c(t) = 1$: following the empirical observation of Jelassi and Defazio (2020)

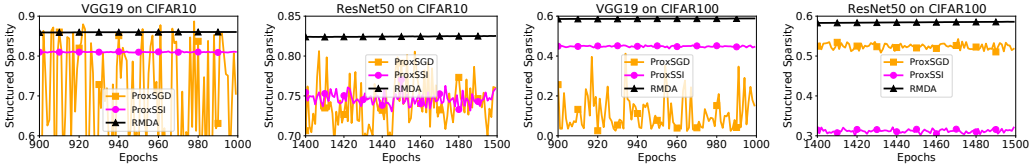
Theoretical Guarantees

Summary of theoretical results (details omitted, not the point here):

- (Variance reduction) As long as the iterates eventually move slow enough, the averaged stochastic gradient $\alpha_t^{-1} V_t$ converges to the real gradient $\nabla f(x^{t-1})$ almost surely
- (Stationarity) If the iterates converge to a point, this point is stationary almost surely
- (Manifold identification) When the first two items and (NOD) hold, the optimal manifold at x^* is identified within finite steps almost surely

Experiment of Structured Sparsity using Group-LASSO Norm

- ProxSGD (Yang et al., 2019): proximal SG+momentum
- ProxSSI (Deleu and Bengio, 2021): AdamW + proximal operations



Algorithm	Test accuracy	Structured sparsity	Test accuracy	Structured sparsity
	VGG19/CIFAR10		VGG19/CIFAR100	
ProxSGD	92.4 ± 0.3%	72.6 ± 6.0%	71.9 ± 0.1%	8.6 ± 4.9%
ProxSSI	92.5 ± 0.0%	81.1 ± 0.2%	66.2 ± 0.4%	46.4 ± 1.4%
RMDA	93.6 ± 0.2%	86.4 ± 0.3%	72.2 ± 0.2%	58.9 ± 0.4%
	ResNet50/CIFAR10		ResNet50/CIFAR100	
ProxSGD	92.4 ± 0.1%	76.8 ± 4.1%	75.5 ± 0.5%	51.8 ± 0.3%
ProxSSI	94.1 ± 0.1%	74.8 ± 1.3%	74.5 ± 0.3%	32.8 ± 2.5%
RMDA	94.3 ± 0.0%	83.0 ± 0.5%	76.1 ± 0.5%	57.7 ± 3.8%

Might take much more iterations to reach the ideal structure than to reach the ideal prediction accuracy (structured sparsity of RMDA on ResNet50 was still steadily increasing near the final epochs, so it could get even better with more iterations)

Outline

Regularized Optimization Problem

Recall (REG)

$$\min_x F(x) \coloneqq f(x) + \Psi(x), \tag{REG}$$

- ∇f Lipschitz continuous, $\Psi : \mathbb{R}^n \to \mathbb{R}$: convex and partly smooth
- F is lower-bounded and the solution set Ω is non-empty

Inexact Successive Quadratic Approximation (ISQA)

At the tth iteration, with iterate x^t, ISQA finds an update direction p^t by solving

$$p^t \approx \operatorname*{argmin}_{p\in\mathbb{R}^n} \quad Q_{H_t}^{x^t}\left(p; x^t\right) := \nabla f\left(x^t\right)^\top d + \frac{1}{2} d^\top H_t d + \Psi\left(x^t + d\right) - \Psi\left(x^t\right) \quad \text{(SUBPROB)}$$

for some symmetric and positive-semidefinite H_t.

- A stepsize α_t along p^t is then decided for updating the iterate
- Many existing algorithms included in this framework: proximal Newton (PN) when $H_t = \nabla^2 f(x^t)$, proximal quasi-Newton (PQN), proximal gradient, and so on
- Subproblem has no closed-form solution when H_t is not diagonal: apply an iterative solver to obtain an approximate solution
- abbreviation: $Q_t(p) := Q_{H_t}^{x^t}(p; x^t)$

Solution Inexactness

- For PN and PQN, under suitable conditions, superlinear convergence in the number of times updating x^t can be obtained
- Similar to the smooth case (i.e. $\Psi \equiv 0$): requires increasing solution accuracy of (SUBPROB)
- Unlike the smooth case: no closed-form or finite-termination solver (direct inverse/matrix factorization/conjugate gradient) exists for (SUBPROB)
- Superlinear convergence only in theory and in outer iterations, but not observed in real running time

Possible Remedy and Further Difficulties

- If Ψ is partly smooth around a point x^*, and the iterates converge to x^*, then after identifying the optimal manifold $\mathcal{M}_{x^*} \ni x^*$, we can switch to smooth optimization
- Low per-iteration cost from the low dimensionality of the manifold
- Finite termination in subproblem solving from smoothness
- If (SUBPROB) is always solved exactly and $\alpha_t \equiv 1$, it is known that the optimal manifold can be identified, as long as the iterates converge to x^* (Hare, 2011)
- But due to the inexactness in the approximate subproblem solution, iterates of ISQA without further conditions in general do not identify the optimal manifold (Example 1)
- Numerical experience is the opposite: ISQA can identify the active manifold in practice
- Analyze this and propose acceleration in L. (2020)

Algorithm Details

- Choice of H_t: bounded and positive-definite

$$\exists M, m > 0, \quad \text{such that} \quad M \succeq H_t \succeq m, \ \forall t \geq 0. \tag{BD+PD}$$

- Inexact solution: need $Q_t(p^t) < Q_t(0) = 0$, and consider (choice of ϵ_t in next page)

$$Q_t(p^t) - \min_p Q_t(p) \leq \epsilon_t \tag{OBJ}$$

- Step size: given $\gamma \in (0, 1)$ find α_t such that the objective decrease is sufficiently large

$$F(x^t + \alpha_t p^t) \leq F(x^t) + \alpha_t \gamma Q_t(p^t) \tag{Armijo}$$

Algorithm 2: Framework of ISQA

input : x^0, $\gamma, \beta \in (0, 1)$
for $t = 0, 1, \ldots$ **do**
  $\alpha_t \leftarrow 1$, pick $\epsilon_t \geq 0$ and H_t, and solve (SUBPROB) for p^t satisfying (OBJ)
  while (Armijo) *not satisfied* **do** $\alpha_t \leftarrow \beta \alpha_t$
  $x^{t+1} \leftarrow x^t + \alpha_t p^t$

Identification from Subproblem Solver I

- p^{t*} denotes the optimal solution to (SUBPROB) and $Q_t^* := Q_t(p^{t*})$
- Consider relative precision in (OBJ) for easier analysis:

$$\begin{aligned} &\exists \eta \in [0,1): \quad \epsilon_t = \eta\left(Q_t(0) - Q_t^*\right) = -\eta Q_t^*, \quad \forall t \\ &\Rightarrow Q_t(p^t) \leq (1-\eta) Q_t^*. \end{aligned} \tag{Relative}$$

- Existence of η easily satisfied by applying a Q-linear-convergent solver to (SUBPROB) for a fixed number of iterations due to (BD+PD) (enforcing a certain η needs explicit knowledge of upper and lower bounds of H_t)
- Define the generalized proximal mapping: for any function g, $\tau \geq 0$, and Λ PD,

$$\operatorname{prox}_{\tau g}^{\Lambda}(x) := \operatorname*{argmin}_y \frac{1}{2}\langle x-y,\, \Lambda(x-y)\rangle + \tau g\,(y)$$

Identification from Subproblem Solver II

Theorem 4 (L. (2020))

Assume (Relative) *for* $\eta \in [0, 1)$ *and* (NOD) *holds. If the update direction* p^t *satisfies*

$$x^t + p^t = \operatorname{prox}_{\Psi}^{\Lambda_t}\left(y^t - \Lambda_t^{-1}\left(\nabla f\left(x^t\right) + H_t\left(y^t - x^t\right) + s^t\right)\right), \tag{Prox}$$

with

- Λ_t *symmetric and PD,* $M_1 \geq \|\Lambda_t\|$ *for* $M_1 > 0$,
- $\|y^t - (x^t + p^{t*})\|$ *decreases to* 0 *with* $|Q_t^*|$, *and*
- $\|s^t\|$ *decreases to* 0 *with* $\|y^t - (x^t + p^{t*})\|$

then there are $\epsilon, \delta > 0$ *such that:*

1. $\|x_t - x^*\| \leq \delta$,
2. $|Q_t^*| \leq \epsilon$, *and*
3. $\alpha_t = 1$

imply $x^{t+1} \in \mathcal{M}_{x^*}$.

Applicable Subproblem Solvers

- $|Q_t^*| \to 0$ for ISQA under (BD+PD) and (Relative) is known (L. and Wright, 2019)
- Almost all popular subproblem solvers satisfy such conditions: including, but not limited to
  - Proximal gradient (PG)
  - Accelerated PG (APG)
  - Variance reduction methods like Prox-SAGA/SVRG
  - Proximal (cyclic) block-coordinate descent (CD)
- Almost all general-purpose solvers used in practice, so ISQA essentially achieves manifold identification

Iterate Convergence Under a Growth Condition

- We at least need a convergent subsequence of iterates to get to a neighborhood of x^*
- Solution set might be unbounded, so might not have a convergent subsequence
- First-order methods have some implicit regularization so convexity alone is sufficient for ensuring a bounded iterate sequence, but not the case for other methods like ISQA
- Assume F satisfy the following growth condition for some $\zeta, \xi > 0$, and $\theta \in (0, 1]$:

$$\zeta \mathrm{dist}(x, \Omega) \leq \left(F(x) - \min_x F(x) \right)^\theta, \ \forall x \in \left\{ x \mid F(x) - \min_x F(x) \leq \xi \right\} \quad \text{(GROWTH)}$$

- Special cases: quadratic growth ($\theta = 1/2$), weak sharp minima ($\theta = 1$)
- Cannot use the framework of Attouch et al. (2013) due to the inexactness in subproblem

Iterate Convergence Under (GROWTH)

Lemma 5 (L. (2020))

Assume f is convex, ∇f is Lipschitz continuous, and (GROWTH) holds locally. Then

1. *For $\theta \in [1/2, 1]$: $\delta_t := F(x^t) - \min_x F(x)$ converges to 0 Q-linearly.*
2. *For $\theta \in (0, 1/2)$, $\delta_t = O(t^{-1/(1-2\theta)})$*

Theorem 6 (L. (2020))

For $\theta \in (1/4, 1]$, $x^t \to x^*$ for some $x^* \in \Omega$.

Iterate Convergence Under (GROWTH)

Lemma 5 (L. (2020))

Assume f is convex, ∇f is Lipschitz continuous, and (GROWTH) holds locally. Then

1. *For $\theta \in [1/2, 1]$: $\delta_t := F(x^t) - \min_x F(x)$ converges to 0 Q-linearly.*
2. *For $\theta \in (0, 1/2)$, $\delta_t = O(t^{-1/(1-2\theta)})$*

Theorem 6 (L. (2020))

For $\theta \in (1/4, 1]$, $x^t \to x^*$ for some $x^* \in \Omega$.

Recently found that some cases already imply finding an optimum: (preprint with Steve coming soon)

Theorem 7

for $\theta > 1/2$, an optimal solution is found within finite iterations.

Manifold Identification in Practice

- Theory suggests to force $\alpha_t = 1$: enlarge H_t and re-solve when (Armijo) fails with $\alpha_t = 1$
- x^* and $\mathcal{M}_{x^*}$ unknown a priori, so cannot be certain whether $\mathcal{M}_{x^*}$ has been identified
- Solution: if x^t stays in the same manifold for sufficiently many consecutive iterations, likely we have found $\mathcal{M}_{x^*}$
- Still need safeguards: what if the current one is still wrong?
- Alternate between PG (directly on the original problem) and smooth optimization (PG is generally much cheaper than ISQA in terms of per iteration cost)

Accelerated Algorithm from Solution Structure Utilization

The proposed algorithm ISQA$^+$:

- ISQA stage:
  1. Solve (SUBPROB)
  2. If x^t stays within the same manifold for T iterations: switch to the smooth stage
- Smooth stage:
  1. One iteration of Newton within the current manifold
  2. One iteration of PG
  3. If the manifold changes after PG, or the smooth step fails to decrease the objective, go back to the ISQA stage

Superlinear convergence in outer iterations follows from that of Newton, and in running time because the subproblem can be solved to optimality in finite time

- With proper damping, just need a Hölderian error bound to get superlinear convergence (assumption of strong convexity or PD of Hessian on $\mathcal{M}_{x^*}$ not needed)

Experiment Setting

- ℓ_1-regularized logistic regression: domain $\mathbb{R}^d$,

$$\Psi(x) = \lambda\|x\|_1,\ f(x) = \sum_{i=1}^{n} \log\left(1 + \exp\left(-b_i\langle a_i,\, x\rangle\right)\right),$$

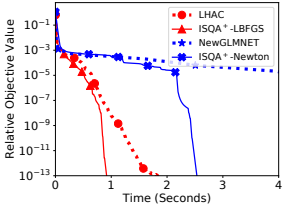
  ($\lambda = 1$ in the experiments)

- Algorithms to compare:
  - LHAC (Scheinberg and Tang, 2016): an inexact proximal quasi-Newton (L-BFGS) method with CD for the subproblem
  - NewGLMNET (Yuan et al., 2012): a line-search Proximal Newton method with a CD subproblem solver
  - ISQA$^+$-LBFGS and ISQA$^+$-Newton: our algorithm with the first stage using L-BFGS (similar to LHAC) and real Hessian (similar to NewGLMNET) for H_t, respectively

$$\text{Relative Objective Value} = \frac{\text{current objective value} - \text{optimal objective value}}{\text{optimal objective value}}.$$

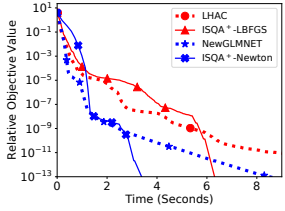
a9a

$n = 32,561, d = 123$



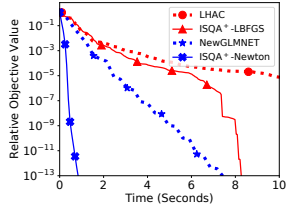
realsim

$n = 72,309, d = 20,958$



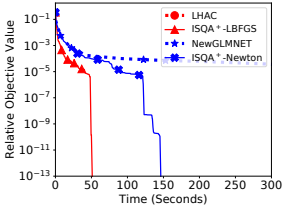
news20

$n = 19,996, d = 1,355,191$



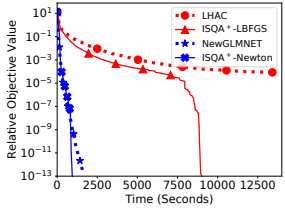
covtype.scale (LHAC failed)

$n = 581,012, d = 54$

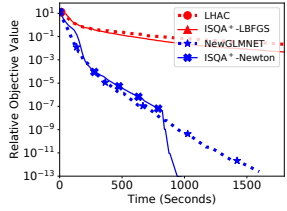


webspam

$n = 350,000, d = 16,609,143$



webspam (finer scale)



Outline

Nuclear-norm Regularized Optimization

- Consider

$$\min_{X \in \mathbb{R}^{m \times n}} \quad F(X) \coloneqq f(X) + \lambda \|X\|_*, \qquad \text{(Matrix-Completion)}$$

  with f convex, ∇f L-Lipschitz continuous, and $\lambda > 0$

- Properties of the nuclear norm (Lewis and Overton, 1996):
  - ℓ_1 norm applied to the singular values
  - Promotes sparsity in the singular values: leading to low-rank solutions
  - Proximal operation can be conducted in closed-form
- The problem itself looks very simple: convex regularized optimization, a smooth term and an "easy-to-compute" term with a closed-form proximal operator
- Difficulties:
  - m and n can be really large in modern applications: may be unable to explicitly compute and store the whole X in the memory, so exact SVD is not that easy
  - SVD is also very expensive in this case

Matrix Factorization

- Another way to compute the nuclear norm(Rennie and Srebro, 2005):

$$2\|X\|_* = \min_{W,H:WH^\top=X} \|W\|_F^2 + \|H\|_F^2 \qquad \text{(Nuclear-Frobenius)}$$

- A natural idea to solve (Matrix-Completion) is then to explicitly write down the decomposition for a given rank k: BM decomposition (Burer and Monteiro, 2003)

$$\min_{W\in\mathbb{R}^{m\times k},H\in\mathbb{R}^{n\times k}} \quad F(W,H) \coloneqq f\left(WH^\top\right) + \frac{\lambda}{2}\left(\|W\|_F^2 + \|H\|_F^2\right). \qquad \text{(Matrix-Factorization)}$$

- Also known as matrix factorization in the machine learning community
- If k is sufficiently large, (Matrix-Factorization) is equivalent to (Matrix-Completion) in the sense that any global solution of one can be converted to that of the other
- For k small, per-iteration cost is also low

Advantages and Disadvantages of Matrix Factorization

- Advantages
  - Avoiding expensive SVDs
  - Smooth objective
  - Low cost in matrix storage and multiplication: we can store the whole W and H directly
- Disadvantages
  - Need to pre-specify k: extra work for parameter tuning
  - Problem nonconvex: possible to get stuck at saddle points or spurious local optima
  - Exist cases with spurious local minima (Yalcin et al., 2022; O'Carroll et al., 2022)

Combining the best of the two Worlds

- Manifold identification: nuclear norm is partly smooth, with the manifold being $\mathcal{M}_X \coloneqq \{Y \mid \mathsf{rank}(Y) = \mathsf{rank}(X)\}$
- The convex formulation thus provides information with the right rank
- Solving the convex formulation also guarantees convergence to the global optimum (in objective value)
- We can switch between the two formulations whenever needed (given the SVD of X)
- Our approach in L. et al. (2022): go with (Matrix-Factorization), and whenever the algorithm seems to converge (when the gradient is small), switch to (Matrix-Completion) to escape spurious stationary points and adjust rank
- Acceleration not from high order methods, but from low iteration cost and parallelism

Inexact Proximal Gradient for (Matrix-Completion)

- Given current iterate X, Lipschitz constant L (just need a loose upper bound), update direction obtained by:

$$\Delta X \approx \underset{D\in\mathbb{R}^{m\times n}}{\operatorname{argmin}} \quad \left(Q_X(D) := \langle \nabla f(X),\, D\rangle + \frac{L}{2}\|D\|_F^2 + \lambda\|X+D\|_*\right)$$

$$\Leftrightarrow \quad X + \Delta X \approx \operatorname{prox}_{L^{-1}\lambda\|\cdot\|_*}\left(X - L^{-1}\nabla f(X)\right)$$

- Proximal operation through approximate SVD by power method with warmstart using the current iterate (no strict decrease guarantee)

$$Z := X - L^{-1}\nabla f(X), \quad X + \Delta X \in \left\{Y \mid \min_{g\in\partial Q_X(Y)} \|g\| \le \epsilon\right\}. \quad \text{(Inexact–Subgrad)}$$

- Exact proximal on inexact SVD: ϵ_t depends on precision of SVD

$$X + \Delta X = \operatorname{prox}_{\alpha\lambda\|\cdot\|_*}(Z + \mathcal{E}_t), \quad \|\mathcal{E}_t\| \le \epsilon_t$$

- The iterate is then updated by

$$X^+ = X + \Delta X.$$

Algorithm

Algorithm 2: Algorithm Framework of MF-Global

input : $\lambda > 0$, $k \in \mathbb{N}$, nonnegative sequences $\{\epsilon_t\}$, $\tilde{W}_0 \in \mathbb{R}^{m\times k}$, $\tilde{H}_0 \in \mathbb{R}^{n\times k}$

for $t = 0, \ldots$ **do**

MF Stage: Compute (W_t, H_t) as an approximate solution to (Matrix-Factorization) with rank $= k$ (starting from $(\tilde{W}_t, \tilde{H}_t)$) satisfying $F(W_t, H_t) \le F(\tilde{W}_t \tilde{H}_t^\top)$.

MC Stage: Start from $\tilde{X}_t = W_t H_t^\top$, solve (Matrix-Completion) by one step of inexact proximal gradient to get X_{t+1} satisfying (Inexact–Subgrad), and let U_t, S^t, V_t be SVD of X_{t+1} (available from the PG step)

$k \leftarrow \mathrm{rank}(S^t)$

$\tilde{W}_{t+1} \leftarrow U_t\sqrt{S^t}, \quad \tilde{H}_{t+1} \leftarrow V_t\sqrt{S^t}$

X_t or $\tilde{X}_t$ never formed explicitly, access through matrix-matrix products

Theorem 8 (L. et al. (2022))

Assume f is lower bounded. If $\sum \epsilon_t^2 < \infty$, then

1. $\mathrm{dist}(X_t, \Omega) \to 0$ *(Ω: solution set)*, $F(x_t) \to F^* := \min_X F(X)$
2. $\{X_t\}$ *has at least one limit point, and any limit point is a global solution of* (Matrix-Completion)

Theorem 8 (L. et al. (2022))

Assume f is lower bounded. If $\sum \epsilon_t^2 < \infty$, then

1. $\mathrm{dist}(X_t, \Omega) \to 0$ *(Ω: solution set)*, $F(x_t) \to F^* := \min_X F(X)$
2. $\{X_t\}$ *has at least one limit point, and any limit point is a global solution of* (Matrix-Completion)

Theorem 9

If $\epsilon_t = O(t^{-2})$, then $F(X_t) - F^* = O(t^{-1})$

Difficulty: objective not strictly decreasing, and the alternative step destroys geometry properties of PG

Theorem 8 (L. et al. (2022))

Assume f is lower bounded. If $\sum \epsilon_t^2 < \infty$, then

1. $\mathrm{dist}(X_t, \Omega) \to 0$ *(Ω: solution set),* $F(x_t) \to F^* := \min_X F(X)$
2. $\{X_t\}$ *has at least one limit point, and any limit point is a global solution of* (Matrix-Completion)

Theorem 9

If $\epsilon_t = O(t^{-2})$, then $F(X_t) - F^* = O(t^{-1})$

Difficulty: objective not strictly decreasing, and the alternative step destroys geometry properties of PG

Theorem 10 (L. et al. (2022))

If $\epsilon_t \to 0, X_{t_i} \to X^* \in \Omega$, and (NOD) *holds at X^*, then there is i_0 such that*

$$\mathrm{rank}(X_{t_i}) = \mathrm{rank}(X^*), \quad \forall i \geq i_0.$$

Numerical Results

Compare with the state of the art for (Matrix-Completion) using 8 cores:

- Active-ALT (Hsieh and Olsen, 2014): alternates between inexact PG and solving a lower-dimensional convex subproblem. Power method with warmstart for SVD
- AIS-Impute (Yao et al., 2018): Inexact APG method and the same power method for approximate SVD.

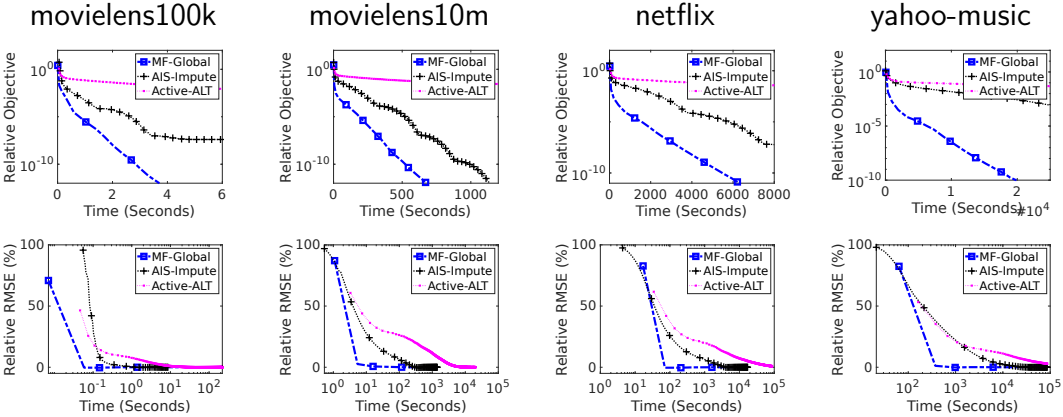


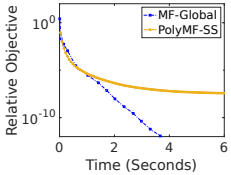
Figure: Top row: relative objective value. Bottom row: relative RMSE (time in log scale).

Data set	m	n	λ	final k
movielens100k	943	1682	15	68
movielens10m	65133	71567	100	50
netflix	17770	2649429	300	68
yahoo-music	624961	1000990	10000	52

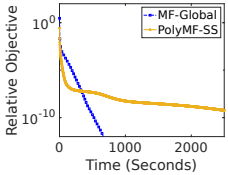
Comparison with a Solver for (Matrix-Factorization)

- Compare with running our subroutine for (Matrix-Factorization) only (PolyMF-SS, Wang et al., 2017) without convex steps, with optimal rank given to PolyMF-SS from the beginning on

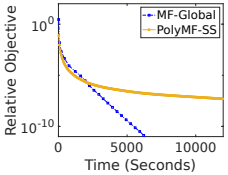
movielens100k



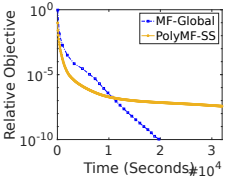
movielens10m



netflix



yahoo-music



- MF-Global escapes spurious stationary points

Outline

Degeneracy

- If we don't have any nondegeneracy condition, it is possible that the active manifold is not unique
- Iterates might jump among manifolds in this case
- But (hopefully) we can still use that to make the problem less difficult and get some partial acceleration

Best Subset Selection

Consider

$$\min_{x \in A_s} f(x), \tag{SUBSET}$$

where f has Lipschitz continuous gradient, $s \in \mathbb{N}$, and A_s is the sparsity set given by

$$A_s \coloneqq \{x \in \mathbb{R}^n : \|x\|_0 \leq s\}$$

- Nonconvex regularization
- Combinatorial nature:

$$A_s = \bigcup_{J \in \mathcal{J}_s} A_J, \quad A_J \coloneqq \operatorname{span}\{e_j : j \in J\}, \quad \mathcal{J}_s \coloneqq \{J \subseteq \{1, 2, \ldots, n\} : |J| \leq s\},$$

  with e_j being the jth standard unit vector in $\mathbb{R}^n$.
- Finite pieces of A_J, each is a subspace
- Use this idea to accelerate proximal gradient (non-unique projection, use arbitrary element)

$$w^{k+1} \in T_{\mathrm{PG}}^{\lambda}(w^k) \coloneqq P_{A_s}(w^k - \lambda \nabla f(w^k)) \tag{PG}$$

Theorem 11 (Alcantara and L. (2022))

Let $\{x^k\}$ be a sequence generated by (PG) *with $\lambda \in (0, L^{-1})$. Then the following hold:*

(a) *(Subsequential convergence) $\{f(w^k)\}$ is strictly decreasing, and limit point w^* of $\{w^k\}$ is a stationary point of* (SUBSET)*: $w^* \in P_{A_s}(w^* - \lambda \nabla f(w^*))$*

(b) *(Subspace identification and full convergence) There exists $N \in \mathbb{N}$ such that*

$$\{w^k\}_{k=N}^{\infty} \subseteq \bigcup_{J \in \mathcal{I}_{w^*}} A_J, \qquad \mathcal{I}_{w^*} := \{J \in \mathcal{J}_s : w^* \in A_J\}. \quad \text{(Non-Unique-Identification)}$$

whenever $w^k \to w^*$. In particular, if $T_{\mathrm{PG}}^{\lambda}(w^*)$ is a singleton for an limit point w^* of $\{w^k\}$, then $w^k \to w^*$, and hence (Non-Unique-Identification) *holds.*

(c) *(Local linear convergence) If $T_{\mathrm{PG}}^{\lambda}(w^*)$ is a singleton and $w \mapsto w - \lambda \nabla f(w)$ is a contraction over A_J for all $J \in \mathcal{I}_{w^*}$, then $\{w^k\}$ converges to w^* at a Q-linear rate.*

Two Acceleration Schemes

- The "identified" subspace might not be unique, but whichever we get always contains w^*
- Use (Non-Unique-Identification) to propose two acceleration schemes, both with faster convergence rates under suitable conditions
  1. Extrapolation like APG, but only when two consecutive iterates lie in the same A_J: prefer the current piece when there are multiple choices
  2. Switch to a smooth method if the piece is fixed: could get superlinear/quadratic convergence as before
     - Actually can still get superlinear/quadratic convergence even if the iterates jump among different $J \in \mathcal{I}_{w^*}$

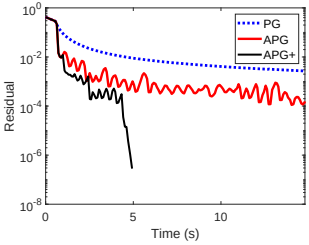
Experiments

- Compare:
  - PG
  - APG: Our same-piece extrapolation acceleration
  - APG+: plus the smooth optimization part (with Newton's method)
- Consider the residual as the problem is nonconvex:

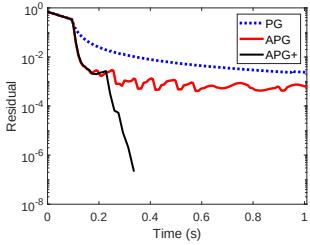
$$\text{Residual}(x) := \frac{\|x - P_{A_s}\left(x - \lambda \nabla f\left(w\right)\right)\|}{1 + \|x\| + \lambda \|\nabla f\left(x\right)\|}$$

Dataset	problem	#instances (m)	#features (n)
news20	Logistic regression	15,997	1,355,191
rcv1.binary	Logistic regression	20,242	47,236
webspam	Logistic regression	280,000	16,609,143
E2006-log1p	Least square	16,087	4,272,227
E2006-tfidf	Least square	16,087	150,360

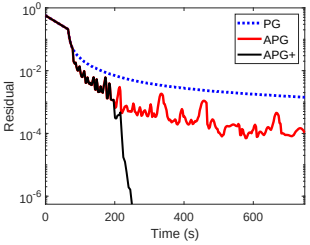
Logistic regression



() news20, $s = 0.01m$

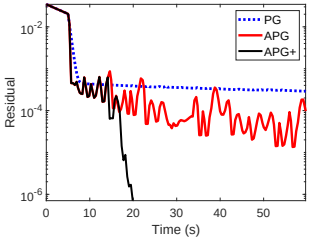


() rcv1.binary, $s = 0.01m$

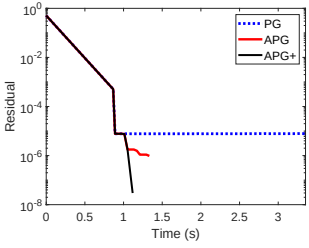


() webspam, $s = 0.001m$

Least square



() E2006-log1p, $s = 0.01m$



() E2006-tfidf, $s = 0.01m$

Thanks for Listening

Questions?

- C. L. Accelerating inexact successive quadratic approximation for regularized optimization through manifold identification, 2020. Accepted by Mathematical Programming
- Zih-Syuan Huang and C. L. Training structured neural networks through manifold identification and variance reduction. In *ICLR*, 2022
- C. L., Ling Liang, Tianyun Tang, and Kim-Chuan Toh. Escaping spurious local minima of low-rank matrix factorization through convex lifting, 2022. arXiv:2204.14067
- Jan Harold Alcantara and C. L. Accelerated projected gradient algorithms for sparsity constrained optimization problems. In *NeurIPS*, 2022