机器学习中的优化算法与分析选讲

文再文

北京大学北京国际数学研究中心

运筹千里纵横论坛

2020年5月10日

数学优化的机遇、挑战和定位

最优化算法与理论 ⟷ 机器学习的融合交叉

- **应用问题**：国家和社会的重点需求
- **模型问题**：
  - 经典问题：逻辑回归，支撑向量机，Boosting, ...
  - 干净明确的现代问题：压缩感知，低秩矩阵恢复，“相位恢复", ...
  - 计算资源驱动型问题：深度学习，强化学习, 生成对抗网络...
- **算法问题**：
  - 典型要求：对资源要求低、低复杂度、更好更快...
  - 利用问题结构特点：随机抽样的算法、KFAC ...
- **理论问题**：
  - 泛化能力: 研究模型（如深层神经网络）的泛化误差的上界...
  - 经典优化理论分析：局部/全局收敛性、复杂度分析...
  - 非凸问题全局最优解: 凸松弛、误差界/KL条件、能量景观...

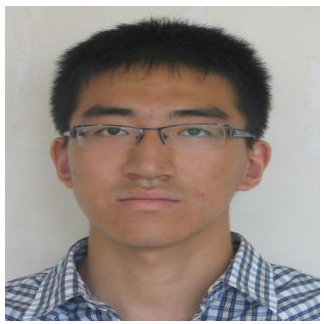
更加开放，从问题源头入手，能否解决关键核心需求?

Coauthors in our group or alumnus

(a) Andre

(b) 李勇锋

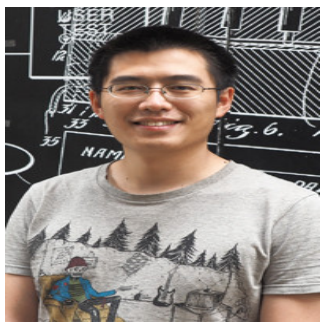
(c) 赵明明



(d) 刘浩洋

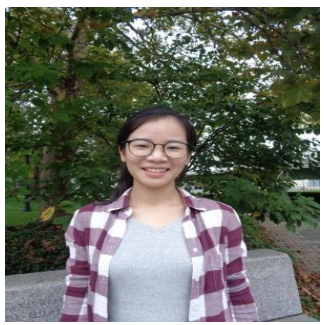
(e) 杨明瀚

(f) 许东



(g) 张君宇

(h) 马超



(i) 段雅琦

(j) 张海翔

References

- Yang Minghan, Andre Milzarek, Wen Zaiwen, Zhang Tong, **Stochastic semi-smooth Quasi-Newton method for nonsmooth optimization**
- Zhao Mingming, Li Yongfeng, Wen Zaiwen, **A stochastic trust region framework for policy optimization**
- Andre Milzarek, Xiao Xiantao, Cen Sicong, Wen Zaiwen, Michael Ulbrich; **A stochastic semi-smooth Newton method for nonsmooth nonconvex optimization**, SIAM Journal on Optimization
- Duan Yaqi, Wang Mengdi, Wen Zaiwen, Yuan Yaxiang, **Adaptive Low Rank Approximation for State Aggregation of Markov Chain**, SIAM Journal on Matrix Analysis and Applications
- Zhang Junyu, Liu Haoyang, Wen Zaiwen, Zhang Shuzhong, **A Sparse Completely Positive Relaxation of the Modularity Maximization for Community Detection**, SIAM Journal on Scientific Computing
- Ma Chao, Liu Xin, Wen Zaiwen, **Globally convergent Levenberg-Marquardt method for phase retrieval**, IEEE Transactions on Information Theory
- Zhang Haixiang, Andre Milzarek, Wen Zaiwen, Yin Wotao, **On the geometric analysis of a quartic-quadratic optimization problem under a spherical constraint**, arXiv: 1908.00745

Outline

Why Optimization in Machine Learning?

Many problems in ML can be written as

$$\min_{x\in\mathcal{W}} \quad \sum_{i=1}^{N} \frac{1}{2}\|a_i^\top x - b_i\|_2^2 + \mu\varphi(x) \quad \text{linear regression}$$

$$\min_{x\in\mathcal{W}} \quad \frac{1}{N}\sum_{i=1}^{N} \log(1+\exp(-b_i a_i^\top x)) + \mu\varphi(x) \quad \text{logistic regression}$$

$$\min_{x\in\mathcal{W}} \quad \frac{1}{N}\sum_{i=1}^{N} \ell(h(x,a_i),b_i) + \mu\varphi(x) \quad \text{general formulation}$$

- The pairs (a_i, b_i) are given data, b_i is the label of the data point a_i
- $\ell_i(\cdot)$: measures model fit for data point i (avoids under-fitting)
- $\varphi(x)$: regularization avoids over-fitting: $\|x\|_2^2$ or $\|x\|_1$, etc
- $h(x,a)$: linear function or models constructed from deep neural networks

Sparse Logistic Regression

The ℓ_1 logistic regression problem.

$$\min_{x\in\mathbb{R}^n} \quad \frac{1}{N}\sum_{i=1}^{N}\log(1+\exp(-b_i a_i^T x)) + \mu\varphi(x).$$

- The data pair $\{a_i, b_i\} \in \mathbb{R}^n \times \{-1, 1\}, i \in [N]$,

Data Set	# data N	# features n	sparsity(%)
cina	16,033	132	70.49
a9a	32,561	123	88.72
ijcnn1	49,990	22	40.91
covtype	581,012	54	77.88
url	2,396,130	3,231,961	99.99
susy	5,000,000	18	1.18
higgs	11,000,000	28	7.89
news20	19,996	1,355,191	99.97
rcv1	20,242	47,236	99.84
kdda	8,407,752	20,216,830	99.99

Deep Learning

The objective function is the CrossEntropy function plus ℓ_1 term:

$$\min_x \quad \frac{1}{N}\sum_{i=1}^{N} -\log\left(\frac{\exp(h(x,a_i)[b_i])}{\sum_j \exp(h(x,a_i)[j])}\right) + \mu\varphi(x)$$

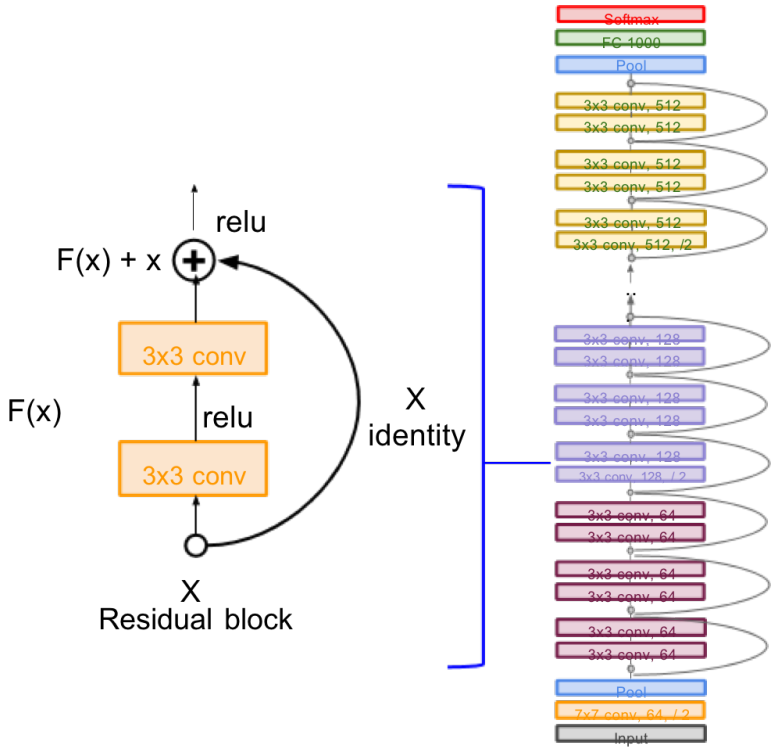
where $h(x, a_i)$ is output from network, and (a_i, b_i) are data points.

	Cifar-10	Cifar-100
# num_class	10	100
# number per class (training set)	5,000	500
# number per class (testing set)	1,000	100
# Total parametes of VGG-16	15,253,578	15,299,748
# Total parameters of ResNet-18	11,173,962	11,220,132

Table: A description of datasets used in the neural network experiments

ResNet Architecture

- Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Cited by **44889 (2015-2020/5)** at Google scholar
- Stack residual blocks. Every residual block has two 3x3 conv layers.
- Make networks from shallow to deep.
- Fancy network architecture. Many Applications.
- High-computationally-cost!
- ResNet-50 on ImageNet, **29 hours using 8 Tesla P100 GPUs**



Stochatic optimization problem

- Consider

$$\min_{x \in \mathbb{R}^n} \ \Psi(x) := f(x) + \varphi(x)$$

- Expected and Empirical Risk Minimization:

$$f(x) := \mathbb{E}[F(x, \xi)], \qquad f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$$

- Assume $f(x)$ is smooth but $\varphi(x)$ is convex and non-smooth.
- Large-scale machine learning problems: both the number of data samples N and dimension n are very large
- Full evaluation of $f(x)$ and $\nabla f(x)$ is not tractable or simply too expansive.

Stochastic Gradient Algorithms in Deep learning

Consider problem $\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n f_i(x)$

References: chapter 8 in www.deeplearningbook.org

- Gradient descent

$$x^{t+1} = x^t - \frac{\alpha^t}{n}\sum_{i=1}^n \nabla f_i(x^t)$$

- Stochastic gradient descent

$$x^{t+1} = x^t - \alpha^t \nabla f_i(x^t)$$

- Adaptive Subgradient Methods (Adagrad): let $g_t = \nabla f_i(x^t)$, $g_t^2 = \mathrm{diag}[g_t g_t^T] \in \mathbb{R}^d$, and initial $G_1 = g_1^2$. At step t

$$x^{t+1} = x^t - \frac{\alpha^t}{\sqrt{G^t + \epsilon \mathbf{1}_d}}\nabla f_i(x^t)$$
$$G^{t+1} = G^t + g_{t+1}^2$$

where we use element-wise vector-vector multiplication.

The KFAC Method

- Take an L-layer feed-forward neural network for example

$$s_j = W_j u_{j-1}, \quad u_j = \phi_j(s_j),$$

where $u_0 = a$ is the input, $u_L(a) \in \mathbb{R}^m$ is the output, W_j is the weight matrix and ϕ_j is the block-wise activation function.

- KFAC approximates FIM by a block-diagonal matrix.

$$\mathbf{F}^j := Q_{j-1,j-1} \otimes G_{j,j},$$

where $g_j^i(z) := \frac{\partial \ell(h(x,a_i),b)}{\partial s_j^i}$ and

$$Q_{j-1,j-1} = \frac{1}{|B|} \sum_{i \in B} u_{j-1}^i (u_{j-1}^i)^\top, G_{j,j} = \frac{1}{|B|} \sum_{i \in B} \mathbb{E}_{z \sim p(z|x,a_i)} [g_j^i(z) g_j^i(z)^\top],$$

- KFAC update: $d^j = -(\mathbf{F}^j)^{-1} g_j$

Hessian of Smooth Problem ($\varphi(x) = 0$)

The Hessian matrix $\nabla^2\Psi(x)$ can be divided into two different parts:

$$\nabla^2\Psi(x) = H(x) + \Pi(x),$$

where $H(x)$ is relatively cheap, $\Pi(x)$ is expensive.

- Subsampled Hessian matrix

$$\nabla^2_{\mathcal{S}_H}\Psi(x) := \frac{1}{|\mathcal{S}_H|}\sum_{i\in\mathcal{S}_H}\nabla^2 f_i(x).$$

- Deep learning

$$H(x) = \frac{1}{N}\sum_{i=1}^{N} J_h^i \nabla_h^2 \ell_i(x)(J_h^i)^\top, \quad \Pi(x) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m}\nabla_{h_j}\ell_i(x)\nabla_x^2 h_j^i(x),$$

- KFAC Approximation

Structured Stochastic Quasi-Newton (QN) Methods

- Quasi-Newton direction:

$$(B_k + \lambda_k I)d^k = -g_k$$

- Approximation to the Hessian

$$B_k = H_k + A_k,$$

where $H_k = H(x^k)$ and A_k is a quasi-Newton refinement to $\Pi(x^k)$

$$A_k := \text{LBFGS}(U_k, V_k) = A_k^0 - C_k P_k^{-1} C_k^\top$$

- Explicit computation:

$$(B_k + \lambda_k I)^{-1} = (\tilde{H}_k - C_k P_k^{-1} C_k^\top)^{-1} = \tilde{H}_k^{-1} + \tilde{H}_k^{-1} C_k T_k^{-1} C_k^\top \tilde{H}_k^{-1}$$

- Explicit Inverse by Low-Rank Structures

Composite optimization problem

Consider $\min_{x\in\mathbb{R}^n}\ \Psi(x) := f(x) + \varphi(x)$

- Proximal gradient method

$$x^{k+1} \quad = \quad \mathrm{prox}_{\varphi}^{\lambda}\left(x^k - \nabla f(x^k)/\lambda\right), k = 0, 1, \cdots,$$

where the *proximal mapping* is:

$$\mathrm{prox}_{\varphi}^{\lambda}(x) := \underset{u\in\mathbb{R}^n}{\mathrm{argmin}}\ \{\varphi(u) + \frac{\lambda}{2}\|u - x\|_2^2\}.$$

- Basic idea is to solve the equivalent nonlinear equation:

$$F(x) := x - \mathrm{prox}_{\varphi}^{\lambda}(x - \nabla f(x)/\lambda) = 0.$$

- $F(x)$ is nondifferentiable, but semi-smooth in many applications.
  (a) F is directionally differentiable at x; and
  (b) for any $d \in \mathbb{R}^n$ and $J \in \partial F(x + d)$,

$$\|F(x + d) - F(x) - Jd\|_2 = o(\|d\|_2) \quad \text{as } d \to 0.$$

- Apply semi-smooth Newton-type method !

Semi-smooth Newton-type method

- Denote J^k as the generalized Jacobian (Hessian) matrix $\partial F(x^k)$.
- Construct a "Newton" direction:

$$d^k = -W^k F(x^k),$$

  where W^k is exact or approximation of inverse of J^k.
- Employ the Newton step

$$x^{k+1} = x^k + d^k.$$

- However, there exists some bottleneck:
  (a) a suitable globalization strategy
  (b) the computation of W^k and $F(x^k)$ for very high dimensional problem.

Stochastic Approximation and Extragradient Strategy

- Use stochastic approximation technique!
  Estimate $v^k \approx \nabla f(x^k)$ from stochastic oracle and set

$$F_{v^k}(x^k) := x^k - \mathrm{prox}_{\varphi}^{\lambda}(x^k - v^k/\lambda).$$

  Example: Assume the samples s are chosen independently, then a possible estimate of $\nabla f(x)$ is $\nabla f_s(x^k) := \frac{1}{|s|}\sum_{i\in s}\nabla f_i(x^k)$.

- Use extra-gradient step for globalization!

  (a) First employ the "Newton" step:

$$z^k = x^k + \beta_k d^k,$$

  where $d^k = -W^k F_{v^k}(x^k)$.

  (b) Perform an extra gradient step:

$$x^{k+1} = \mathrm{prox}_{\varphi}^{\lambda}(x^k + \alpha_k d^k - v_+^k/\lambda), \quad v_+^k \approx \nabla f(z^k).$$

The choice of β_k and α_k are very flexible !

Convergence Assumption

Basic Assumption

A.1 The gradient mapping ∇f is Lipschitz continuous on $\mathbb{R}^n$ with modulus $L_f \geq 1$.

A.2 The objective function ψ is bounded from below on $\mathbf{dom}\ \varphi$.

A.3 $\varphi : \mathbb{R}^n \to (-\infty, \infty]$ is convex, lower semicontinuous, and proper.

Stochastic Assumption

B.1 The mapping $\mathsf{D}^k : \Omega \to \mathbb{R}^n$ is an $\mathcal{F}^k$-measurable function for all k.

B.2 There is $\nu_k > 0$ such that we have $\mathbb{E}[\|\mathsf{D}^k\|^2 \mid \mathcal{F}_+^{k-1}] \leq \nu_k^2 \cdot \mathbb{E}[\|F_{\mathsf{V}^k}(\mathsf{X}^k)\|^2 \mid \mathcal{F}_+^{k-1}]$ a.e. and for all $k \in \mathbb{N}$.

B.3 For all $k \in \mathbb{N}$, it holds $\mathbb{E}[\mathsf{V}^k \mid \mathcal{F}_+^{k-1}] = \nabla f(\mathsf{X}^k)$, $\mathbb{E}[\mathsf{V}_+^k \mid \mathcal{F}^k] = \nabla f(\mathsf{Z}^k)$ a.e. and there exists $\sigma_k, \sigma_{k,+} > 0$ such that a.e.

$$\mathbb{E}[\|\nabla f(\mathsf{X}^k) - \mathsf{V}^k\|^2 \mid \mathcal{F}_+^{k-1}] \leq \sigma_k^2 \quad \text{and} \quad \mathbb{E}[\|\nabla f(\mathsf{Z}^k) - \mathsf{V}_+^k\|^2 \mid \mathcal{F}^k] \leq \sigma_{k,+}^2,$$

where

$$\mathcal{F}^k = \sigma(V^0, V_+^0, \ldots, V^k) \quad \text{and} \quad \mathcal{F}_+^k = \sigma(\mathcal{F}_k \cup \sigma(V_+^k)).$$

Theorem 1

Suppose that the assumptions (A.1)–(A.3) *and* (B.1)–(B.3) *are satisfied and we have*

$$\lambda_{k,+} \leq \frac{1}{L_f}, \quad \lambda_k \leq \frac{(1-\bar{\rho})\lambda_{k,+}}{1+\mu_k^2},$$

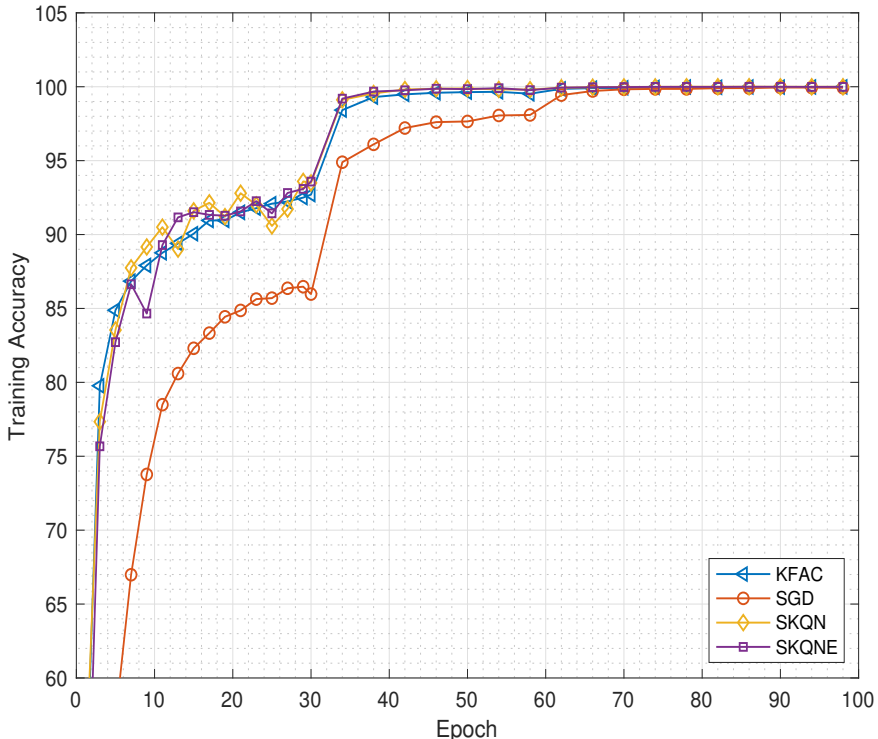
where $\mu_k = \nu_k(\alpha_k + L_f\beta_k\lambda_{k,+})$. *Then, under the additional conditions*

$$\sum \lambda_k = \infty, \quad \sum \lambda_k \sigma_k^2 < \infty, \quad \sum \lambda_{k,+}\sigma_{k,+}^2 < \infty$$

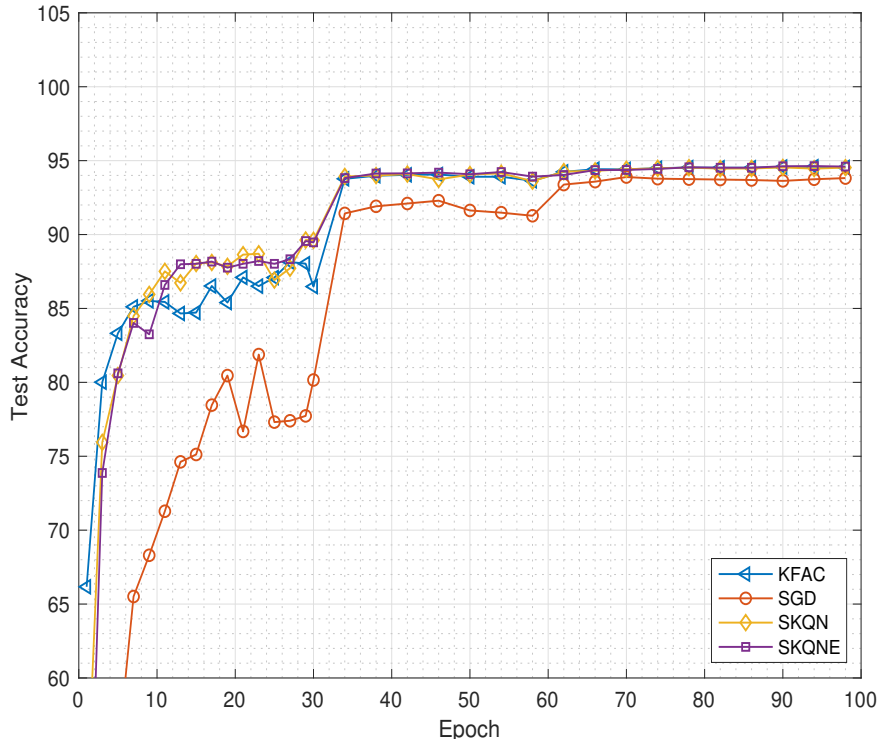
it follows $\liminf_{k\to\infty} \mathbb{E}[\|F(\mathsf{X}^k)\|^2] = 0$ *and* $\liminf_{k\to\infty} F(\mathsf{X}^k) = 0$ *a.s. and* $(\psi(\mathsf{X}^k))_k$ *a.s. converges to some random variable* Y^* *with* $\lim_{k\to\infty} \mathbb{E}[\psi(\mathsf{X}^k)] = \mathbb{E}[\mathsf{Y}^*]$.

Locally*, if we further assume the function satisfy KL-property and some mild assumption, we can show then* $(\mathsf{X}^k)_k$ *converges almost surely to a* $\operatorname{crit}\psi$*-valued random variable* X^*.

Deep learning: ResNet-18 on Cifar10. $\varphi(x) = 0$

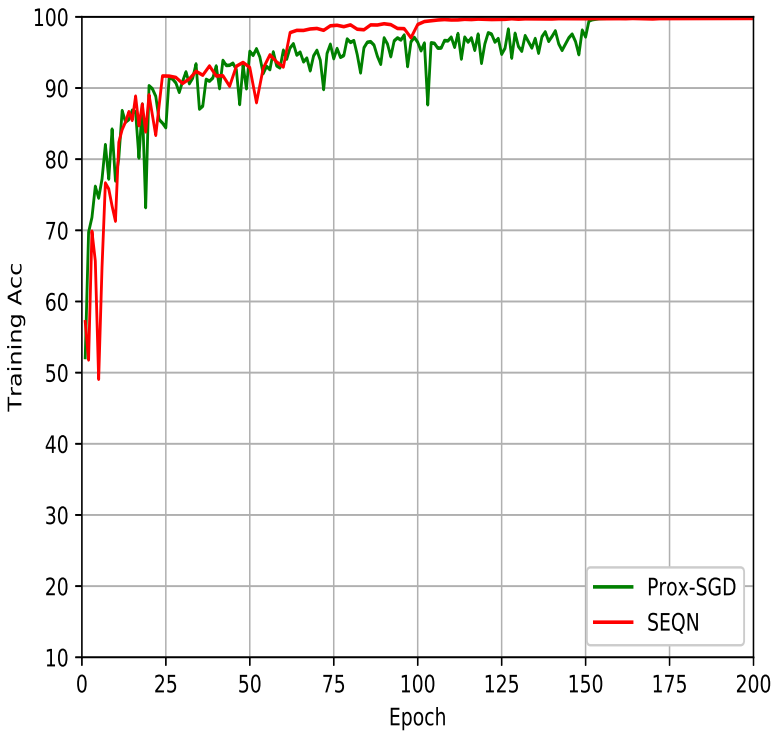


(k) Training accuracy

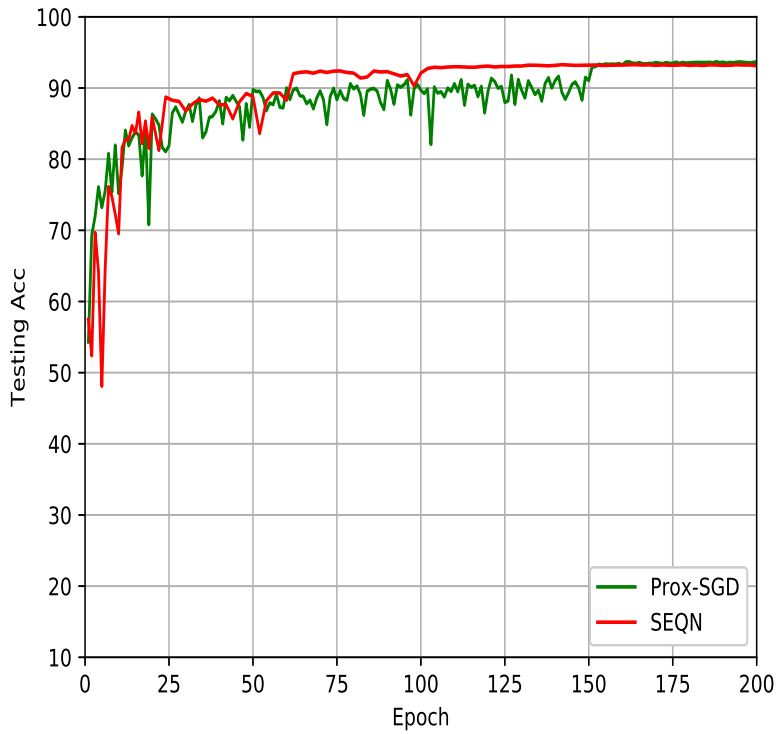


(l) Testing accuracy

Deep learning: ResNet-18 on Cifar10, $\varphi(x) = \|x\|_1$



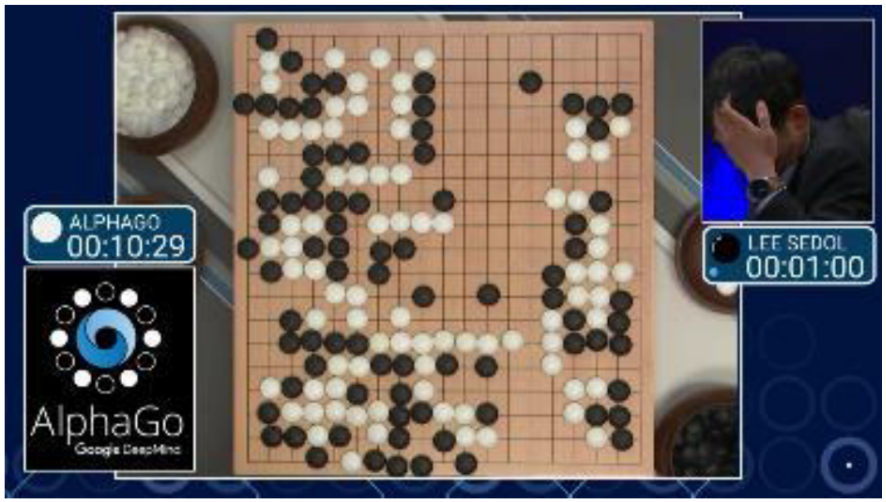
(a) Training accuracy



(b) Testing accuracy

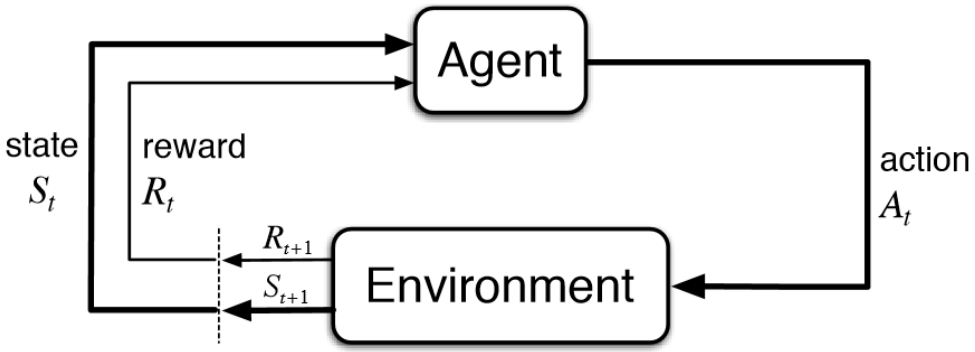
Outline

Reinforcement learning



Preliminaries

- Consider an infinite-horizon discounted Markov decision process (MDP), usually defined by a tuple $(\mathcal{S}, \mathcal{A}, P, R, \rho_0, \gamma)$;



- ρ_0: the distribution of s_0
- γ: discount factor $\in (0, 1)$
- P: transition probability

- A trajectory: $\tau = \{s_0, a_0, r(s_0, a_0), s_1, \ldots, s_t, a_t, r(s_t, a_t), s_{t+1}, \ldots\}$.
- At a given state, choose action from $\pi(\cdot|s)$: $\int_{\mathcal{A}} \pi(a|s)da = 1$.
- The policy is supposed to maximize the total expected reward:

$$\max_{\pi} \ \eta(\pi) = \mathrm{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right],$$

$$\text{with } s_0 \sim \rho_0, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t).$$

Preliminaries

- State-action value function:

$$Q_\pi(s,a) = \mathrm{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\middle| s_0 = s, a_0 = a\right],$$

$$Q_\pi(s,a) = r(s,a) + \gamma\sum_{s'}P(s'|s,a)\sum_{a'}\pi(a'|s')Q_\pi(s',a'),$$

$$Q^*(s,a) = r(s,a) + \gamma\sum_{s'}P(s'|s,a)\max_{a'}\ Q^*(s',a').$$

- State value function:

$$V_\pi(s) = \sum_a \pi(a|s)Q_\pi(s,a) = \sum_a \pi(a|s)\left[r(s,a) + \gamma\sum_{s'}P(s'|s,a)V_\pi(s')\right],$$

$$V^*(s) = \max_a Q^*(s,a) = \max_a\left[r(s,a) + \gamma\sum_{s'}P(s'|s,a)V^*(s')\right].$$

- Advantage function: $A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s)$, $\sum_a \pi(a|s)A_\pi(s,a) = 0$.

Deep reinforcement learning

- In real-world tasks: high dimensionality, limited observations,...
- In deep reinforcement learning, the policy π and/or value functions are usually parameterized with differentiable neural networks.
- The policy-based optimization:

$$\max_{\theta} \quad \eta(\theta).$$

- The value-based optimization:

$$\min_{\phi} \mathrm{E}_{s,a}\left\{Q_{\phi}(s,a) - \mathrm{E}_{s' \sim \mathrm{P}(\cdot|\mathrm{s},\mathrm{a})}\left[r(s,a) + \gamma \max_{a'} Q_{\phi}(s',a')|s,a\right]\right\}^2.$$

- Challenges: theoretical analysis; generalization; stability; trade off between exploration and exploitation...

VPG, NPG

- Policy gradient: $\nabla\eta(\theta) = \mathbb{E}_{\rho_\theta,\pi_\theta}\left[\nabla \log \pi_\theta(a|s) A_\theta(s,a)\right]$.
- $\rho_\theta(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi_\theta)$ is the (unnormalized) discounted visitation frequencies.
- Vanilla[1]/Natural[2] policy gradient: $\theta_{k+1} = \theta_k + \alpha M(\theta_k)\nabla_\theta \eta(\theta_k)$.
- $M(\theta_k)^{-1} = \mathbb{E}_{\rho_{\theta_k},\pi_{\theta_k}}\left[\nabla_\theta \log \pi_{\theta_k}(s,a) \nabla_\theta \log \pi_{\theta_k}(s,a)^T\right]$.
- A local approximation of η:

$$\eta(\theta) = \eta(\theta_k) + \sum_s \rho_\theta(s) \sum_a \pi_\theta(a|s) A_{\theta_k}(s,a),$$

$$L_{\theta_k}(\theta) = \eta(\theta_k) + \sum_s \rho_{\theta_k}(s) \sum_a \pi_\theta(a|s) A_{\theta_k}(s,a).$$

- $\eta(\theta_k) = L_{\theta_k}(\theta_k), \nabla\eta(\theta_k) = \nabla L_{\theta_k}(\theta_k)$.

[1] R. S. Sutton, el al., Policy gradient methods for reinforcement learning with function approximation.

[2] S. M. Kakade, A natural policy gradient.

TRPO, PPO

- Trust region policy optimization (TRPO)[3]:

$$\max_{\theta} \; L_{\theta_k}(\theta), \text{ s.t., } \bar{D}_{KL}(\pi_{\theta_k} \| \pi_\theta) \leq \delta,$$

$$\max_{\theta} \; g_k^T(\theta - \theta_k), \text{ s.t., } \frac{1}{2}(\theta - \theta_k)^T H_k(\theta - \theta_k) \leq \delta.$$

- $g_k = \nabla L_{\theta_k}(\theta)|_{\theta=\theta_k}$, $H_k = \nabla^2 \bar{D}_{KL}(\pi_{\theta_k} \| \pi_\theta)|_{\theta=\theta_k}$.
- $\theta_{k+1} = \theta_k + \alpha H_k^{-1} g_k$, essentially a natural gradient update.
- Proximal policy gradient (PPO) method[4], denote $r_k(s,a) = \frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}$:

$$L_{\theta_k}(\theta) = \eta(\theta_k) + \mathrm{E}_{\rho_{\theta_k}, \pi_{\theta_k}} \left[r_k(s,a) A_{\theta_k}(s,a) \right],$$

$$\max_{\theta} \mathrm{E}_{\rho_{\theta_k}, \pi_{\theta_k}} \left[\min \left[r_k(s,a) A_{\theta_k}(s,a), clip(r_k(s,a), 1-\epsilon, 1+\epsilon) A_{\theta_k}(s,a) \right] \right].$$

[3] J. Schulman, et al., Trust region policy optimization.

[4] J. Schulman, et al., Proximal policy optimization algorithms

Stochastic Trust Region Algorithm

- The objective function

$$\max_{\theta} \quad \eta(\theta).$$

- At k-th iteration, obtain a trail point $\tilde{\theta}_{k+1}$ from the subproblem:

$$\max_{\theta} \quad L_{\theta_k}(\theta), \quad \text{s.t. } \mathbb{E}_{s\sim\rho_{\theta_k}}\left[D(\pi_{\theta_k}(\cdot|s), \pi_{\theta}(\cdot|s))\right] \leq \delta_k.$$

- Compute the ratio $r_k = \frac{\eta(\tilde{\theta}_{k+1}) - \eta(\theta_k)}{L_{\theta_k}(\tilde{\theta}_{k+1}) - L_{\theta_k}(\theta_k)}$.
- Update $\theta_{k+1} = \begin{cases} \tilde{\theta}_{k+1}, & r_k \geq \beta_0, \\ \theta_k, & \text{o.w.}, \end{cases}$ with $\beta_0 > 0$.
- Update $\delta_{k+1} = \mu_{k+1}\|\nabla L_{\theta_{k+1}}(\theta_{k+1})\|$ with $\gamma_1 > 1 \geq \gamma_2 > \gamma_3$,

$$\mu_{k+1} = \begin{cases} \gamma_1\mu_k, & r_k \geq \beta_1, \\ \gamma_2\mu_k, & r_k \in [\beta_0, \beta_1), \\ \gamma_3\mu_k, & \text{o.w.}, \end{cases}.$$

Convergence Results

Lemma 2 (Lower bound of ΔL_{π_k})

Suppose $\{\pi_k\}$ *is the sequence generated by our trust region method, then we have* $L_{\pi_k}(\pi_{k+1}) - L_{\pi_k}(\pi_k) \geq \min(1, (1-\gamma)\delta_k)\mathbb{A}^*_{\pi_k}$.

Lemma 3 (Lower bound of r_k)

The ratio r_k *satisfies that* $r_k \geq \min\left(1 - \frac{4\epsilon_k\gamma\delta_k^2}{p_0^2(1-\gamma)^2\mathbb{A}^*_{\pi_k}}, 1 - \frac{4\epsilon_k\gamma\delta_k}{p_0^2(1-\gamma)^3\mathbb{A}^*_{\pi_k}}\right)$, *where* $p_0 = \min_s \rho_0(s)$ *and* $\epsilon_k = \max_{s,a} |A_{\pi_k}(s,a)|$.

Theorem 4 (Convergence)

Suppose $\{\pi_k\}$ *is the sequence generated by our trust region method, then we have the following conclusions*

1. $\varliminf_{k\to\infty} \mathbb{A}^*_{\pi_k} = 0$.
2. $\lim_{k\to\infty} \eta(\pi_k) = \eta(\pi^*)$, *where* π^* *is the optimal policy.*

Empirical algorithm

- Terminate condition:

$$\frac{|\hat{L}_{\theta_k}(\theta_{k,l+1}) - \hat{L}_{\theta_k}(\theta_{k,l})|}{1 + \left|\hat{L}_{\theta_k}(\theta_{k,l})\right|} \leq \epsilon, \text{ or } \frac{|\mathrm{Ent}(\theta_{k,l+1}) - \mathrm{Ent}(\theta_k)|}{1 + |\mathrm{Ent}(\theta_k)|} \geq \epsilon.$$

- Ratio:

$$r_k = \frac{\eta(\tilde{\theta}_{k+1}) - \eta(\theta_k)}{L_{\theta_k}(\tilde{\theta}_{k+1}) - L_{\theta_k}(\theta_k)} \Longrightarrow r_k = \frac{\hat{\eta}(\tilde{\theta}_{k+1}) - \hat{\eta}(\theta_k)}{\hat{\sigma}_\eta(\theta_k) + \hat{L}_{\theta_k}(\tilde{\theta}_{k+1}) - \hat{L}_{\theta_k}(\theta_k)}.$$

- $\hat{\sigma}_\eta(\theta)$ is the empirical standard deviation of $\eta(\theta)$.
- Acceptance criteria: $\theta_{k+1} = \begin{cases} \tilde{\theta}_{k+1}, & r_k \geq \beta_0, \\ \theta_k, & \text{o.w.} \end{cases}$, with a small negative constant $\beta_0 < 0$.
- Mandatory acceptance: after several consecutive rejections, force to accept the best performed point among the past rejections.

Mujoco in Baselines

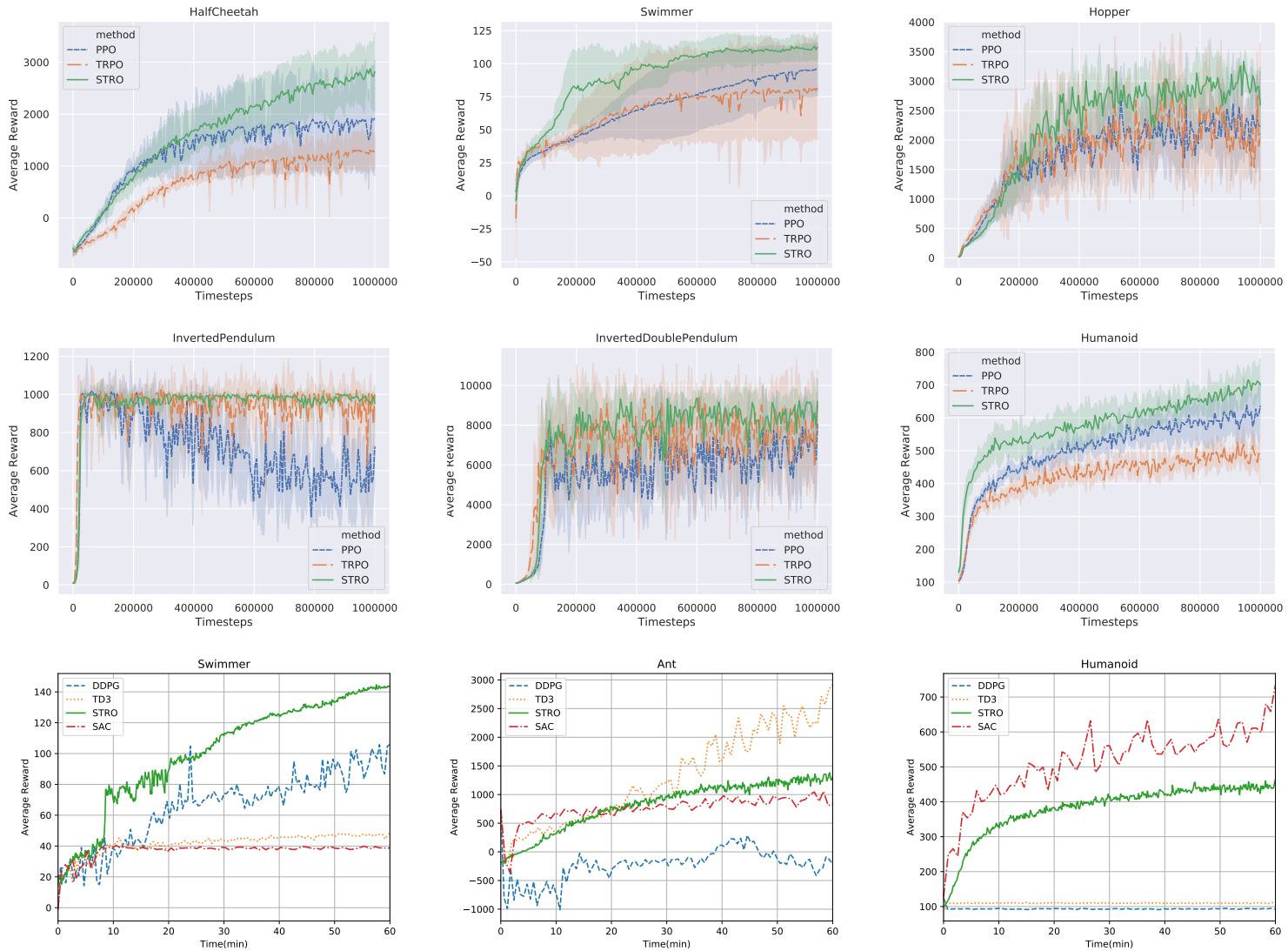


Figure: Training curves on Mujoco-v2 continuous control benchmarks.

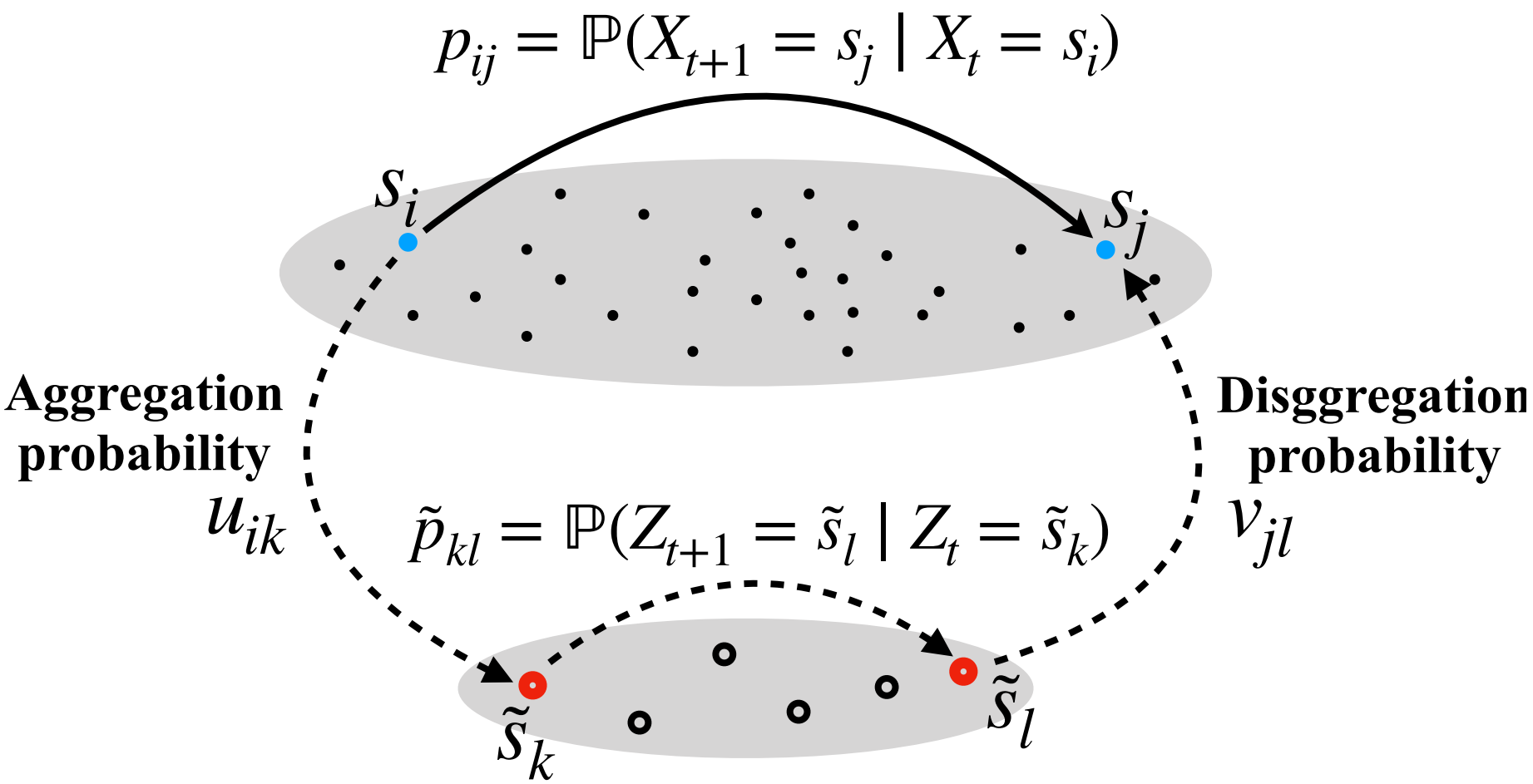
Atari games

Table: Max Average Reward (100 episodes) $\pm$ standard deviation over 5 trails of 1e7 time steps.

Environment	PPO	TRPO	STRO
Pong	20$\pm$0	3$\pm$7	**20**$\pm$0
MsPacman	2125$\pm$322	1538$\pm$159	**2452**$\pm$487
Seaquest	1004$\pm$141	692$\pm$92	**1172**$\pm$346
Bowling	50$\pm$17	38$\pm$15	**105**$\pm$6
Freeway	30$\pm$0	28$\pm$3	**31**$\pm$0
PrivateEye	100$\pm$0	88$\pm$16	**100**$\pm$0

Outline

Model Reduction by State Aggregation



$$p_{ij} = \sum_{k,l=1}^{r} u_{ik}\tilde{p}_{kl}v_{jl} \Rightarrow \text{Martrix form } P = U\tilde{P}V^T$$

Low-Nonnegative-Rank Approximation

$$\text{minimize}_{X\in\mathbb{R}^{d\times d}} \quad f_\lambda(X) := g(X) + \chi_{\mathcal{E}}(X) + \lambda\Omega(X) \tag{1}$$

Theorem 5 (Sufficient and necessary conditions for global optimality)

$\hat{X}$ is globally optimal for (1) *with an optimal factorization $\hat{X} = \hat{U}\hat{V}^T$ iff $\exists \mu \in \mathbb{R}^d$ s.t.*

$$\begin{cases} \mathbf{u}^T\left(\mu\mathbf{1}_d^T - \nabla g(\hat{X})\right)\mathbf{v} \le \lambda, \quad \forall \mathbf{u},\mathbf{v}\in\mathbb{R}_+^d \text{ with } \|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1, \\ \left[\mu\mathbf{1}_s^T - \nabla g(\hat{X})\hat{V}\right]_+ = \lambda\hat{U}\mathbf{diag}\left\{\dfrac{\|\hat{V}_j\|_2}{\|\hat{U}_j\|_2}\right\}_{j=1}^s, \\ \left[\mathbf{1}_d\mu^T\hat{U} - \left(\nabla g(\hat{X})\right)^T\hat{U}\right]_+ = \lambda\hat{V}\mathbf{diag}\left\{\dfrac{\|\hat{U}_j\|_2}{\|\hat{V}_j\|_2}\right\}_{j=1}^s. \end{cases}$$

- $g(X) := \frac{1}{2}\left\|\hat{\Xi}(\hat{P}^{(n)} - X)\right\|_F^2$, $\chi_{\mathcal{E}}(X)$ is the indicator function on $X\mathbf{1}_d = \mathbf{1}_d$.
- $\Omega(X)$ does not have an explicit form.

Outline

Phase retrieval by non-convex optimization

Solve the system of quadratic equations:

$$y_r = |\langle a_r, x\rangle|^2, \quad r = 1, 2, ..., m.$$

- **Gaussian model:**

$$a_r \in \mathbb{C}^n \overset{i.i.d.}{\sim} \mathcal{N}(0, I/2) + i\mathcal{N}(0, I/2).$$

Nonlinear least square problem

$$\min_{z \in \mathbf{C}^n} \quad f(z) = \frac{1}{4m} \sum_{k=1}^{m} (y_k - |\langle a_k, z\rangle|^2)^2$$

f is nonconvex, many local minima

Wirtinger flow: Candes, Li and Soltanolkotabi ('14)

- **Spectral Initialization:**
  1. Input measurements $\{a_r\}$ and observation $\{y_r\}(r = 1, 2, ..., m)$.
  2. Calculate z_0 to be the leading eigenvector of $Y = \frac{1}{m}\sum_{r=1}^{m} y_r a_r a_r^*$.
  3. Normalize z_0 such that $\|z_0\|^2 = n\frac{\sum_r y_r}{\sum_r \|a_r\|^2}$.

- **Iteration via Wirtinger derivatives:** for $\tau = 0, 1, \ldots$

$$z_{\tau+1} = z_\tau - \frac{\mu_{\tau+1}}{\|z_0\|^2}\nabla f(z_\tau)$$

The Modified LM method for Phase Retrieval

Levenberg-Marquardt Iteration:

$$z_{k+1} = z_k - \left(\Psi(z_k) + \mu_k I\right)^{-1} g(z_k)$$

Algorithm

1. **Input:** Measurements $\{a_r\}$, observations $\{y_r\}$. Set $\epsilon \geq 0$.
2. Construct z_0 using the spectral initialization algorithms.
3. **While** $\|g(z_k)\| \geq \epsilon$ **do**
   - Compute s_k by solving equation

$$\Psi_{z_k}^{\mu_k} s_k = \left(\Psi(z_k) + \mu_k I\right) s_k = -g(z_k).$$

     until

$$\left\|\Psi_{z_k}^{\mu_k} s_k + g(z_k)\right\| \leq \eta_k \|g(z_k)\|.$$

   - Set $z_{k+1} = z_k + s_k$ and $k := k + 1$.
3. **Output:** z_k.

Convergence of the Gaussian Model

Theorem 6

If the measurements follow the Gaussian model, the LM equation is solved accurately ($\eta_k = 0$ for all k), and the following conditions hold:

- $m \geq cn \log n$, *where* c *is sufficiently large;*
- *If* $f(z_k) \geq \frac{\|z_k\|^2}{900n}$, *let* $\mu_k = 70000n\sqrt{nf(z_k)}$; *if else, let* $\mu_k = \sqrt{f(z_k)}$.

Then, with probability at least $1 - 15e^{-\gamma n} - 8/n^2 - me^{-1.5n}$, *we have* $dist(z_0, x) \leq (1/8)\|x\|$, *and*

$$\mathrm{dist}(z_{k+1}, x) \leq c_1 dist(z_k, x),$$

Meanwhile, once $f(z_s) < \frac{\|z_s\|^2}{900n}$, *for any* $k \geq s$ *we have*

$$dist(z_{k+1}, x) < c_2 dist(z_k, x)^2.$$

Numerical Result: Natural Image

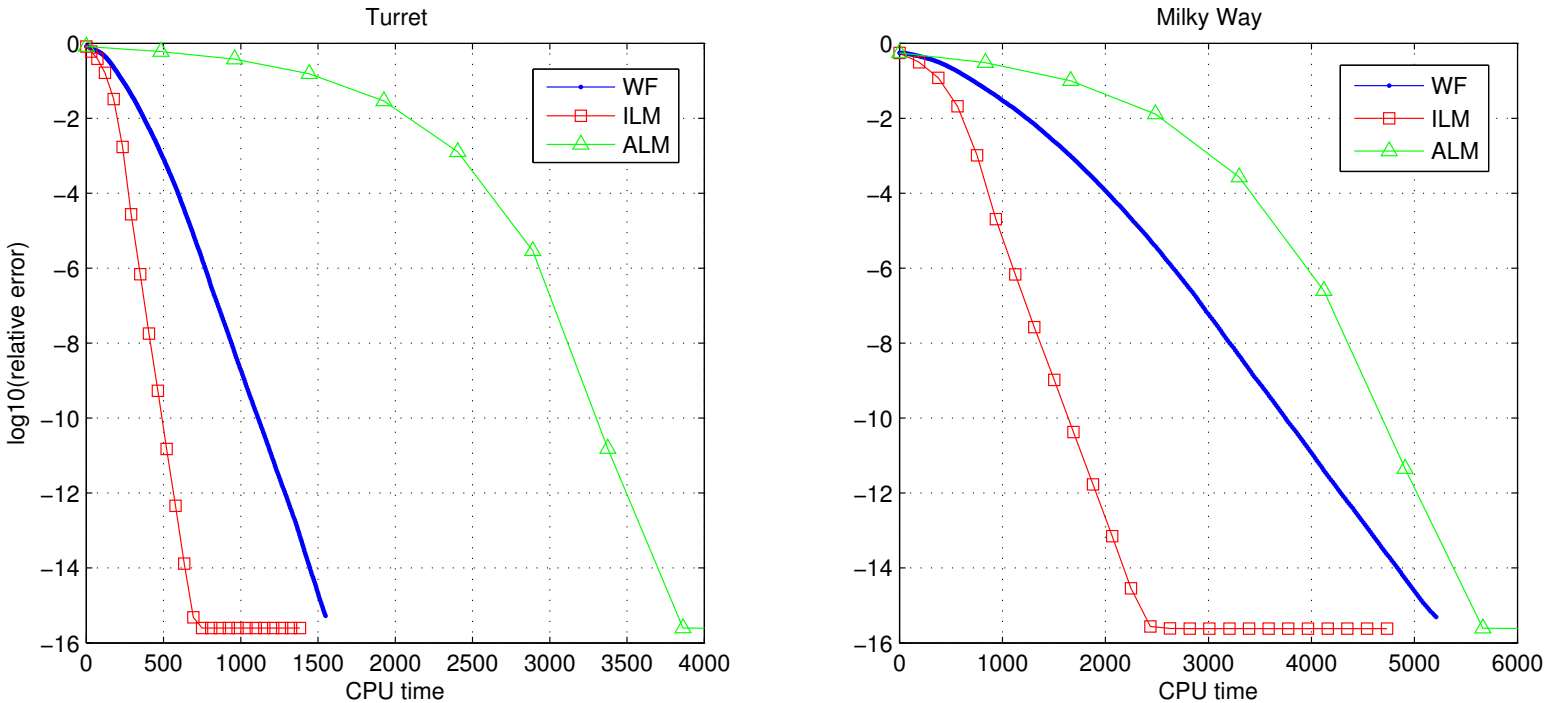


Figure: Relation between relative error and CPU time used for natural images recovery.

Outline

Modularity minimization for community detection

- The modularity maximization problem $X = \Phi^*(\Phi^*)^\top$:

$$\begin{aligned} \max \quad & \langle A - \tfrac{1}{2\lambda} dd^T, X \rangle \\ \text{s.t.} \quad & X \in \{0,1\}^{n\times n} \text{ is a partiton matrix.} \end{aligned}$$

- Nonconvex completely positive relaxation:

$$\min_{U\in\mathbb{R}^{n\times k}} \langle -A + \frac{1}{2\lambda} dd^T, UU^T \rangle$$
$$\text{s.t.} U \geq 0, \|u_i\|^2 = 1, \|u_i\|_0 \leq p, i = 1, \ldots, n$$

Theorem 7 (Theoretical Error Bounds)

Define $G_a = \sum_{i\in C_a^*} \theta_i, H_a = \sum_{b=1}^k B_{ab} G_b, f_i = H_a \theta_i$, *Under the assumption* $\max_{1\leq a<b\leq k} \frac{B_{ab}+\delta}{H_a H_b} < \lambda < \min_{1\leq a\leq k} \frac{B_{aa}-\delta}{H_a^2}$ *for some* $\delta > 0$. *Let* U^* *be the global optimal solution, and define* $\Delta = U^*(U^*)^\top - \Phi^*(\Phi^*)^\top$. *Then with high probability*

$$\|\Delta\|_{1,\theta} \leq \frac{C_0}{\delta} \left(1 + \left(\max_{1\leq a\leq k} \frac{B_{aa}}{H_a^2} \|f\|_1\right)\right) (\sqrt{n\|f\|_1} + n)$$

Outline

Analysis on a quartic-quadratic optimization problem

Definition 1 (Model Problem)

Suppose matrix $A \in \mathbb{C}^{n\times n}$ is Hermitian and $\beta > 0$ is a constant. We consider the following minimization problem.

$$\min_{z\in\mathbb{C}^n} f(z) := \frac{1}{2} z^* A z + \frac{\beta}{2} \sum_{k=1}^{n} |z_k|^4, \quad \text{s.t. } \|z\| = 1.$$

Example: Non-rotating BEC Problem

The ground state of non-rotating Bose-Einstein Condensation (BEC) problem is usually defined as the minimizer of the following dimensionless energy functional

$$E(\phi) := \int_{\mathbb{R}^d} \left[\frac{1}{2} |\nabla \phi(\mathbf{x})|^2 + V(\mathbf{x}) |\phi(\mathbf{x})|^2 + \frac{\beta}{2} |\phi(\mathbf{x})|^4 \right] \mathrm{d}\,\mathbf{x},$$

where $d = 1, 2, 3$ is the dimension, $V(\mathbf{x})$ denotes the potential and $\beta \in \mathbb{R}$ is the interaction coefficient. We also need the wave function to be normalized: $\|\phi\|_{L^2(\mathbb{R}^d)} = 1$.

Batch normalization (BN) from deep learning

- Given weight vector w, the output x from the previous layer
- Batch normalization transform on $z := w^\top x$

$$BN(z) = \frac{z - \mathbf{E}[z]}{\sqrt{Var[z]}} = \frac{w^\top (x - \mathbf{E}[x])}{\sqrt{w^\top R_{xx} w}} = \frac{u^\top (x - \mathbf{E}[x])}{\sqrt{u^\top R_{xx} u}}$$

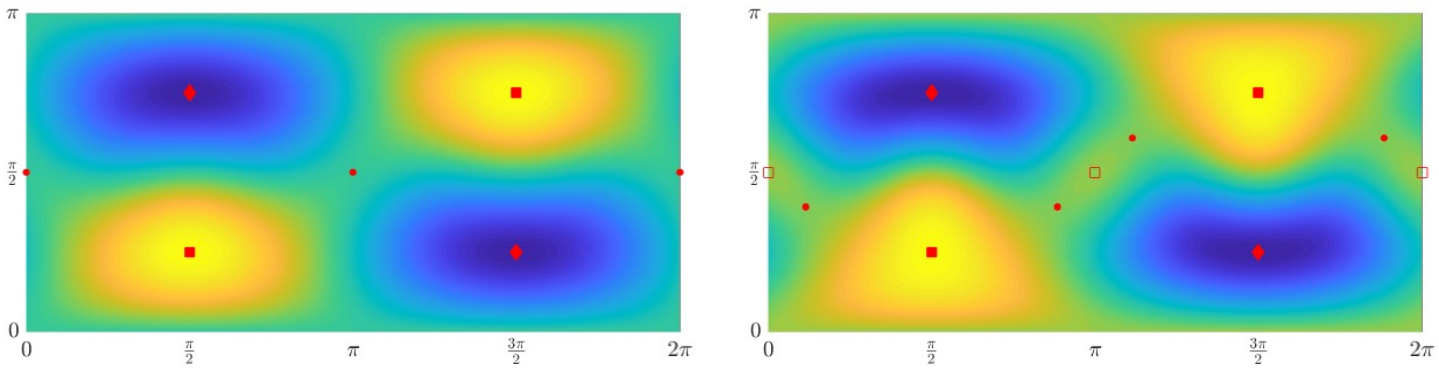
  where $u = w/\|w\|$, $\mathbf{E}[x]$ and R_{xx} are the mean and covariance of x.
- Note that $BN(w^\top x) = BN(u^\top x)$, then the wight vector satisfies

$$w \in S^{n-1}$$

  where S^{n-1} is is the $(n-1)$-dimensional sphere in $\mathbb{R}^n$.
- Deep networks with multiple layers and multiple units per layer

$$\min_{\mathcal{X} \in \mathcal{M}} \mathcal{L}(X), \text{ s.t.} \quad \mathcal{M} = S^{n_1 - 1} \times \cdots \times S^{n_m - 1} \times \mathbb{R}^l$$

Landscape of the objective function



(a) $\beta = 0.25$ (b) $\beta = 0.75$

(c) $\beta = 1.25$ (d) $\beta = 3.25$

The red point marker: saddle points. Local and global minima are indicated by non-filled and filled diamond markers. The location of local and global maxima is marked by non-filled and filled squares.

Geometric Analysis In Real Case

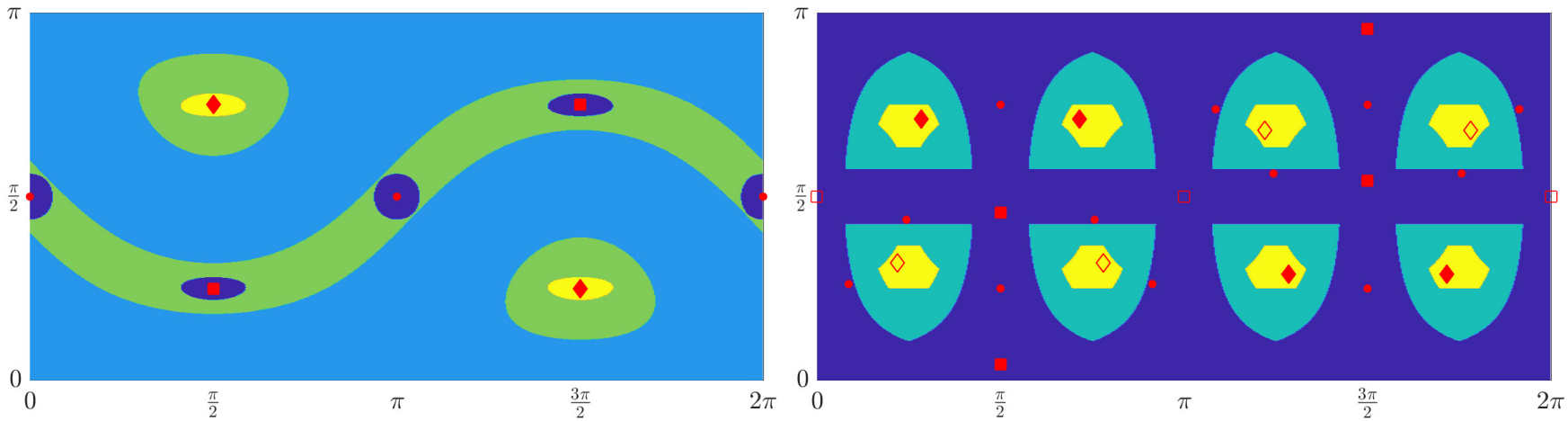
Theorem 8

Suppose that the coefficient β satisfies $\beta \geq \frac{8n}{n-1}(1+\gamma)\rho n^{3/2}$ for some given $\gamma > 0$. Then, the function f has the $(C_\gamma \rho, \frac{\gamma}{\sqrt{2}}\rho, C_\gamma \rho)$-strict-saddle property with $C_\gamma := \frac{4}{n-1}(1+\gamma)n^{3/2} - 1$.

Three Regions

1. (Strong convexity). $\mathcal{R}_1 = \{z \in \mathbb{S}^{n-1} : \max_{1\leq k\leq n} |z_k^2 - 1/n| \leq 1/2n\}$.
2. (Large gradient). $\mathcal{R}_2 = \{z \in \mathbb{S}^{n-1} : \max_{1\leq k\leq n} |z_k^2 - 1/n| \geq 1/2n, \min_{1\leq k\leq n} z_k^2 \geq 1/12n\}$.
3. (Negative curvature). $\mathcal{R}_3 = \{z \in \mathbb{S}^{n-1} : \min_{1\leq k\leq n} z_k^2 \leq 1/12n\}$.

Geometric Analysis In Real Case



(a) $\beta = 0.2$ (b) $\beta = 3.75$

Figure (a): The overlap of the sets $\mathcal{R}_1$–$\mathcal{R}_2$ and $\mathcal{R}_2$–$\mathcal{R}_3$ is shown in green. The set $\mathcal{R}_1$ is the union of the yellow and the two surrounding green areas, while $\mathcal{R}_2$ is the union of all green and light blue areas. The region $\mathcal{R}_3$ is the union of the dark blue sets and the enclosing green area. Figure (b): the (disjoint) yellow, turquoise, and dark blue areas directly correspond to the sets $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$, respectively. Non-filled and filled diamond markers are used for local and global minima. Local and global maxima are marked by non-filled and filled squares.

Geometric Analysis In Real Case

Corollary 9

If $\beta > 4\rho n^2$, the problem has at least 2^n local minima. Furthermore, if $\beta > \frac{18n^3}{n-1}\rho$, then the problem has exactly 2^n local minima

Theorem 10

Suppose that $\beta > \frac{18n^3}{n-1}\rho$. Then, it follows

$$f(\mathbf{y}) - \min_{\mathbf{z}\in S^{n-1}} f(\mathbf{z}) \leq \frac{1}{18n} \cdot \left[\min_{\mathbf{z}\in S^{n-1}} f(\mathbf{z}) - \lambda_n(A) \right], \tag{2}$$

for all local minimizer $\mathbf{y} \in S^{n-1}$ where $\lambda_n(A)$ denotes the smallest eigenvalue of the matrix A.

Estimation of the Kurdyka-Łojasiewicz Exponent

- Find the largest $\theta \in (0, \frac{1}{2}]$ such that for all stationary points $\mathbf{z}$, the Łojasiewicz inequality,

$$|f(\mathbf{y}) - f(\mathbf{z})|^{1-\theta} \leq \eta_{\mathbf{z}} \|\operatorname{grad} f(\mathbf{y})\|, \quad \forall\, \mathbf{y} \in B(\mathbf{z}, \delta_{\mathbf{z}}) \cap \mathbb{CS}^{n-1}, \tag{3}$$

  holds with some constants $\delta_{\mathbf{z}}, \eta_{\mathbf{z}} > 0$.
- Let $A = \operatorname{diag}(\mathbf{a}) \in \mathbf{C}^{n\times n}$, $\mathbf{a} \in \mathbb{R}^n$, be a diagonal matrix. Then, the largest KL exponent is at least $\frac{1}{4}$.
- Suppose $A \in \mathbb{R}^{n\times n}$ is a symmetric matrix and $\mathbf{z}$ is a stationary point satisfying

$$H := A + 2\beta \operatorname{diag}(|\mathbf{z}|^2) - 2\lambda I \succeq 0,$$

  where $\lambda = \mathbf{z}^* \nabla_{\mathbf{z}} f(\mathbf{z}) = \frac{1}{2}\mathbf{z}^* A\mathbf{z} + \beta\|\mathbf{z}\|_4^4$. Then, the largest KL exponent at $\mathbf{z}$ is at least $\frac{1}{4}$.

Contact Information

Many Thanks For Your Attention!

- 北大课程：大数据分析中的算法，华文慕课回放
  http://bicmr.pku.edu.cn/~wenzw/bigdata2020.html
- 教材：刘浩洋, 户将, 李勇锋，文再文，最优化计算方法http://bicmr.pku.edu.cn/~wenzw/optbook.html
- Looking for Ph.D students and Postdoc
  Competitive salary as U.S and Europe
- http://bicmr.pku.edu.cn/~wenzw
- E-mail: wenzw@pku.edu.cn
- Office phone: 86-10-62744125