中国科学: 数学 2024 年 第 54 卷 第 4 期: 671～698
SCIENTIA SINICA Mathematica
论 文

《中国科学》杂志社
SCIENCE CHINA PRESS
CrossMark click for updates

基于预测向心加速的生成对抗网络训练

李科科[1], 杨新民[2*], 张柯[2]

1. 四川大学数学学院, 成都 610065;
2. 重庆国家应用数学中心, 重庆 401331
E-mail: likeke135@163.com, xmyang@cqnu.edu.cn, 20208023@cqnu.edu.cn

收稿日期: 2022-12-07; 接受日期: 2023-06-26; 网络出版日期: 2024-02-07; * 通信作者
国家自然科学基金 (批准号: 11991020 和 11991024) 资助项目

摘要 为缓解生成对抗网络 (generative adversarial networks, GAN) 训练过程中的极限循环行为, 本文受向心加速算法及 Liang 和 Stokes (2019) 的修正的预测方法 (modified predictive method, MPM) 的启发, 基于对匀速圆周运动的几何观察提出了预测向心加速算法 (predictive centripetal acceleration algorithm, PCA). 首先, 在二元线性博弈 (特殊的 GAN) 上证明了 PCA 的最后一次迭代收敛性. 然后, 将 PCA 分别与随机梯度下降 (stochastic gradient descent, SGD) 算法和自适应性矩估计 (adaptive moment estimation, Adam) 算法结合, 提出了随机 PCA (stochastic PCA, SPCA) 和 PCA-Adam 用于实际训练 GAN. 最后, 在二元线性博弈、多元 Gauss 分布以及 CIFAR10 和 CelebA 数据集上的实验分别验证了所提出算法的有效性.

关键词 生成对抗网络 预测向心加速算法 二元线性博弈 收敛性

MSC (2020) 主题分类 90C26, 90C29, 90C30, 90C46

1 引言

生成对抗网络 (generative adversarial networks, GAN) [24] 是目前深度学习领域的研究热点和主要方向之一, 在人脸生成、视频生成、生物芯片数据处理、医学图像处理和隐私保护等方面均有广泛应用 (参见文献 [2,9,15,37,50,51,56,60]). GAN 将学习训练数据集背后分布规律的问题建模为生成器 (G 网络) 和判别器 (D 网络) 之间的一个二人零和博弈. G 网络将随机噪声的采样 z 映射到训练样本空间, 目的是近似真实数据的分布; D 网络试图辨别输入的样本来自训练数据集还是由 G 网络生成. 该零和博弈期望达到均衡时 G 网络能学习到真实数据的分布, 并且可以生成使 D 网络难以辨别的样本 (参见文献 [12, 23, 24]).

GAN 虽然是渐近一致的, 且有不需要使用 Markov 链、不依赖任何先验假设和生成类似真实样本方式简单等优点 [23, 24], 但无论是在理论上还是在实际应用中依然存在许多问题 [23, 47, 54]. 训练算

英文引用格式: Li K K, Yang X M, Zhang K. Training GAN with predictive centripetal acceleration (in Chinese). Sci Sin Math, 2024, 54: 671–698, doi: 10.1360/SSM-2022-0237

© 2024《中国科学》杂志社

法是 GAN 研究的核心内容之一[58], 梯度类算法是训练机器学习任务的主要方法[11], 求解可微二人零和博弈最简单的梯度类算法是梯度下降上升算法 (gradient descent ascent algorithm, GDA)[55]. 训练 GAN 本质是用优化算法找二人零和博弈的均衡点, 但理论上 GAN 训练过程的全局收敛性悬而未决[23,47]. 特别地, 近期文献 [47] 针对 GAN 训练过程提出的公开问题是: 何种条件下能够证明 GAN 训练过程是全局收敛的? 事实上, 早在 2016 年 Goodfellow[23] 就注意到, 对一部分博弈, 如凸凹博弈, 在平均迭代意义下 GDA 的确收敛 (参见文献 [45,61,62]); 但对另一部分博弈, 如二元线性博弈, GDA 在最后一次迭代意义下不收敛. 然而, 文献 [12,39] 指出, 实际训练 GAN 需要改变算法输出来作相应的分析, 遍历平均不再适用, 因此需要考虑算法在最后一次迭代意义下的收敛性. 最近, Chavdarova 等[10] 指出, 在最后一次迭代意义下 GDA 求解二元线性博弈会以充分小的步长绕最优解循环, 否则发散. 进一步地, GDA 训练一般的 GAN 在最后一次迭代意义下理论上是否收敛, 该问题依然没有得到回答 (参见文献 [23]). 近年来, 在一些特殊博弈上讨论 GDA 最后一次迭代意义下收敛性的主要研究如下. Liang 和 Stokes[33] 在光滑性和局部强凸强凹假设下证明了 GDA 以线性速率收敛到稳定的局部 Nash 均衡点. Lin 等[34] 先指出对凸凹博弈, 虽然文献 [45] 得到了 GDA 平均迭代意义下的收敛性, 但一般情形下 GDA 可能收敛到极限环甚至发散; 然后在最后一次迭代意义下得到了双时间尺度 GDA 求解非凸凹博弈时的收敛性结果, 该结果表明双时间尺度 GDA 可有效找到函数 $\Phi(\cdot)=\max_{y\in\mathcal{Y}} f(\cdot,y)$ 的一个稳定点, 揭示了 GDA 在应用中的实际性能. 此外, Flokas 等[18] 证明了在多策略隐二元线性博弈中, 连续时间 GDA 动态过程的轨迹在安全初始化假设下微分同胚于 Poincaré 递归流的一个轨迹. Bailey 等[6] 指出, 在对抗博弈中使用交替梯度上升下降算法 (alternating GDA, AGDA) 时, 局中人的策略是有界和循环的, 并指出在二元线性博弈中当步长满足一定条件时, 连续时间 AGDA 动态过程满足 Poincaré 递归. Fiez 等[17] 指出任意周期性二元线性博弈连续时间 GDA 动态过程满足 Poincaré 递归且是保体积的, 此外其轨迹是有界的.

实践中成功训练 GAN 仍然非常困难[23,25,32], 且基于 GDA 的 GAN 训练过程常常震荡且容易出现极限循环行为[8,12,48,55], 因此文献 [11,25,41] 都直接用 “臭名昭著地难 (notoriously hard)” 等词来形容训练 GAN 的困难程度. 近年来, 已有部分工作基于 GDA 提出的新算法试图解决其在 GAN 训练过程中出现的问题, 如迭代过程的震荡和极限循环行为等[12,48]. 特别地, Hsieh[28] 提出了关于二人零和博弈的公开问题—如何使用优化技巧消除二人零和博弈中的极限循环行为? 近期, 学者们提出了试图缓解 GAN 训练过程中的极限循环行为的方法, 主要分为基于博弈理论的方法和基于 GDA 的方法. 基于博弈理论的方法的研究主要如下. Mescheder 等[41] 引入了共识优化算法 (consensus optimization algorithm, CO), 在适当假设下得到了其迭代会收敛到局部 Nash 均衡点附近, 并借助 GAN 学习 Gauss 核实验说明了 CO 的有效性. Abernethy 等[1] 利用二阶方法—Hamilton 梯度下降算法 (Hamiltonian gradient descent algorithm, HGD) 来缓解循环行为, 在适当假设下获得了 HGD 在凸凹博弈上的线性收敛性. 基于 GDA 方法的研究主要如下. 2018 年国际数学家大会 Nevanlinna 奖获得者 Daskalakis 及其合作者在文献 [12] 中试图用乐观梯度下降上升算法 (optimistic GDA, OGDA) 缓解 GAN 训练过程中的极限循环行为. 他们先分析了 OGDA 在二元线性博弈上最后一次迭代意义下的收敛率, 然后将 OGDA 融入 Adam[31] 提出 optimistic-Adam, 并在 Wasserstein GAN[4] 上进行了数值实验以凸显 optimistic-Adam 性能优于 Adam. 受文献 [12] 的启发, Mertikopoulos 等[39] 将乐观的 (optimistic) 解释为一个 “外梯度” 步, 即将训练过程沿当前梯度进行外推, 并在相依性假设和 Lipschitz 假设下建立了 OGDA 收敛性理论, 随后提出 optimistic-Adam 的另一版本, 借助 DCGAN 进行了实验以说明其算法的有效性. Peng 等[48] 指出训练 GAN 易受迭代循环行为的影响, 并基于对匀速圆周运动的直觉观察, 提出了同步向心加速算法 (gradient descent with simultaneous centripetal acceleration algorithm,

Grad-SCA) 和交替向心加速算法 (gradient descent with alternating centripetal acceleration algorithm, Grad-ACA) 试图缓解极限循环行为, 在适当假设下得到 Grad-SCA 和 Grad-ACA 在二元线性博弈上的线性收敛率, 并借助 GAN 学习 Gauss 核实验验证了所提出算法 RMSprop-ACA 的有效性.

Mertikopoulos 等 [39] 认为目前对抗性学习远没有被很好地理解, 尤其是 GAN, 而且 GAN 很难训练, 标准优化方法往往无法收敛到合理的解. 从目前 GAN 训练算法研究的现状来看, 由于直接对 GAN 研究其训练算法的收敛性非常困难 [47,53,54], 因此在 GAN 的一些简化模型上提出能够保证收敛性的新算法, 并用其提高实际应用中 GAN 生成样本的质量, 此类研究非常有意义, 详见文献 [11, 12, 20, 26, 33, 39, 48, 52]. 鉴于此, 本文针对文献 [28] 提出的公开问题 (如何使用优化技巧消除二人零和博弈中的极限循环行为), 结合 GAN 作初步探究. 注意到文献 [12,39,43,48] 在缓解极限循环行为时使用了近似向心加速项, 自然地, 存在以下问题: 如何构造精确向心加速项, 如何使用它缓解极限循环行为, 效果又怎样? 回答上述问题是本文的主要研究工作. 具体地, 本文试图缓解 GAN 训练过程中的极限循环行为, 基于对匀速圆周运动的直觉观察并结合 Lookahead 方法 [11,63] 构造了一个精确向心加速项, 使用该精确向心加速项对 GDA 进行校正, 从而获得预测投影向心加速算法 (predictive projection centripetal acceleration algorithm, PPCA). 首先, 在二元线性博弈上发现 GDA 当前步符号梯度与下一步符号梯度分别构成直角三角形的一条直角边和斜边, 进而在几何上解释了文献 [12, 命题 1]; 并得到对二元线性博弈即使是在矩阵满秩条件下 GDA 也不收敛的结果. 其次, 由 PPCA 得到其简化版本—PCA, 并在二元线性博弈上获得 PCA 在最后一次迭代意义下的线性收敛率. 与文献 [12, 定理 1] 相比, 为获得近似度相同的近似 Nash 均衡点, PCA 所需迭代界小于该定理 1 的迭代界. 然后, 对比 PCA 与 GDA 及其使用近似向心加速项校正的方法在缓解极限循环行为方式上的不同, 并通过二元线性博弈上的对比实验说明了 PCA 缓解循环行为更有效. 最后, 将 PCA 分别与 SGD 和 Adam [31] 结合提出了 SPCA 和 PCA-Adam, 以文献 [12] 用 GAN 学习多元 Gauss 分布实验为例展示了 SPCA 缓解极限循环行为效果更稳定. 此外, 借助 DCGAN 进行的数值实验表明 PCA-Adam 训练 GAN 性能优于 Extra-Adam (参见文献 [20]).

2 预备知识

用 $\mathbb{R}^n$ 表示 n 维 Euclid 空间, $\|\theta\|_2$ 表示 $\mathbb{R}^n$ 中向量 θ 的 2 范数, $\prod_{(\theta)}(\phi)$ 表示向量 $\phi\in\mathbb{R}^n$ 在向量 $\theta\in\mathbb{R}^n$ 上的投影, $\nabla V(\cdot)$ 表示函数 $V(\cdot)$ 的梯度算子. 在二人零和博弈中, 一个局中人的损失函数是 V, 另一个局中人的损失函数是 $-V$, 其中 $V(\cdot,\cdot):\Theta\times\Phi\to\mathbb{R}$, $\Theta\times\Phi\subseteq\mathbb{R}^n\times\mathbb{R}^n$. 损失函数 $V(\cdot,\cdot)$ 将博弈局中人的策略 $(\theta,\phi)\in\Theta\times\Phi$ 映射到一个实数值, 它代表着局中人 ϕ 的收益, 同时也是局中人 θ 的损失. 称局中人 ϕ (试图最大化 V) 为最大化局中人, 另一个局中人 θ 为最小化局中人. 在经典博弈论中, 二人零和博弈可表示为如下形式:

$$\min_{\theta\in\Theta}\max_{\phi\in\Phi}V(\theta,\phi). \tag{2.1}$$

显然, GAN 损失函数 $\min_G\max_D V(G,D)=\mathrm{E}_{x\sim\mathbb{P}_{\mathrm{data}}}[\log D(x)]+\mathrm{E}_{z\sim\mathbb{P}_z}[\log(1-D(G(z)))]$ 可写为博弈 (2.1) 的形式, 其中 G、D、$\mathbb{P}_{\mathrm{data}}$ 和 $\mathbb{P}_z$ 分别为生成器网络、判别器网络、真实数据分布和噪声分布. 本文主要关注可微二人零和博弈, 即局中人损失函数 $V(\cdot,\cdot)$ 在 $\Theta\times\Phi\subseteq\mathbb{R}^n\times\mathbb{R}^n$ 上是可微的. 下面介绍博弈均衡点的相关概念.

定义 2.1 [29,30] 如果对任意的 $(\theta,\phi)\in\Theta\times\Phi$, 都有 $V(\theta^\star,\phi)\leqslant V(\theta^\star,\phi^\star)\leqslant V(\theta,\phi^\star)$, 则点 $(\theta^\star,\phi^\star)$ 称为函数 $V(\cdot,\cdot)$ 的鞍点 (Nash 均衡点).

定义 2.2 [29,30] 如果存在 $\delta>0$, 使得对任意满足 $\|\theta-\theta^*\|_2\leqslant\delta$ 和 $\|\phi-\phi^*\|_2\leqslant\delta$ 的 $(\theta,\phi)\in\Theta\times\Phi$, 都有 $V(\theta^\star,\phi)\leqslant V(\theta^\star,\phi^\star)\leqslant V(\theta,\phi^\star)$, 则点 $(\theta^\star,\phi^\star)$ 称为函数 $V(\cdot,\cdot)$ 的局部鞍点 (局部 Nash 均衡点).

定义 2.3 [1] 如果 $\nabla V(\theta,\phi)=0$, 则点 $(\theta,\phi)\in\mathbb{R}^n\times\mathbb{R}^n$ 称为函数 $V(\cdot,\cdot)$ 的临界点.

Nouiehed 等 [46] 指出, 当函数 $V(\cdot,\cdot)$ 关于第一变元下半连续凸、关于第二变元上半连续凹时, 由 Von Neumann 定理可知, 在约束集合紧性和凸性假设下此类博弈的 Nash 均衡点总是存在的, 且有许多算法可找到 Nash 均衡点 (具体算法详见文献 [22, 59]). 然而, 当函数 $V(\cdot,\cdot)$ 关于第一变元非凸、关于第二变元非凹时, Jin [29] 和 Jin 等 [30] 表示找到一般情形下此类博弈的全局 Nash 均衡点是 NP (non-deterministic polynomial-time) 难的. 事实上, 此类问题的一种特殊情形, 即找一般非凸问题的全局极小值点也是 NP 难的 [35], 甚至对一般非凸非凹博弈找到其局部 Nash 均衡点也是 NP 难的 [46]. 此外, Jin [29] 和 Jin 等 [30] 给出了关于鞍点存在性的一个有趣事实, 即一般非凸优化问题中全局最优解是局部最优解, 因而局部最优总是存在的, 然而对于非凸非凹博弈其局部或全局 Nash 均衡点可能不存在. 特别地, Farnia 和 Ozdaglar [16] 指出对 GAN 这类博弈其 Nash 均衡点可能不存在. 因此, Jin 等 [30] 指出二人零和博弈是一项极具挑战性的研究课题, 值得从非凸优化中独立出来研究. 事实上, GAN 还是一个多目标优化问题. Azizian 等 [5] 表示 GAN 是涉及多参与者和多个目标的优化问题, 博弈是单目标优化问题的一种推广, 但出现了单目标所没有的新问题. Chavdarova 等 [11] 指出多目标优化在深度学习中应用很成功的一个例子是 GAN.

3 预测向心加速算法 (PCA)

本节首先简要介绍缓解循环行为的相关算法, 然后介绍 PCA 的构造过程和迭代格式, 最后通过数值实验验证 PCA 缓解循环行为的有效性.

3.1 相关算法

除特殊说明外, 用 $\nabla_\theta V(\theta,\phi)$ 和 $\nabla_\phi V(\theta,\phi)$ 表示 $V(\theta,\phi)$ 分别关于 θ 和 ϕ 的偏梯度, α、α_1、α_2、β、β_1、β_2 和 γ 为以下相应算法的参数. 首先, 回顾训练 GAN 常用的梯度下降上升算法 (GDA [6,41,48]) 及其交替版本—AGDA [6,41,48] 如下:

$$\theta_{t+1}=\theta_t-\alpha\nabla_\theta V(\theta_t,\phi_t),\quad \phi_{t+1}=\phi_t+\alpha\nabla_\phi V(\theta_t,\phi_t); \tag{GDA}$$

$$\theta_{t+1}=\theta_t-\alpha\nabla_\theta V(\theta_t,\phi_t),\quad \phi_{t+1}=\phi_t+\alpha\nabla_\phi V(\theta_{t+1},\phi_t). \tag{AGDA}$$

然后, 介绍修正的预测方法 (modified predictive method, MPM [33])、OGDA [13,43] (或 optimistic mirror descent algorithm, OMD [12,33,48]) 和广义乐观梯度下降上升算法 (generalized optimistic GDA, GOGDA [43]):

$$\begin{aligned}&\text{预测步}:\quad \theta_{t+1/2}=\theta_t-\gamma\nabla_\theta V(\theta_t,\phi_t),\quad \phi_{t+1/2}=\phi_t+\gamma\nabla_\phi V(\theta_t,\phi_t),\\&\text{更新步}:\quad \theta_{t+1}=\theta_t-\beta\nabla_\theta V(\theta_{t+1/2},\phi_{t+1/2}),\quad \phi_{t+1}=\phi_t+\beta\nabla_\phi V(\theta_{t+1/2},\phi_{t+1/2});\end{aligned} \tag{MPM}$$

$$\begin{aligned}&\theta_{t+1}=\theta_t-2\alpha\nabla_\theta V(\theta_t,\phi_t)+\alpha\nabla_\theta V(\theta_{t-1},\phi_{t-1}),\\&\phi_{t+1}=\phi_t+2\alpha\nabla_\phi V(\theta_t,\phi_t)-\alpha\nabla_\phi V(\theta_{t-1},\phi_{t-1});\end{aligned} \tag{OGDA}$$

$$\begin{aligned}&\theta_{t+1}=\theta_t-(\alpha+\beta)\nabla_\theta V(\theta_t,\phi_t)+\beta\nabla_\theta V(\theta_{t-1},\phi_{t-1}),\\&\phi_{t+1}=\phi_t+(\alpha+\beta)\nabla_\phi V(\theta_t,\phi_t)-\beta\nabla_\phi V(\theta_{t-1},\phi_{t-1}).\end{aligned} \tag{GOGDA}$$

最后, 介绍同步向心加速算法 (Grad-SCA [48]) 和交替向心加速算法 (Grad-ACA [48]) 如下:

$$\begin{aligned} &G_\theta = \nabla_\theta V(\theta_t,\phi_t) + \frac{\beta_1}{\alpha_1}(\nabla_\theta V(\theta_t,\phi_t) - \nabla_\theta V(\theta_{t-1},\phi_{t-1})), \theta_{t+1} = \theta_t - \alpha_1 G_\theta,\\ &G_\phi = \nabla_\phi V(\theta_t,\phi_t) + \frac{\beta_2}{\alpha_2}(\nabla_\phi V(\theta_t,\phi_t) - \nabla_\phi V(\theta_{t-1},\phi_{t-1})), \phi_{t+1} = \phi_t + \alpha_2 G_\phi; \end{aligned} \quad \text{(Grad-SCA)}$$

$$\begin{aligned} &G_\theta = \nabla_\theta V(\theta_t,\phi_t) + \frac{\beta_1}{\alpha_1}(\nabla_\theta V(\theta_t,\phi_t) - \nabla_\theta V(\theta_{t-1},\phi_{t-1})), \theta_{t+1} = \theta_t - \alpha_1 G_\theta,\\ &G_\phi = \nabla_\phi V(\theta_{t+1},\phi_t) + \frac{\beta_2}{\alpha_2}(\nabla_\phi V(\theta_{t+1},\phi_t) - \nabla_\phi V(\theta_t,\phi_{t-1})), \phi_{t+1} = \phi_t + \alpha_2 G_\phi. \end{aligned} \quad \text{(Grad-ACA)}$$

为易于理解, 文献 [48, 图 1] 直观地展示了向心加速方法的思想. 首先, 文献 [48] 考虑了质点绕圆心的匀速圆周运动模型, 令 ∇V_t 表示 t 时刻的瞬时速度, 则向心加速项 (向心力) $\lim_{\delta t\to 0}(\nabla V_{t+\delta t}-\nabla V_t)/\delta t$ 指向圆心. 然后, 文献 [48] 认为绕 Nash 均衡点打圈的行为可能和绕原点的匀速圆周运动类似, 向心加速项提供了一个方向, 沿着该方向迭代可更快地接近目标. 最后, 文献 [48] 利用近似向心加速项 $(\nabla V_{t+1} - \nabla V_t)$ 校正 GDA 获得算法 Grad-SCA.

自然地, 一个想法是, 在几何上如何理解近似向心加速方法, 如何构造一个直接指向圆心的方向来缓解极限循环行为, 相比近似向心加速项效果又怎样?

基于上述问题, 本文同样考虑质点绕圆心的匀速圆周运动模型, 如图 1 所示, 令 ∇V_t 表示 t 时刻的瞬时速度. 受 OGDA、Grad-SCA、GOGDA 和 MPM 的启发, 本文首先执行一个预测步 $t+1/2$ 得到 $t+1/2$ 时刻的瞬时速度 $\nabla V_{t+1/2}$, 此时还可获得 t 时刻的近似向心加速项 $(\nabla V_{t+1/2} - \nabla V_t)$. 其次, 将 t 时刻的近似向心加速项 $(\nabla V_{t+1/2} - \nabla V_t)$ 投影到 t 时刻的瞬时速度 ∇V_t 上, 可构造出一个精确向心加速项—预测投影向心加速项 $(\nabla V_{t+1/2} - \nabla V_t) - \prod_{(\nabla V(\theta_t,\phi_t))}(\nabla V(\theta_{t+1/2},\phi_{t+1/2}) - \nabla V(\theta_t,\phi_t))$, 该项直接指向圆心 (见图 1). 然后, 本文同样认为绕均衡点打圈的行为与绕圆心的匀速圆周运动类似, 因此与近似向心加速方向相比, 沿着预测投影向心加速方向迭代可以更快速地接近目标. 最后, 用预测投影向心加速项校正 GDA 可获得 PPCA. 值得注意的是, 在上述过程中若使用近似向心加速项 $(\nabla V_{t+1/2} - \nabla V_t)$ 直接对 GDA 作不同程度的校正, 可分别得到 OGDA、GOGDA 和 Grad-SCA. 下面

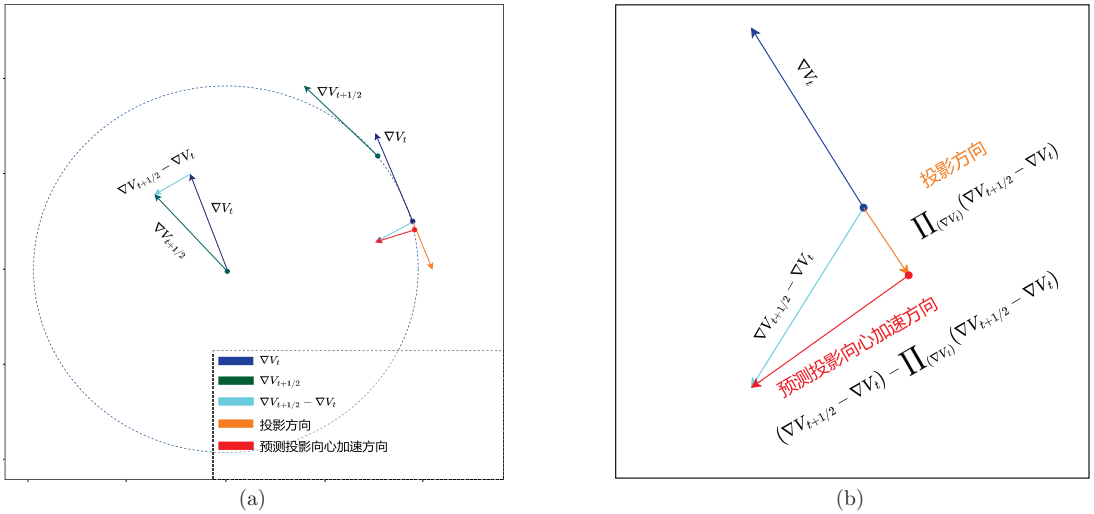


图 1 (网络版彩图) (a) 预测投影向心加速算法的示意图; (b) (a) 中 $\nabla V_{t+1/2} - \nabla V_t$ 在 ∇V_t 上的投影 $\prod_{(\nabla V_t)}(\nabla V_{t+1/2} - \nabla V_t)$ 和预测投影向心加速项 $(\nabla V_{t+1/2} - \nabla V_t) - \prod_{(\nabla V_t)}(\nabla V_{t+1/2} - \nabla V_t)$ 的一个放大与描述

本文给出 PPCA, 其迭代格式如下:

$$\begin{aligned}
&\text{预测步}: \quad \theta_{t+1/2}=\theta_t-\gamma\nabla_\theta V(\theta_t,\phi_t), \quad \phi_{t+1/2}=\phi_t+\gamma\nabla_\phi V(\theta_t,\phi_t);\\
&\text{梯度步}: \quad \begin{pmatrix}\theta_{t+1}\\ \phi_{t+1}\end{pmatrix}=\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}+\alpha\nabla\bar{V}(\theta_t,\phi_t)+\beta\Big((\nabla\bar{V}(\theta_{t+1/2},\phi_{t+1/2})-\nabla\bar{V}(\theta_t,\phi_t))\\
&\qquad\qquad -\prod\nolimits_{(\nabla\bar{V}(\theta_t,\phi_t))}(\nabla\bar{V}(\theta_{t+1/2},\phi_{t+1/2})-\nabla\bar{V}(\theta_t,\phi_t))\Big),
\end{aligned}\tag{3.1}$$

其中 $\nabla\bar{V}(\theta_t,\phi_t):=(\,-\nabla_\theta V(\theta_t,\phi_t)^{\mathrm{T}},\nabla_\phi V(\theta_t,\phi_t)^{\mathrm{T}}\,)^{\mathrm{T}}$.

3.2 算法 PCA

对二元线性博弈的研究可以揭示梯度在博弈中的本质问题 [7]. 此外, 二元线性博弈常被作为理论上讨论 GAN 训练过程出现问题原因的简单模型 [5] 以及分析与理解新算法和新技巧的重要简单示例 [12, 19–21, 33, 62]. 事实上, 在二元线性博弈中能够体现出 GDA 在 GAN 训练过程中存在的困难, 同时, 二元线性博弈可代表一些简单形式的 GAN, 如生成器和判别器均为线性函数的 Wasserstein GAN (参见文献 [12, 20, 40, 49]). 此外, 自 2016 年 Goodfellow [23] 分析了二维二元线性博弈上 GDA 连续时间轨迹为圆的方程之后, 基于 GDA 提出新算法的研究几乎都在二元线性博弈上讨论了所提出算法的有效性, 如近期文献 [7, 12, 13, 20, 22, 33, 40, 62] 以及文献 [3, 5, 6, 11, 36, 42, 43, 48] 和 [17, 19, 61] 等.

遵循上述文献的研究思路, 本小节先讨论 PPCA 在如下二元线性博弈上的一些性质:

$$\min_{\theta\in\mathbb{R}^n}\max_{\phi\in\mathbb{R}^n} V(\theta,\phi)=\theta^{\mathrm{T}}A\phi+b^{\mathrm{T}}\theta+c^{\mathrm{T}}\phi, \quad A\in\mathbb{R}^{n\times n}, \quad b,c\in\mathbb{R}^n,\tag{3.2}$$

其中 $\theta,\phi\in\mathbb{R}^n$ 蕴涵着局中人 θ 和 ϕ 有相同的决策空间. 假设矩阵 A 满秩, 由博弈 (3.2) 的一阶最优性条件可知,

$$A\phi^*+b=0, \quad A^{\mathrm{T}}\theta^*+c=0,$$

其中 (θ^*,ϕ^*) 是博弈 (3.2) 的临界点. 借鉴文献 [42] 分析二元线性博弈 Nash 均衡存在性时所用技巧, 此时博弈 (3.2) 的目标函数 $V(\theta,\phi)$ 可写为 $V(\theta,\phi)=(\theta-\theta^*)^{\mathrm{T}}A(\phi-\phi^*)+$常数. 借鉴文献 [21,42,48,61,62] 处理二元线性博弈的思想, 不失一般性, 本文也作变换 $\theta=\theta-\theta^*$, $\phi=\phi-\phi^*$. 故博弈 (3.2) 可写为

$$\min_{\theta\in\mathbb{R}^n}\max_{\phi\in\mathbb{R}^n} V(\theta,\phi)=\theta^{\mathrm{T}}A\phi, \quad A\in\mathbb{R}^{n\times n}.\tag{3.3}$$

显然, 博弈 (3.3) 只有一个 Nash 均衡点 $(0,0)$. 此外, Zhang 和 Yu [61, 62] 指出博弈 (3.2) 中线性项在分析时不涉及本质问题, 因而可以在博弈 (3.2) 中直接令 $b=c=0$.

下面给出 PPCA 在二元线性博弈上迭代格式中投影项为 0 的结果, 该结果极大地简化了 PPCA 的迭代格式, 并对提出算法 PCA 及分析其在二元线性博弈 (3.3) 上的收敛性至关重要.

引理 3.1 如果二元线性博弈 (3.3) 中矩阵 A 满秩, 且 $\bar{V}(\theta_t,\phi_t)\neq 0$, 则 PPCA 迭代格式 (3.1) 中投影项 $\prod_{(\nabla\bar{V}(\theta_t,\phi_t))}(\nabla\bar{V}(\theta_{t+1/2},\phi_{t+1/2})-\nabla\bar{V}(\theta_t,\phi_t))=0$.

证明 由投影的定义可知,

$$\prod\nolimits_{(\nabla\bar{V}(\theta_t,\phi_t))}(\nabla\bar{V}(\theta_{t+1/2},\phi_{t+1/2})-\nabla\bar{V}(\theta_t,\phi_t))$$

$$= \frac{\langle(\nabla\bar{V}(\theta_{t+1/2},\phi_{t+1/2})-\nabla\bar{V}(\theta_t,\phi_t)),\nabla\bar{V}(\theta_t,\phi_t)\rangle}{\|\nabla\bar{V}(\theta_t,\phi_t)\|_2^2}\nabla\bar{V}(\theta_t,\phi_t), \tag{3.4}$$

其中 $\langle\cdot,\cdot\rangle$ 表示 $\mathbb{R}^{2n}$ 空间的内积.

注意到对二元线性博弈 (3.3), 有

$$\begin{aligned}\nabla\bar{V}(\theta_{t+1/2},\phi_{t+1/2})-\nabla\bar{V}(\theta_t,\phi_t) &= \begin{pmatrix}-A\phi_{t+1/2}\\ A^{\mathrm{T}}\theta_{t+1/2}\end{pmatrix}-\begin{pmatrix}-A\phi_t\\ A^{\mathrm{T}}\theta_t\end{pmatrix}\\ &= \begin{pmatrix}-A(\phi_{t+1/2}-\phi_t)\\ A^{\mathrm{T}}(\theta_{t+1/2}-\theta_t)\end{pmatrix}\\ &= \begin{pmatrix}0 & -A\\ A^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}\theta_{t+1/2}-\theta_t\\ \phi_{t+1/2}-\phi_t\end{pmatrix}.\end{aligned} \tag{3.5}$$

由 (3.1) 中预测步的迭代格式, 可得

$$\begin{pmatrix}\theta_{t+1/2}-\theta_t\\ \phi_{t+1/2}-\phi_t\end{pmatrix} = \begin{pmatrix}-\gamma A\phi_t\\ \gamma A^{\mathrm{T}}\theta_t\end{pmatrix} = \begin{pmatrix}0 & -\gamma A\\ \gamma A^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}. \tag{3.6}$$

(3.5) 结合 (3.6), 蕴涵着

$$\begin{aligned}\nabla\bar{V}(\theta_{t+1/2},\phi_{t+1/2})-\nabla\bar{V}(\theta_t,\phi_t) &= \begin{pmatrix}0 & -A\\ A^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}\theta_{t+1/2}-\theta_t\\ \phi_{t+1/2}-\phi_t\end{pmatrix}\\ &= \begin{pmatrix}0 & -A\\ A^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}0 & -\gamma A\\ \gamma A^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}\\ &= \begin{pmatrix}-\gamma AA^{\mathrm{T}} & 0\\ 0 & -\gamma A^{\mathrm{T}}A\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}.\end{aligned} \tag{3.7}$$

要证明投影项 $\prod_{(\nabla\bar{V}(\theta_t,\phi_t))}(\nabla\bar{V}(\theta_{t+1/2},\phi_{t+1/2})-\nabla\bar{V}(\theta_t,\phi_t))$ 为 0, 只需要证明 (3.4) 中分子为 0 即可, 即证明 $\langle(\nabla\bar{V}(\theta_{t+1/2},\phi_{t+1/2})-\nabla\bar{V}(\theta_t,\phi_t)),\nabla\bar{V}(\theta_t,\phi_t)\rangle=0$. 将 (3.7) 代入 (3.4) 中内积项, 有

$$\begin{aligned}&\langle(\nabla\bar{V}(\theta_{t+1/2},\phi_{t+1/2})-\nabla\bar{V}(\theta_t,\phi_t)),\nabla\bar{V}(\theta_t,\phi_t)\rangle\\ &= \left\langle\begin{pmatrix}-\gamma AA^{\mathrm{T}} & 0\\ 0 & -\gamma A^{\mathrm{T}}A\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix},\ \begin{pmatrix}0 & -A\\ A^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}\right\rangle\\ &= \begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}-\gamma AA^{\mathrm{T}} & 0\\ 0 & -\gamma A^{\mathrm{T}}A\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}0 & -A\\ A^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}\\ &= \begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}0 & \gamma AA^{\mathrm{T}}A\\ -\gamma A^{\mathrm{T}}AA^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}\\ &= (-\gamma{\phi_t}^{\mathrm{T}}A^{\mathrm{T}}AA^{\mathrm{T}},\gamma{\theta_t}^{\mathrm{T}}AA^{\mathrm{T}}A)\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}\end{aligned}$$

$$
\begin{aligned}
&= -\gamma {\phi_t}^{\mathrm{T}} A^{\mathrm{T}} A A^{\mathrm{T}} \theta_t + \gamma {\theta_t}^{\mathrm{T}} A A^{\mathrm{T}} A \phi_t \\
&= -\gamma {\phi_t}^{\mathrm{T}} A^{\mathrm{T}} A A^{\mathrm{T}} \theta_t + \gamma {\phi_t}^{\mathrm{T}} A^{\mathrm{T}} A A^{\mathrm{T}} \theta_t \\
&= 0.
\end{aligned}
$$

故 (3.4) 中投影项 $\prod_{(\nabla \bar{V}(\theta_t,\phi_t))}(\nabla \bar{V}(\theta_{t+1/2},\phi_{t+1/2}) - \nabla \bar{V}(\theta_t,\phi_t)) = 0$. □

由引理 3.1 知, 对于博弈 (3.3) 而言, 近似向心加速项 $\nabla \bar{V}(\theta_{t+1/2},\phi_{t+1/2}) - \nabla \bar{V}(\theta_t,\phi_t)$ 与 $\nabla \bar{V}(\theta_t,\phi_t)$ 正交, 因此近似向心加速项 $\nabla \bar{V}(\theta_{t+1/2},\phi_{t+1/2}) - \nabla \bar{V}(\theta_t,\phi_t)$ 在 t 时刻以任意非零预测率 γ 指向圆心方向.

注 3.1 文献 [12, 命题 1] 给出了 GDA 求解二元线性博弈 $\min_{\theta\in\mathbb{R}^n}\max_{\phi\in\mathbb{R}^n}\theta^{\mathrm{T}}\phi$ 会以满足 $\theta_0,\phi_0 \neq 0$ 的任意初始点 (θ_0,ϕ_0) 发散. 受文献 [12, 命题 1] 的启发, 本文提出以下问题: 在几何上如何理解该命题 1 以及 GDA 求解博弈 $\min_{\theta\in\mathbb{R}^n}\max_{\phi\in\mathbb{R}^n}\theta^{\mathrm{T}}A\phi$ (其中 $A\in\mathbb{R}^{n\times n}$ 满秩) 在最后一次迭代意义下也发散吗? 事实上, 引理 3.1 回答了上述两个问题. 由引理 3.1 可知, $\nabla \bar{V}(\theta_t,\phi_t)$、$\nabla \bar{V}(\theta_{t+1/2},\phi_{t+1/2})$ 和 $\nabla \bar{V}(\theta_{t+1/2},\phi_{t+1/2}) - \nabla \bar{V}(\theta_t,\phi_t)$ 会构成一个分别以 $\nabla \bar{V}(\theta_{t+1/2},\phi_{t+1/2}) - \nabla \bar{V}(\theta_t,\phi_t)$ 和 $\nabla \bar{V}(\theta_t,\phi_t)$ 为直角边以及 $\nabla \bar{V}(\theta_{t+1/2},\phi_{t+1/2})$ 为斜边的直角三角形 (见图 2), 这意味着每一次迭代后, 下一次迭代点符号梯度的模相对当前步是不断增大的. 因此 GDA 在二元线性博弈 (3.3) 上是发散的, 即引理 3.1 从几何上解释了文献 [12, 命题 1]. 同时引理 3.1 从几何角度阐述了这一事实, 即用 GDA 求解博弈 $\min_{\theta\in\mathbb{R}^n}\max_{\phi\in\mathbb{R}^n}\theta^{\mathrm{T}}A\phi$ 会以满足 $\theta_0,\phi_0\neq 0$ 的任意初始点 (θ_0,ϕ_0) 发散, 其中 A 为满秩矩阵.

此外, 由引理 3.1 可知 PPCA 在二元线性博弈上投影项为 0, 故 PPCA 在博弈 (3.3) 上迭代格式可简化为

$$
\begin{aligned}
&\text{预测步}: \quad \theta_{t+1/2} = \theta_t - \gamma\nabla_\theta V(\theta_t,\phi_t), \quad \phi_{t+1/2} = \phi_t + \gamma\nabla_\phi V(\theta_t,\phi_t); \\
&\text{更新步}: \quad \theta_{t+1} = \theta_t - \alpha\nabla_\theta V(\theta_t,\phi_t) - \beta(\nabla_\theta V(\theta_{t+1/2},\phi_{t+1/2}) - \nabla_\theta V(\theta_t,\phi_t)), \\
&\qquad\qquad \phi_{t+1} = \phi_t + \alpha\nabla_\phi V(\theta_t,\phi_t) + \beta(\nabla_\phi V(\theta_{t+1/2},\phi_{t+1/2}) - \nabla_\phi V(\theta_t,\phi_t)).
\end{aligned} \tag{3.8}
$$

为便于叙述, 后文实验部分称算法 (3.8) 为算法 PCA. 因其迭代格式相对简单而容易与深度学习

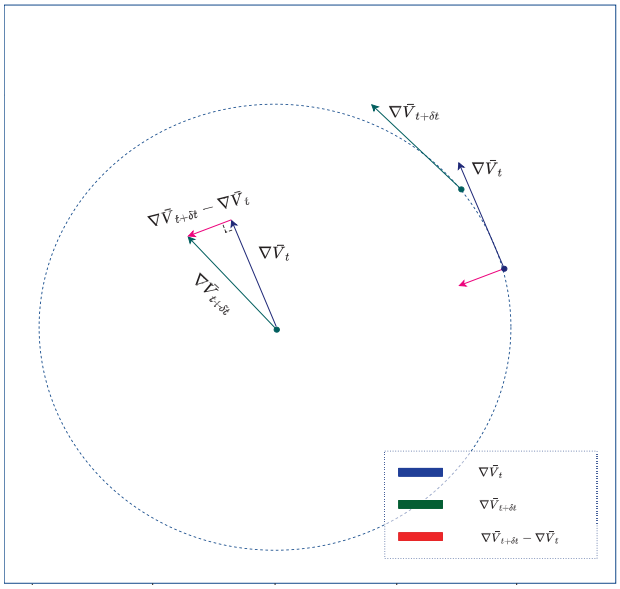


图 2 (网络版彩图) **GDA 在二元线性博弈 (3.3) 上不收敛原因的几何示意图**

算法结合, 且需要的计算开销相对较少, 故本文将其作为主要算法进行研究. 此外, PPCA 在二元线性博弈上迭代格式为 (3.8), 因此在二元线性博弈的理论和实验部分不再区分 PPCA 和 PCA.

不难发现, 通过近似向心加速项对 GDA 进行不同程度的调整可分别获得 OGDA、GOGDA 和 Grad-SCA. 因此, 结合引理 3.1 可得到如下注记.

注 3.2 虽然 PCA 思想源于 OGDA、GOGDA 和 Grad-SCA, 但 PCA 仍与它们有所不同, 主要表现在对 GDA 的调整方式上. OGDA、GOGDA 和 Grad-SCA 使用 t 时刻的近似向心加速项 $\nabla\bar{V}(\theta_{t+1},\phi_{t+1})-\nabla\bar{V}(\theta_t,\phi_t)$ 对算法在 $t+1$ 时刻的表现作调整; 而 PCA 先得到预测步的梯度信息, 后通过该梯度信息得到一个 t 时刻指向圆心的方向对算法在 t 时刻的表现作调整. 特别地, 根据引理 3.1 可知该差异性在二元线性博弈 (3.3) 上表现尤为明显.

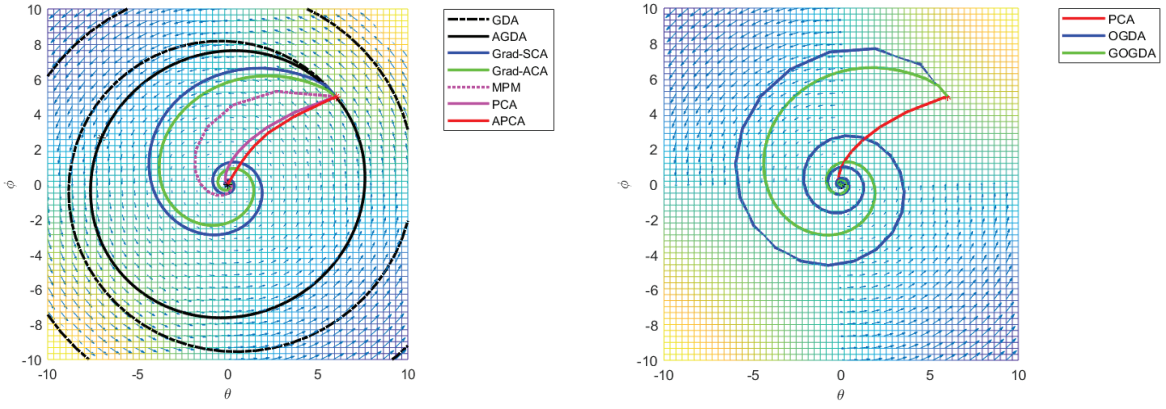
受 AGDA 和 Grad-ACA 的启发, 结合引理 3.1, 本文给出 PCA 的交替版本—APCA, 其迭代格式如下:

$$\begin{aligned}
&\text{预测步}: \quad \theta_{t+1/2}=\theta_t-\gamma\nabla_\theta V(\theta_t,\phi_t), \quad \phi_{t+1/2}=\phi_t+\gamma\nabla_\phi V(\theta_t,\phi_t);\\
&\text{更新步}: \quad \theta_{t+1}=\theta_t-\alpha\nabla_\theta V(\theta_t,\phi_t)-\beta(\nabla_\theta V(\theta_{t+1/2},\phi_{t+1/2})-\nabla_\theta V(\theta_t,\phi_t)),\\
&\qquad\quad \phi_{t+1}=\phi_t+\alpha\nabla_\phi V(\theta_{t+1},\phi_t)+\beta(\nabla_\phi V(\theta_{t+1/2},\phi_{t+1/2})-\nabla_\phi V(\theta_t,\phi_t)).
\end{aligned}$$

为了验证预测投影加速想法的有效性, 本节在二元线性博弈上进行了数值实验, 实验结果如图 3 和 4 所示.

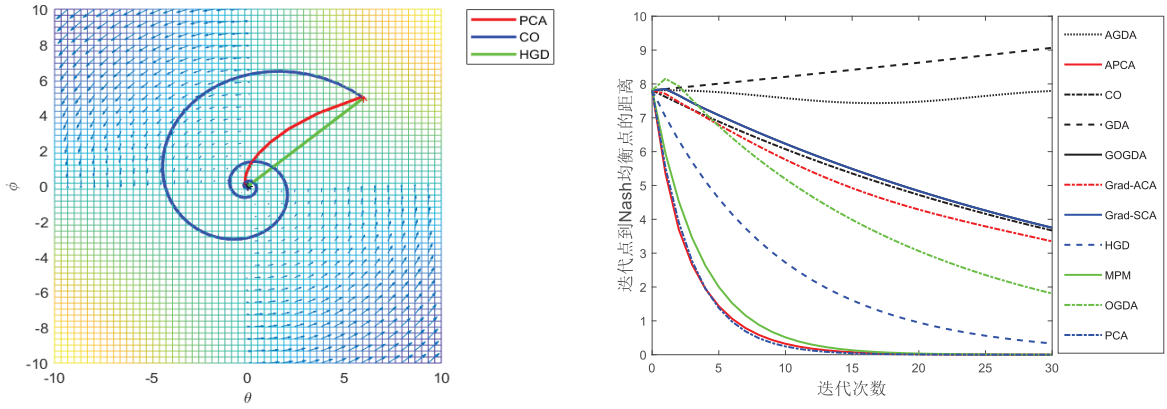
图 3(a) 表明, GDA ($\alpha=0.1,\beta=0$) 发散而 AGDA ($\alpha=0.1$, $\beta=0$) 保持迭代在一个闭轨迹上; Grad-SCA 和 Grad-ACA ($\alpha_1=\alpha_2=0.1$, $\beta_1=\beta_2=0.3$) 可收敛到均衡点, 但交替版本更快一些; MPM ($\gamma=1$, $\beta=0.3$) 可收敛到均衡点; PCA 和 APCA ($\gamma=1$, $\alpha=0.1$, $\beta=0.3$) 均可收敛到均衡点, 且没有循环行为, 后者表现更快一些. 图 3(b) 显示 OGDA ($\alpha=0.3$) 和 GOGDA ($\alpha=0.1$, $\beta=0.3$) 缓慢地向均衡点旋转, 而 PCA ($\gamma=1$, $\alpha=0.1$, $\beta=0.3$) 直接收敛到均衡点.

上面是 PCA 与基于 GDA 迭代格式的其他经典算法 (如 GDA、AGDA、Grad-SCA、Grad-ACA 和 MPM) 在二元线性博弈 (对抗二元线性博弈 [42]) 上的对比实验, 结果表明 PCA 和 APCA 比其他校正 GDA 的方法更有效. 接下来, 本节用 PCA 与基于博弈的经典算法 (如 CO [1,41] 和二阶方法 HGD [1])



(a) (b)

图 3 (网络版彩图) **(a) GDA、AGDA、Grad-SCA、Grad-ACA、MPM、PCA 和 APCA 在二元线性博弈上的数值效果图; (b) OGDA、GOGDA 和 PCA 在二元线性博弈上的数值效果图, 其中 $V(\boldsymbol{\theta},\boldsymbol{\phi})=\boldsymbol{\theta}\cdot\boldsymbol{\phi}$, $\boldsymbol{\theta},\boldsymbol{\phi}\in\mathbb{R}$. 对比实验参数与文献 [48] 中图 2 的相同**



(a) (b)

图 4 (网络版彩图) (a) PCA、HGD 和 CO 在二元线性博弈上的数值效果图, 其中 $V(\boldsymbol{\theta},\boldsymbol{\phi})=\boldsymbol{\theta}\cdot\boldsymbol{\phi}$, $\boldsymbol{\theta},\boldsymbol{\phi}\in\mathbb{R}$; (b) AGDA、APCA、COGDA、Grad-SCA、Grad-ACA、HGD、MPM、OGDA 和 PCA 在二元线性博弈 $V(\boldsymbol{\theta},\boldsymbol{\phi})=\boldsymbol{\theta}\cdot\boldsymbol{\phi}$, $\boldsymbol{\theta},\boldsymbol{\phi}\in\mathbb{R}$ 上迭代 30 次过程中迭代点到 Nash 均衡点的距离图

在二元线性博弈上进行对比实验. 下面对比 PCA、HGD 和 CO 在二元线性博弈上的数值表现 (见图 4(a)). 为更好地说明算法 PCA 和 APCA 的有效性, 本文给出图 3 和 4(a) 中各算法得到近似度相同的近似均衡点时所需迭代次数的一个对比 (见图 4(b)).

从图 4(a) 可以看出, HGD ($\eta=0.1$) 和 PCA ($\gamma=1$, $\alpha=0.1$, $\beta=0.3$) 均可直接收敛到均衡点, CO ($\gamma=0.3$, $\alpha=0.1$) 缓慢地向均衡点旋转. 迭代次数对比结果 (见图 4(b)) 表明 PCA 和 APCA 比其他校正 GDA 的方法更为有效.

4 PCA 在二元线性博弈上的收敛性分析

Mertikopoulos 等 [39] 指出在 GAN 之前对一般二人零和博弈自适应迭代过程的研究已证实, 该迭代过程会持续不变、导致混沌和递归 (循环) 行为等, 最近二元线性博弈作为研究 GAN 训练过程中出现极限循环行为的简单例子被重新重视. 第 3 节给出了 PCA 的迭代格式, 并在二元线性博弈上验证了其缓解极限循环行为的有效性. 下面给出 PCA 求解二元线性博弈时在最后一次迭代意义下的线性收敛性结果.

4.1 PCA 在二元线性博弈上的线性收敛性结果

本小节给出 PCA 求解二元线性博弈 (3.3) 的线性收敛性结果.

定理 4.1 设二元线性博弈 (3.3) 中矩阵 A 满秩, 令 $\delta=\|(\theta_0,\phi_0)\|_2$. 若对任意给定的 $\epsilon>0$ 和 $\gamma>0$, 都有

$$T\geqslant T_{\mathrm{PCA}}:=\left\lceil 2\frac{\lambda_{\max}^2(AA^{\mathrm{T}})}{\lambda_{\min}^2(AA^{\mathrm{T}})-\alpha^2\lambda_{\max}^3(AA^{\mathrm{T}})}\log\frac{\delta}{\epsilon}\right\rceil,$$

则 (3.8) 中 PCA 的迭代过程将以学习率

$$|\alpha|<\frac{\lambda_{\min}(AA^{\mathrm{T}})\sqrt{\lambda_{\max}(AA^{\mathrm{T}})}}{\lambda_{\max}^2(AA^{\mathrm{T}})}$$

和调整率

$$\beta=\frac{\lambda_{\min}(AA^{\mathrm{T}})}{\gamma\lambda^2_{\max}(AA^{\mathrm{T}})}$$

得到一个满足 $(\theta_T,\phi_T)\in B_2(\epsilon)$ 的 ϵ 近似 Nash 均衡点, 其中, $\lambda_{\max}(\cdot)$ 和 $\lambda_{\min}(\cdot)$ 分别表示最大和最小特征值算子, $B_2(\epsilon)$ 表示以原点为圆心、ϵ 为半径的球.

证明 注意到在二元线性博弈 (3.3) 上 PCA 迭代格式 (3.8) 中预测步可简化为

$$\begin{pmatrix}\theta_{t+1/2}\\ \phi_{t+1/2}\end{pmatrix}=\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}+\begin{pmatrix}-\gamma A\phi_t\\ \gamma A^{\mathrm{T}}\theta_t\end{pmatrix}=\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}+\begin{pmatrix}0 & -\gamma A\\ \gamma A^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}. \tag{4.1}$$

然后, (3.8) 中梯度更新步可写为

$$\begin{aligned}\begin{pmatrix}\theta_{t+1}\\ \phi_{t+1}\end{pmatrix}&=\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}+\alpha\begin{pmatrix}-A\phi_t\\ A^{\mathrm{T}}\theta_t\end{pmatrix}+\beta\begin{pmatrix}-A\phi_{t+1/2}-(-A\phi_t)\\ A^{\mathrm{T}}\theta_{t+1/2}-A^{\mathrm{T}}\theta_t\end{pmatrix}\\ &=\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}+\begin{pmatrix}0 & -\alpha A\\ \alpha A^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}+\beta\begin{pmatrix}-A(\phi_{t+1/2}-\phi_t)\\ A^{\mathrm{T}}(\theta_{t+1/2}-\theta_t)\end{pmatrix}\\ &=\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}+\begin{pmatrix}0 & -\alpha A\\ \alpha A^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}+\beta\begin{pmatrix}0 & -A\\ A^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}0 & -\gamma A\\ \gamma A^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}\\ &=\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}+\begin{pmatrix}0 & -\alpha A\\ \alpha A^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}+\begin{pmatrix}-\gamma\beta AA^{\mathrm{T}} & 0\\ 0 & -\gamma\beta A^{\mathrm{T}}A\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}\\ &=\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}+\begin{pmatrix}-\gamma\beta AA^{\mathrm{T}} & -\alpha A\\ \alpha A^{\mathrm{T}} & -\gamma\beta A^{\mathrm{T}}A\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}\\ &=\begin{pmatrix}I-\gamma\beta AA^{\mathrm{T}} & -\alpha A\\ \alpha A^{\mathrm{T}} & I-\gamma\beta A^{\mathrm{T}}A\end{pmatrix}\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix},\end{aligned} \tag{4.2}$$

其中第 3 个等式由 (4.1) 可得.

下面分析算子 $K:=(I-\left(\begin{smallmatrix}\gamma\beta AA^{\mathrm{T}} & \alpha A\\ -\alpha A^{\mathrm{T}} & \gamma\beta A^{\mathrm{T}}A\end{smallmatrix}\right))$ 的奇异值, 或等价地分析 KK^{T} 的奇异值:

$$\begin{aligned}KK^{\mathrm{T}}&=\begin{pmatrix}I-\gamma\beta AA^{\mathrm{T}} & -\alpha A\\ \alpha A^{\mathrm{T}} & I-\gamma\beta A^{\mathrm{T}}A\end{pmatrix}\begin{pmatrix}I-\gamma\beta AA^{\mathrm{T}} & \alpha A\\ -\alpha A^{\mathrm{T}} & I-\gamma\beta A^{\mathrm{T}}A\end{pmatrix}\\ &=\begin{pmatrix}(I-\gamma\beta AA^{\mathrm{T}})^2+\alpha^2AA^{\mathrm{T}} & 0\\ 0 & (I-\gamma\beta A^{\mathrm{T}}A)^2+\alpha^2A^{\mathrm{T}}A\end{pmatrix}.\end{aligned} \tag{4.3}$$

对于任意给定的预测率 γ, 当学习率 α 和调整率 β 选择恰当时, 考虑矩阵 $\alpha^2AA^{\mathrm{T}}+(I-\gamma\beta AA^{\mathrm{T}})^2$ 的最大特征值. 对矩阵 A 进行奇异值分解 (即 $A=UDV$), 可得

$$(I-\gamma\beta AA^{\mathrm{T}})^2+\alpha^2AA^{\mathrm{T}}=U[(I-\gamma\beta D^2)^2+\alpha^2D^2]U^{\mathrm{T}}$$

$$\begin{aligned}&\preceq [1-2\gamma\lambda_{\min}(AA^{\mathrm{T}})\beta+\gamma^2\lambda^2_{\max}(AA^{\mathrm{T}})\beta^2+\alpha^2\lambda_{\max}(AA^{\mathrm{T}})]I\\&=\left[1-\left(\frac{\lambda^2_{\min}(AA^{\mathrm{T}})-\alpha^2\lambda^3_{\max}(AA^{\mathrm{T}})}{\lambda^2_{\max}(AA^{\mathrm{T}})}\right)\right]I,\end{aligned}\tag{4.4}$$

其中 α 和 β 分别满足

$$|\alpha|<\frac{\lambda_{\min}(AA^{\mathrm{T}})\sqrt{\lambda_{\max}(AA^{\mathrm{T}})}}{\lambda^2_{\max}(AA^{\mathrm{T}})}\tag{4.5}$$

和

$$\beta=\frac{\lambda_{\min}(AA^{\mathrm{T}})}{\gamma\lambda^2_{\max}(AA^{\mathrm{T}})}.\tag{4.6}$$

根据 Rayleigh 熵, 结合 (4.2) 和 (4.3), 有

$$\left\|\begin{pmatrix}\theta_{t+1}\\ \phi_{t+1}\end{pmatrix}\right\|_2^2=\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}^{\mathrm{T}}K^{\mathrm{T}}K\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}\leqslant\lambda_{\max}(KK^{\mathrm{T}})\left\|\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}\right\|_2^2.$$

上式结合 (4.3) 和 (4.4), 蕴涵着

$$\left\|\begin{pmatrix}\theta_{t+1}\\ \phi_{t+1}\end{pmatrix}\right\|_2\leqslant\sqrt{1-\left(\frac{\lambda^2_{\min}(AA^{\mathrm{T}})-\alpha^2\lambda^3_{\max}(AA^{\mathrm{T}})}{\lambda^2_{\max}(AA^{\mathrm{T}})}\right)}\left\|\begin{pmatrix}\theta_t\\ \phi_t\end{pmatrix}\right\|_2.\tag{4.7}$$

由递推关系 (4.7), 可得

$$\left\|\begin{pmatrix}\theta_{t+1}\\ \phi_{t+1}\end{pmatrix}\right\|_2\leqslant\left(\sqrt{1-\left(\frac{\lambda^2_{\min}(AA^{\mathrm{T}})-\alpha^2\lambda^3_{\max}(AA^{\mathrm{T}})}{\lambda^2_{\max}(AA^{\mathrm{T}})}\right)}\right)^t\left\|\begin{pmatrix}\theta_0\\ \phi_0\end{pmatrix}\right\|_2.$$

上式结合 $\forall\,(x,\eta)\in\mathbb{R}\times\mathbb{R}^+:(1-x)^\eta\leqslant\mathrm{e}^{-\eta x}$, 有

$$\begin{aligned}\left\|\begin{pmatrix}\theta_{t+1}\\ \phi_{t+1}\end{pmatrix}\right\|_2&\leqslant\left(\sqrt{1-\left(\frac{\lambda^2_{\min}(AA^{\mathrm{T}})-\alpha^2\lambda^3_{\max}(AA^{\mathrm{T}})}{\lambda^2_{\max}(AA^{\mathrm{T}})}\right)}\right)^t\delta\\&\leqslant\exp\left(-\frac{t}{2}\left(\frac{\lambda^2_{\min}(AA^{\mathrm{T}})-\alpha^2\lambda^3_{\max}(AA^{\mathrm{T}})}{\lambda^2_{\max}(AA^{\mathrm{T}})}\right)\right)\delta.\end{aligned}$$

因此, 若

$$T\geqslant T_{\mathrm{PCA}}:=\left\lceil 2\frac{\lambda^2_{\max}(AA^{\mathrm{T}})}{\lambda^2_{\min}(AA^{\mathrm{T}})-\alpha^2\lambda^3_{\max}(AA^{\mathrm{T}})}\log\frac{\delta}{\epsilon}\right\rceil,$$

则 $\|(\theta_T,\phi_T)\|_2\leqslant\epsilon$. □

下面给出定理 4.1 与文献 [12, 定理 1] 最后一次迭代收敛性结果的一个比较.

注 4.1 由 Liang 和 Stokes [33] 的分析可知, 文献 [12, 定理 1] 要获得一个 ϵ 近似 Nash 均衡点, OGDA 需要将学习率缩放到 $\eta(\epsilon)\asymp\epsilon^2$, 且需要的迭代界为

$$T\gtrsim\epsilon^{-4}\log\frac{1}{\epsilon}\cdot\mathrm{Poly}\left(\frac{\lambda_{\max}(AA^{\mathrm{T}})}{\lambda_{\min}(AA^{\mathrm{T}})}\right).\tag{4.8}$$

而定理 4.1 学习率 α 的选取与 ϵ 无关, 且当 PCA 迭代过程进入 Nash 均衡点的 ϵ 邻域时, 需要的迭代界为

$$T \gtrsim \log \frac{1}{\epsilon} \cdot \left(\frac{\lambda_{\max}(AA^{\mathrm{T}})}{\lambda_{\min}(AA^{\mathrm{T}})} \right)^2. \tag{4.9}$$

此时只需要取 $\alpha = 0$. 对比迭代界 (4.8) 和 (4.9) 不难发现, 若 (4.8) 右边为二次多项式, 则 OGDA 迭代界是 PCA 的 ϵ^{-4} 倍; 若 (4.8) 中多项式次数越高, 则 OGDA 迭代界是 PCA 迭代界的倍数越大. 此外, 定理 4.1 在证明过程中不要求参数 $\alpha \geqslant 0$, 只需要其绝对值小于 (4.5) 右边的界即可.

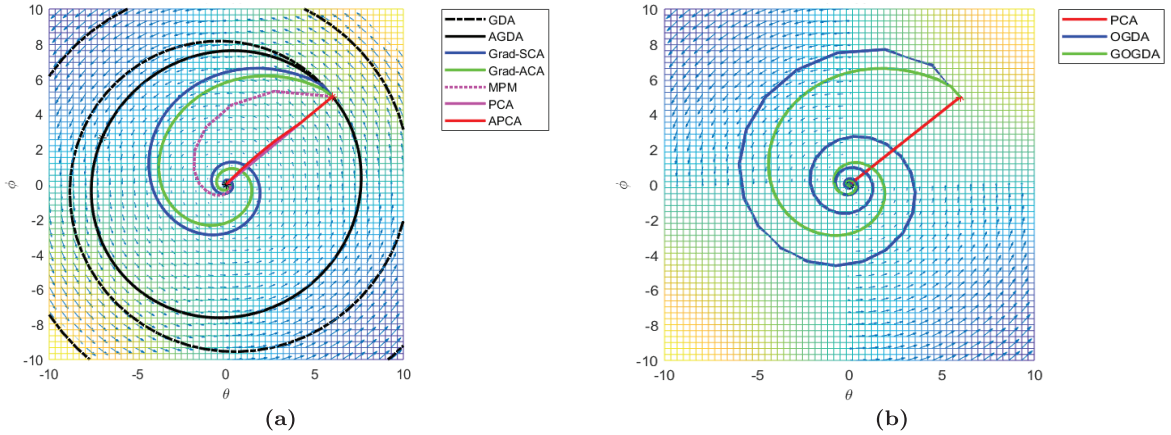
4.2 PCA 与相关算法之间的关系

本小节将对特殊情形下 PCA 的已有理论结果进行简单梳理, 并讨论 GDA、EG、MPM 和 PCA 间的关系.

下面给出 PCA 和 GDA 关系的一个讨论. 若 PCA 中 $\gamma = 0, \alpha \neq 0$, 则 PCA 等价于 GDA, 该算法常用于训练 GAN 及 Wasserstein GAN, 后续文献 [6,12,41,44] 对其有深入研究. 此外, 文献 [23] 指出, 目前既没有理论保证 GDA 训练 GAN 是收敛的, 也没有理论说明 GDA 训练 GAN 是发散的. 然而实践中 GAN 很难成功训练, 且基于 GDA 的 GAN 训练过程常常不收敛 (详见文献 [40]). 实际上, 文献 [12, 命题 1] 给出了 GDA 如下结论: 即使对非常简单的博弈 $\min_{\theta \in \mathbb{R}^n} \max_{\phi \in \mathbb{R}^n} \theta^{\mathrm{T}} \phi$ 而言, GDA 都会以满足 $\theta_0, \phi_0 \neq 0$ 的任意初始点 (θ_0, ϕ_0) 发散. 然而, 定理 4.1 表明, 当参数选取恰当时, PCA 在二元线性博弈上以线性速率收敛, 数值实验如图 3–6 所示.

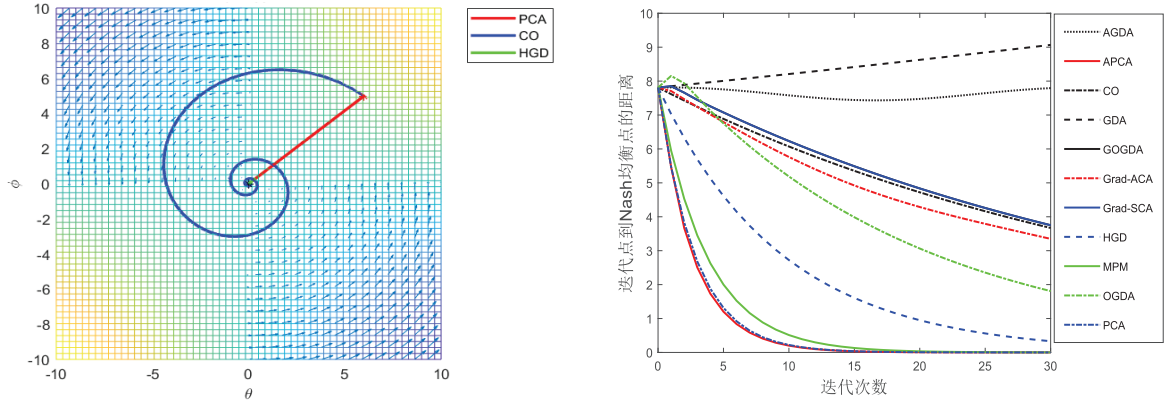
特殊情形 1 PCA 中当 $\gamma \neq 0, \beta \neq 0, \alpha = 0$ 时. 若 PCA 中 $\gamma \neq 0, \beta \neq 0, \alpha = 0$, 则 PCA 在二元线性博弈 (3.3) 上迭代界为 (4.9). 为了验证该情形下算法的有效性, 下面在二元线性博弈 $V(\theta, \phi) = \theta \cdot \phi$ (其中 $\theta, \phi \in \mathbb{R}$) 上进行了数值实验, 详见图 5 和 6.

图 5 和 6 结果表明, 该特殊情形下 PCA 在二元线性博弈上缓解极限循环行为效果明显. 为了更好地说明该特殊情形下 PCA 的有效性, 本节给出图 5 和 6(a) 中各算法得到近似度相同的解时所需



(a) (b)

图 5 (网络版彩图) **(a) GDA、AGDA、Grad-SCA、Grad-ACA、MPM、PCA 和 APCA 在二元线性博弈 $V(\theta, \phi) = \theta \cdot \phi$ 上的数值效果图. PCA 和 APCA 使用参数 $\alpha = 0, \gamma = 1, \beta = 0.3$, 其余算法参数均与图 3 所用参数相同. 数值结果表明 PCA 和 APCA 没有循环行为. (b) PCA、OGDA 和 GOGDA 在二元线性博弈 $V(\theta, \phi) = \theta \cdot \phi$ 上的数值效果图. PCA 使用参数 $\alpha = 0, \gamma = 1, \beta = 0.3$; OGDA 使用参数 $\alpha = 0.3$ (α 不选 0.1 是因为打圈严重导致图不清晰); GOGDA 使用参数 $\alpha = 0.1, \beta = 0.3$. 数值结果表明 PCA 直接收敛到均衡点, OGDA 和 GOGDA 缓慢地向均衡点旋转**



(a) (b)

图 6 (网络版彩图) (a) PCA、CO 和 HGD 在二元线性博弈 $V(\theta,\phi)=\theta\cdot\phi$ ($\theta,\phi\in\mathbb{R}$) 上的数值效果图. PCA 使用参数 $\gamma=1$, $\alpha=0$, $\beta=0.3$; CO 使用参数 $\gamma=0.3$, $\alpha=0.1$; HGD 使用参数 $\alpha=0.1$. HGD 和 PCA 均直接收敛到均衡点, CO 缓慢地向均衡点旋转. (b) AGDA、APCA、GOGDA、Grad-SCA、Grad-ACA、HGD、MPM、OGDA 和 PCA 在二元线性博弈 $V(\theta,\phi)=\theta\cdot\phi$ 上迭代 30 次过程中迭代点到 Nash 均衡点的距离图

迭代次数的一个对比 (如图 6(b) 所示). 迭代次数对比结果图 6 表明, 对二元线性博弈而言, 该特殊情形下的 PCA 和 APCA 比其他算法缓解极限循环行为效果更好, 这可能如前文所述是迭代界变小引起的.

特殊情形 2 PCA 中当 $\beta=\alpha\neq 0$ 时. 显然, 在二元线性博弈 (3.3) 上, 若 PCA 中 $\beta=\alpha\neq 0$, 则 PCA 等价于 MPM. 此时也可获得与文献 [33, 定理 4] 相同的线性收敛性结果.

特殊情形 3 PCA 中当 $\gamma=\alpha=\beta$ 时. 显然, 若 PCA 中 $\gamma=\alpha=\beta$, 则 PCA 等价于文献 [42,43] 中的外梯度算法 (extra-gradient, EG). 此时由于 PCA 迭代格式等价于 EG, 因此可得到文献 [42, 定理 4] 和 [43, 定理 3] 的收敛性结果.

5 PCA 的应用

注意到文献 [12,19–21,38–40] 等的一个共同想法是, 为得到一个原则性训练 GAN 的方法, 先在更受限制的假设下建立原算法的改进, 然后测试这些改进是否会延续到更高要求的学习环境中. 第 3 和 4 节建立了原算法的改进并获得算法 PCA, 为了使其能够适用于更高要求的学习环境, 下面将 PCA 与现有深度学习主要算法 SGD 和 Adam 结合分别提出 SPCA 和 PCA-Adam 用于实际训练 GAN, 以提高其实际性能.

5.1 随机 PCA

近期, Gidel 等 [19,20] 将外推迭代格式写为

$$\begin{aligned}&\text{计算外推点:}\quad \omega_{t+1/2}=P_C[\omega_t+\gamma\nabla\bar{V}(\omega_t)],\\&\text{执行更新:}\quad \omega_{t+1}=P_C[\omega_t+\gamma\nabla\bar{V}(\omega_{t+1/2})],\end{aligned}\tag{5.1}$$

其中, $\omega=(\theta,\phi)$, $P_C[\omega]$ 表示 ω 在集合 C 上的投影. 进一步地, 他们指出外推 (5.1) 存在的问题是算法“浪费”了外推点的梯度, 导致参数每次更新要计算两个不同位置的梯度. 为解决该问题, 他们提出从

过去外推的技巧, 使算法每次更新只需计算一次梯度, 即存储并重新使用外推点的梯度进行外推. 其迭代格式如下:

$$\begin{aligned} &\text{从过去外推:}\quad \omega_{t+1/2}=P_C[\omega_t+\gamma\nabla\bar{V}(\omega_{t-1/2})],\\ &\text{执行更新:}\quad \omega_{t+1}=P_C[\omega_t+\gamma\nabla\bar{V}(\omega_{t+1/2})],\\ &\text{存储:}\quad \nabla\bar{V}(\omega_{t+1/2}). \end{aligned}\tag{5.2}$$

显然, 无约束情形下算法 (5.2) 等价于文献 [12] 中的 OGDA. 基于 Extra-SGD, 对外推步和更新步可用相同的小批量样本, Gidel 等 [20] 提出了重复使用小批量样本进行随机外推的 ReExtra-SGD, 将外推步和更新步梯度的估计联系了起来, 从而缓解了由随机性引起的震荡. 受此思想的启发, 本小节提出随机型 PCA, 即 SPCA (算法 1). 该算法进一步在更新步中引入外推步的梯度估计, 更好地缓解了由随机性导致的震荡. 通过 Wasserstein GAN 学习多元 Gauss 分布的例子, 展示了 SPCA 缓解 GAN 训练过程中极限循环行为的稳定性. 需要说明的是, ReExtra-SGD 使用了在线平均, 而实践中人们更希望算法在迭代结束时迭代点靠近最优解. 当训练 GAN 这类深度神经网络时, 只需考虑无约束情形下算法的实验效果. 因此, 本文实验部分在最后一次迭代意义下展示了 SPCA 的稳定性.

算法 1 SPCA: 使用小批量样本进行随机外推的 PCA

Require: $\omega_0\in C$.

Require: 参数 $\gamma_t,\alpha_t,\beta_t$.

for $t\in\{0,\dots,T-1\}$ **do**

  $\xi_t\sim\mathbb{P}$. (采样)

  $\omega_{t+1/2}\leftarrow\omega_t+\gamma_t\nabla\bar{V}(\omega_t,\xi_t)$. (外推步)

  $\omega_{t+1}\leftarrow\omega_t+\alpha_t\nabla\bar{V}(\omega_t,\xi_t)+\beta_t(\nabla\bar{V}(\omega_{t+1/2},\xi_t)-\nabla\bar{V}(\omega_t,\xi_t))$. (更新步)

end for

return ω_T.

算法 SPCA 可看作加入预测向心加速思想的 SGD. 考虑无约束优化问题, 显然在最后一次迭代意义下, 当 $\alpha_t=\beta_t$ 时, SPCA 等价于 SMPM; 当 $\gamma_t=\alpha_t=\beta_t$ 时, SPCA 等价于 ReExtra-SGD; 当 $\gamma_t=\beta_t=0$ 时, SPCA 等价于 SGD. 换而言之, 在算法 SPCA 框架下理解算法 SMPM 和 ReExtra-SGD 时发现, 在迭代格式上 SGD 比 ReExtra-SGD 更受限制, ReExtra-SGD 比 SMPM 更受限制, SMPM 比 SPCA 更受限制. 对比图 7–14 不难发现, 在实践中当算法参数受限制越多时, 算法可能越容易失败. 算法 SPCA 可用于 SGD、ReExtra-SGD 和 SMPM 训练的机器学习任务, 且在只改变训练算法的条件下, 理论上原算法的数值效果为 SPCA 提供了下界.

5.2 PCA-Adam

基于训练深度神经网络的算法 Adam 对更新步可以有不同的选择, 2019 年 Gidel 等 [19,20] 将外推的技巧融入到 Adam 并提出了 Extra-Adam, 它包含从过去外推和标准外推. Extra-Adam 与文献 [12] 中 optimistic-Adam 主要区别是梯度矩的更新方式不同. 鉴于自适应算法训练 GAN 比随机梯度效果好 [14], 且受 Adam、optimistic-Adam 和 Extra-Adam 设计思想的启发, 本文将 PCA 迭代格式与 Adam 结合提出新算法—PCA-Adam—用于实际训练 GAN.

算法 PCA-Adam 可看作加入预测向心加速思想的 Adam (算法 2, 其中 g_t^2 表示 g_t 按元素平方后的项 $g_t\odot g_t$. 所有对向量的操作都是逐元素进行的. $\beta_1,\beta_2\in[0,1)$: 矩估计的指数衰减率. β_1^t 和 β_2^t 分别表示 β_1 和 β_2 的 t 次幂). 显然, 对无约束问题在最后一次迭代意义下当 $\gamma=\alpha=\beta$ 时, PCA-Adam

等价于 Extra-Adam. 换而言之, 对无约束问题在最后一次迭代意义下, 在算法 PCA-Adam 框架下理解算法 Extra-Adam 时发现, 在迭代格式上 Extra-Adam 比 PCA-Adam 更受限制. 由图 15–18 可见, 在实践中当算法参数受限制越多时, 算法得到的 FID (Fréchet inception distance [27]) 越大, 即生成数据与真实数据差异越大. 对无约束问题在最后一次迭代意义下, 算法 PCA-Adam 可用于由 Extra-Adam 训练的深度学习任务, 且在只改变训练算法的条件下, 理论上 Extra-Adam 的数值效果为 PCA-Adam 提供了下界.

算法 2 PCA-Adam: 带有预测向心加速的 Adam

Require: γ : 预测率, α : 步长, β : 调整率.

Require: $f(\theta)$: 关于参数 θ 的随机目标函数, θ_0 : 初始化参数向量, $m_0 \leftarrow 0$: 初始化一阶矩向量, $v_0 \leftarrow 0$: 初始化二阶矩向量.

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

  Option 1: 标准外推

    $g_{t-1} \leftarrow \nabla_\theta(f_{t-1}(\theta_{t-1}))$.

  Option 2: 从过去外推

    $g_{t-1} \leftarrow \nabla_\theta(f_{t-3/2}(\theta_{t-3/2}))$.

  $m_{t-1/2} \leftarrow \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_{t-1}$, $v_{t-1/2} \leftarrow \beta_2 \cdot v_{t-1} + (1-\beta_2) \cdot g_{t-1}^2$.

  $\widehat{m}_{t-1/2} \leftarrow m_{t-1/2}/(1-\beta_1^{2t-1})$, $\widehat{v}_{t-1/2} \leftarrow v_{t-1/2}/(1-\beta_2^{2t-1})$.

  $\theta_{t-1/2} \leftarrow \theta_{t-1} - \gamma \cdot \frac{\widehat{m}_{t-1/2}}{\sqrt{\widehat{v}_{t-1/2}+\epsilon}}$. (预测步)

  $g_{t-1/2} \leftarrow \nabla_\theta(f_{t-1/2}(\theta_{t-1/2}))$.

  $m_t \leftarrow \beta_1 \cdot m_{t-1/2} + (1-\beta_1) \cdot g_{t-1/2}$, $v_t \leftarrow \beta_2 \cdot v_{t-1/2} + (1-\beta_2) \cdot g_{t-1/2}^2$.

  $\widehat{m}_t \leftarrow m_{t-1/2}/(1-\beta_1^{2t})$; $\widehat{v}_t \leftarrow v_{t-1/2}/(1-\beta_2^{2t})$.

  $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \frac{\widehat{m}_{t-1/2}}{\sqrt{\widehat{v}_{t-1/2}+\epsilon}} - \beta \cdot (\frac{\widehat{m}_t}{\sqrt{\widehat{v}_t+\epsilon}} - \frac{\widehat{m}_{t-1/2}}{\sqrt{\widehat{v}_{t-1/2}+\epsilon}})$. (更新步)

end for

return $\theta_{T-1/2}$ 或 θ_T.

6 数值实验

本节在文献 [12] 利用 GAN 学习多元 Gauss 分布的例子上, 验证了 SPCA 比 SOGDA 和 SMPM 在缓解极限循环行为方面有更好的稳定性. 此外, 借助文献 [20,39] 中 GAN 的网络架构, 在 CIFAR10 和 CelebA 数据集上验证了 PCA-Adam 比 Extra-Adam 训练 GAN 有更好的数值效果.

6.1 随机 PCA 训练 GAN 学习多元 Gauss 分布

本小节对比 SOGDA、SMPM 和 SPCA (算法 1, 其中 $\omega_{t+1/2}$ 取决于 ξ_t, 因此 $\nabla\bar{V}(\omega_t, \xi_t)$ 是 $\nabla\bar{V}(\omega_t)$ 的有偏估计量) 训练 Wasserstein GAN 的数值表现, 以说明算法 SPCA 能更有效地缓解 GAN 训练过程中的极限循环行为. 除特殊说明外, 该部分实验所用 Wasserstein GAN 的网络架构及 SGDA 和 SOGDA 均同于文献 [12].

6.1.1 SPCA 训练 Wasserstein GAN 学习多元 Gauss 分布均值

文献 [12] 的第 3 部分以 Wasserstein GAN 为例说明了 OGDA 缓解极限循环行为的稳定性甚至可延续到其随机版本. 基于文献 [12] 的思想, 本节借助文献 [12] 中的例子在最后一次迭代意义下对比几种算法的随机版本缓解极限循环行为的稳定性, 以凸显 SPCA 稳定性更好的优势.

下面在有采样噪声情形下讨论 GDA、OGDA、MPM 和 PCA 的随机版本在 Wasserstein GAN 上的数值表现. 假设有一组真实样本 $x_1,\dots,x_N$ 和一组噪声样本 $z_1,\dots,z_N$, 与文献 [12] 相同, 本文也试图最小化 Wasserstein GAN 的经验损失

$$\inf_{\theta}\sup_{\phi}\frac{1}{N}\sum_{i=1}^{N}\langle\phi,x_i\rangle-\frac{1}{N}\sum_{i=1}^{N}\langle\phi,z_i+\theta\rangle,$$

上式可等价简化为

$$\inf_{\theta}\sup_{\phi}\frac{1}{N}\sum_{i=1}^{N}\langle\phi,x_i-z_i-\theta\rangle. \tag{6.1}$$

通过 SGDA 来训练生成器和判别器, 即每次只用单个样本 x_i 和 z_i 近似 (6.1) 中经验损失的梯度, 其迭代格式如下:

$$\phi_{t+1}=\phi_t+\eta(x_t-z_t-\theta_t),\quad \theta_{t+1}=\theta_t+\eta\phi_t.$$

此时, SOGDA 迭代格式如下:

$$\phi_{t+1}=\phi_t+2\eta(x_t-z_t-\theta_t)-\eta(x_{t-1}-z_{t-1}-\theta_{t-1}),\quad \theta_{t+1}=\theta_t+2\eta\phi_t-\eta\phi_{t-1}.$$

SMPM 迭代格式如下:

$$\begin{aligned}&\phi_{t+1/2}=\phi_t+\gamma(x_t-z_t-\theta_t),\quad \theta_{t+1/2}=\theta_t+\gamma\phi_t,\\&\phi_{t+1}=\phi_t+\beta(x_t-z_t-\theta_{t+1/2}),\quad \theta_{t+1}=\theta_t+\beta\phi_{t+1/2}.\end{aligned}$$

SPCA 迭代格式如下:

$$\begin{aligned}&\phi_{t+1/2}=\phi_t+\gamma(x_t-z_t-\theta_t),\quad \theta_{t+1/2}=\theta_t+\gamma\phi_t,\\&\phi_{t+1}=\phi_t+\alpha(x_t-z_t-\theta_t)+\beta(x_t-z_t-\theta_{t+1/2}-(x_t-z_t-\theta_t)),\\&\theta_{t+1}=\theta_t+\alpha\phi_t-\beta(-\phi_{t+1/2}+\phi_t).\end{aligned}$$

在有采样噪声情形下, 本文在 Wasserstein GAN 学习多元 Gauss 分布均值的例子上对比了 SGDA、SOGDA、SMPM 和 SPCA 迭代过程的稳定性, 详见图 7–10, 其中, 子图中蓝色为判别器和生成器的参数在第一维上的迭代过程, 橘红色为判别器和生成器的参数在第二维上的迭代过程.

文献 [12] 指出在图 8 中只要批量大小 (batch size, BS) 合适, OGDA 的稳定性可延续至其随机版本. 实际上, 在对应参数相同的情形下对比图 7–10 可知, 即使使用防止循环的权重裁剪技巧, SGDA 还是表现出严重的极限循环行为. SOGDA 在一定程度上缓解了极限循环行为, 但其对批量大小有要求. 实践中, 一个合适的批量大小的数值可能不容易得到. 同样地, SMPM 也在一定程度上缓解了极限循环行为, 但其易受批量大小的影响. 然而, 在该例子上 SPCA 缓解极限循环行为表现优异, 且受批量大小影响不大.

6.1.2 SPCA 训练 Wasserstein GAN 学习多元 Gauss 分布协方差矩阵

Daskalakis 等在文献 [12, 附录 C] 中展示了 SGDA 训练 Wasserstein GAN 的不稳定性, 凸显了 SOGDA 训练 Wasserstein GAN 的稳定性. 遵循其研究思路, 本文讨论有采样噪声时 SGDA、SOGDA、SMPM 和 SPCA 在最后一次迭代意义下迭代过程的稳定性.

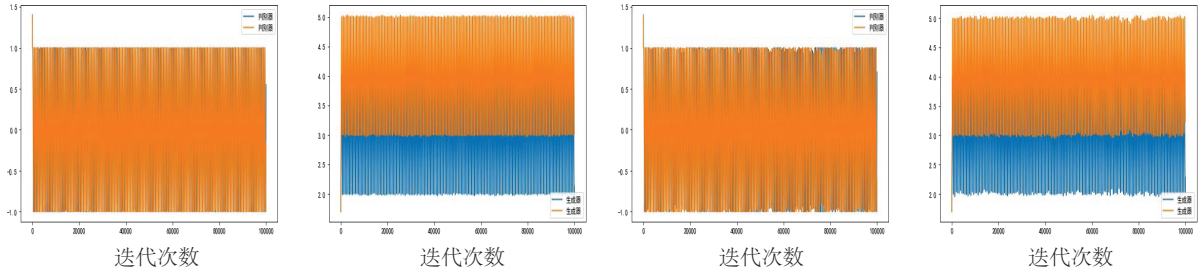


图 7 (网络版彩图) 批量大小为 (a) 200 和 (b) 50 时 SGDA 的迭代过程图 (为了使 SGDA 的迭代过程不循环, 在迭代过程中对判别器参数使用了 1 的权重裁剪) [12]

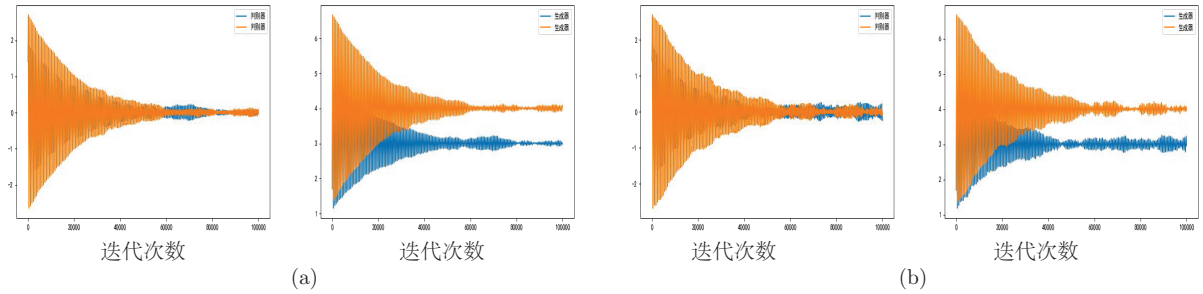


图 8 (网络版彩图) 批量大小为 (a) 200 和 (b) 50 时 SOGDA 的迭代过程图 [12]

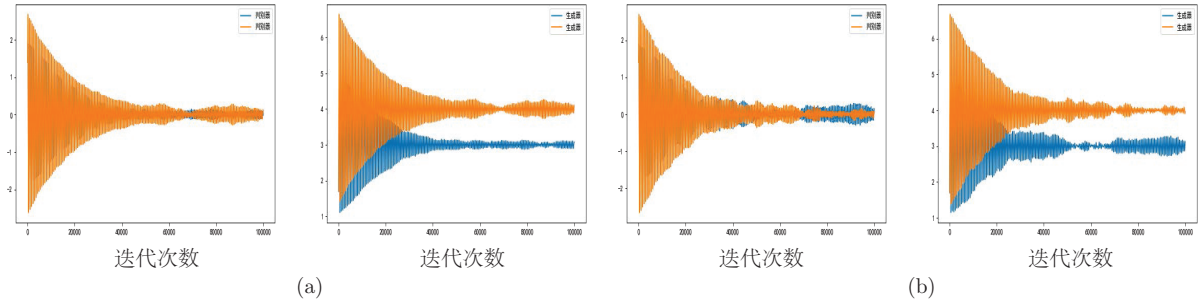


图 9 (网络版彩图) 批量大小为 (a) 200 和 (b) 50 时 SMPM 的迭代过程图

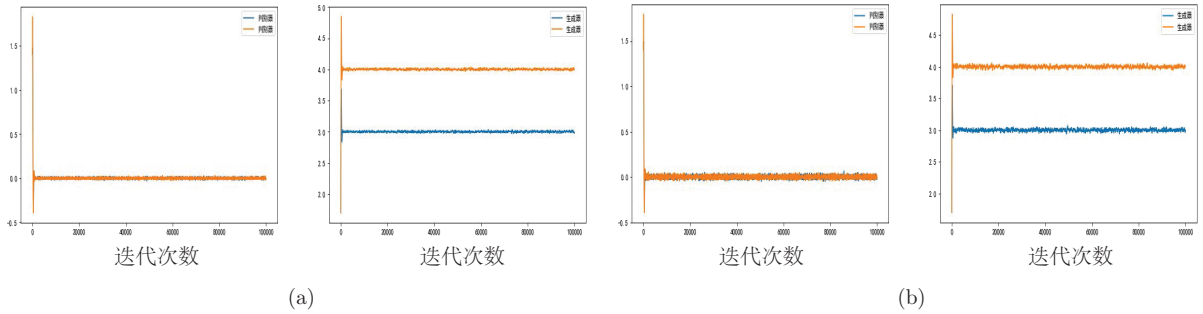


图 10 (网络版彩图) 批量大小为 (a) 200 和 (b) 50 时 SPCA 的迭代过程图

假设数据分布为多元 Gauss 分布, 其均值为 0, 协方差矩阵未知, 即 $x \sim N(0,\Sigma)$. 考虑判别器为二次函数的情形, 即 $D_W(x)=\sum_{ij}W_{ij}x_ix_j=x^{\mathrm{T}}Wx$; 生成器为输入随机噪声 $z\sim N(0,\mathbb{I})$ 的线性函数, 即 $G_V(z)=Vz$, 其中 $W,V\in\mathbb{R}^{n\times n}$. 具有上述生成器和判别器的 Wasserstein GAN 损失函数为

$$L(V,W)=\mathrm{E}_{x\sim N(0,\Sigma)}[x^{\mathrm{T}}Wx]-\mathrm{E}_{z\sim N(0,\mathbb{I})}[z^{\mathrm{T}}V^{\mathrm{T}}WVz]. \tag{6.2}$$

假设协方差矩阵是对称正定矩阵, 由 Cholesky 分解可知 $\Sigma=UU^{\mathrm{T}}$, 故 (6.2) 可简化为

$$\begin{aligned}L(V,W)&=\sum_{ij}W_{ij}(\Sigma_{ij}-\sum_{k}V_{ik}V_{jk})\\&=\sum_{ijk}W_{ij}(U_{ik}U_{jk}-V_{ik}V_{jk}).\end{aligned}$$

上述过程的具体细节详见文献 [12]. 在小批量情形下, 梯度的无偏估计为

$$\hat{\nabla}_{W,t}=x_tx_t^{\mathrm{T}}-V_tz_tz_t^{\mathrm{T}}V_t^{\mathrm{T}},\quad \hat{\nabla}_{V,t}=-(W_t+W_t^{\mathrm{T}})V_tz_tz_t^{\mathrm{T}}, \tag{6.3}$$

其中 x_t 和 z_t 分别是真实分布和随机噪声分布的采样. 因此, 学习多元 Gauss 分布协方差矩阵的 SGDA 迭代格式为

$$W_t=W_{t-1}+\alpha\hat{\nabla}_{W,t-1},\quad V_t=V_{t-1}-\alpha\hat{\nabla}_{V,t-1}.$$

学习多元 Gauss 分布协方差矩阵的 SOGDA 迭代格式为

$$W_t=W_{t-1}+2\alpha\hat{\nabla}_{W,t-1}-\alpha\hat{\nabla}_{W,t-2},\quad V_t=V_{t-1}-2\alpha\hat{\nabla}_{V,t-1}+\alpha\hat{\nabla}_{V,t-2}.$$

学习多元 Gauss 分布协方差矩阵的 SMPM 迭代格式为

$$\begin{aligned}&W_{t-1/2}=W_{t-1}+\gamma\hat{\nabla}_{W,t-1},\quad V_{t-1/2}=V_{t-1}-\gamma\hat{\nabla}_{V,t-1},\\&W_t=W_{t-1}+\alpha\hat{\nabla}_{W,t-1/2},\quad V_t=V_{t-1}-\alpha\hat{\nabla}_{V,t-1/2}.\end{aligned}$$

学习多元 Gauss 分布协方差矩阵的 SPCA 迭代格式为

$$\begin{aligned}&W_{t-1/2}=W_{t-1}+\gamma\hat{\nabla}_{W,t-1},\quad V_{t-1/2}=V_{t-1}-\gamma\hat{\nabla}_{V,t-1},\\&W_t=W_{t-1}+\alpha\hat{\nabla}_{W,t-1}+\beta(\hat{\nabla}_{W,t-1/2}-\hat{\nabla}_{W,t-1}),\\&V_t=V_{t-1}-\alpha\hat{\nabla}_{V,t-1}-\beta(\hat{\nabla}_{V,t-1/2}-\hat{\nabla}_{V,t-1}).\end{aligned}$$

在有采样噪声情形下, 本小节在 Wasserstein GAN 学习多元 Gauss 分布协方差矩阵的例子上对比了 SGDA、SOGDA、SMPM 和 SPCA 迭代过程的稳定性, 详见图 11–14.

图 11–14 描述了 SGDA 迭代过程的不稳定性, 突出了 SOGDA、SMPM 和 SPCA 迭代过程的稳定性, 通过对比发现 SPCA 迭代过程中后期的稳定性更好.

6.2 PCA-Adam 训练 GAN 生成图片

本小节实验目的与文献 [12,19,20,39,52] 相同, 不是通过改进 GAN 网络架构或改变 GAN 的损失函数以得到更好的 FID 值, 而是利用之前所介绍的优化技巧 (在二元线性情形下有理论支撑) 以更好

的优化方式训练标准的 GAN. 优化技巧与 GAN 网络架构的改进及损失函数的设置属于不同维度, 因而所介绍的优化技巧可用于训练任何类型的 GAN.

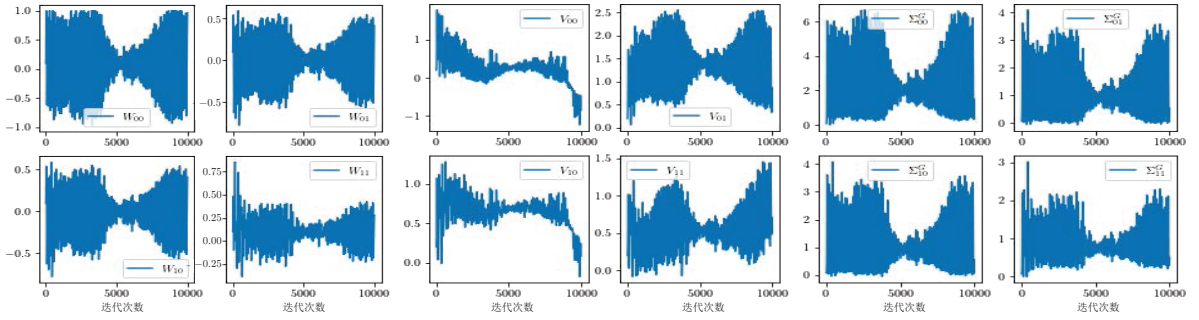
需要说明的是, IS (inception score[53]) 值和 FID 值都是 GAN 生成数据质量好坏的评价指标, FID 值在计算过程中通过引入真实数据的信息, 进而弥补了 IS 值因未考虑真实数据而无法反映真实数据和生成数据之间距离的不足. FID 值从特征层面刻画了真实数据和生成数据特征向量之间的距离, 其值越小表示两组数据越相似. FID 值计算公式为

$$d^2((\mu_x,\Sigma_x),(\mu_g,\Sigma_g))=\|\mu_x-\mu_g\|_2^2+\mathrm{Tr}(\Sigma_x+\Sigma_g-2(\Sigma_x\Sigma_g)^{1/2}),$$

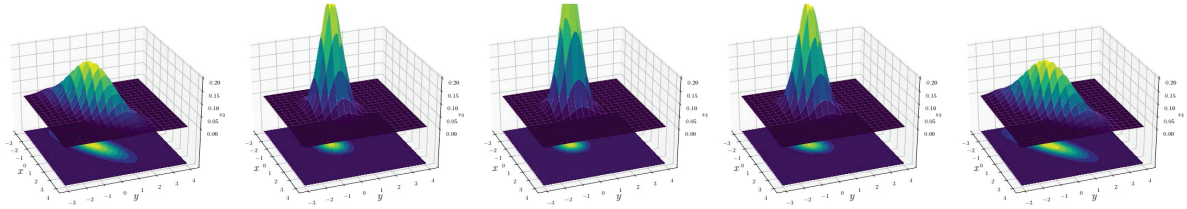
其中, μ_x 和 Σ_x 分别表示真实数据特征的均值和协方差矩阵, μ_g 和 Σ_g 分别表示生成数据特征的均值和协方差矩阵, Tr 表示矩阵的迹.

6.2.1 PCA-Adam 在 CIFAR10 数据集上训练 GAN 生成图片

文献 [12, 20, 39] 设计出新 Adam 型算法后均利用 GAN 网络架构在 CIFAR10 或 CelebA 数据集上进行了测试, 并通过 IS 值或 FID 值说明其所提出算法的有效性. 遵循文献 [12, 20, 39] 的研究思路, 本文在与其相同的网络架构及算法相应位置参数相同的条件下, 通过 FID 值说明 PCA-Adam 的有效性, 具体网络架构、实验环境和参数设置详见表 1–3. 值得说明的是, 文献 [20, 图 12] 已在该实验上对比了 Extra-Adam 和 Adam 训练 DCGAN 的 FID 值, 并表明 Extra-Adam 较 Adam 训练 DCGAN 有更小的 FID 值, 因此本文只需要与 Extra-Adam 训练 DCGAN 作 FID 值的对比, 对比结果详见图 15.

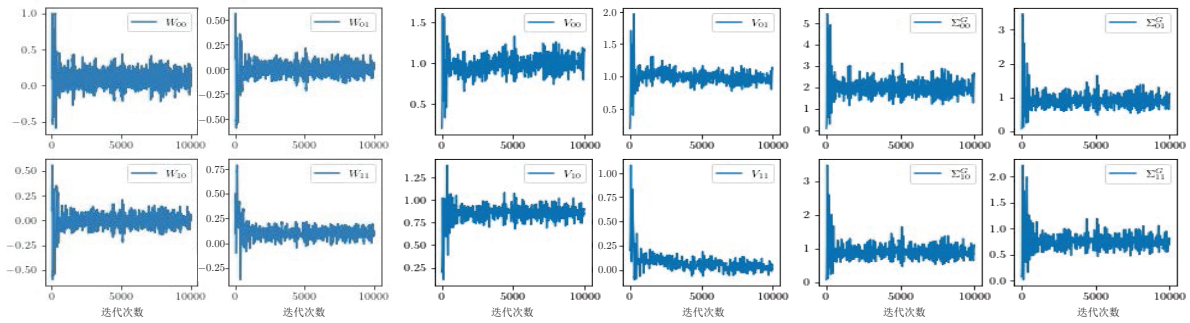


(a) 判别器权重　(b) 生成器权重　(c) 生成的协方差矩阵

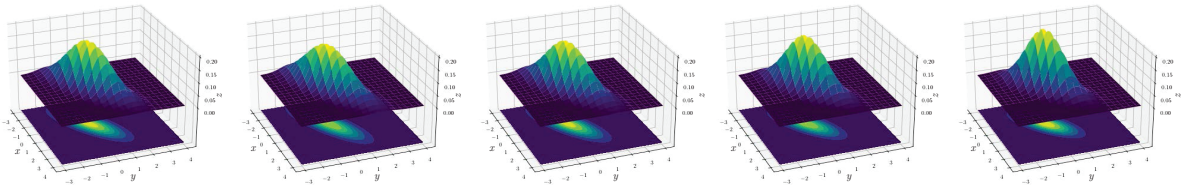


(d) 真实分布　(e) 迭代 $T-50$ 次　(f) 迭代 $T-35$ 次　(g) 迭代 $T-20$ 次　(h) 迭代 T 次

图 11　(网络版彩图) (a)–(c) 为用 SGDA 训练 Wasserstein GAN 学习二维 Gauss 分布 ($d=2$) 协方差矩阵的迭代过程图, 其中, BS $=50$, $\alpha=0.02$, $T=10{,}000$, $\lambda=0.1$, 且判别器使用权重裁剪技巧, 将其权重裁剪到 $[-1,1]$; (d)–(f) 分别为真实分布和接近训练结束时各点生成器分布[12]

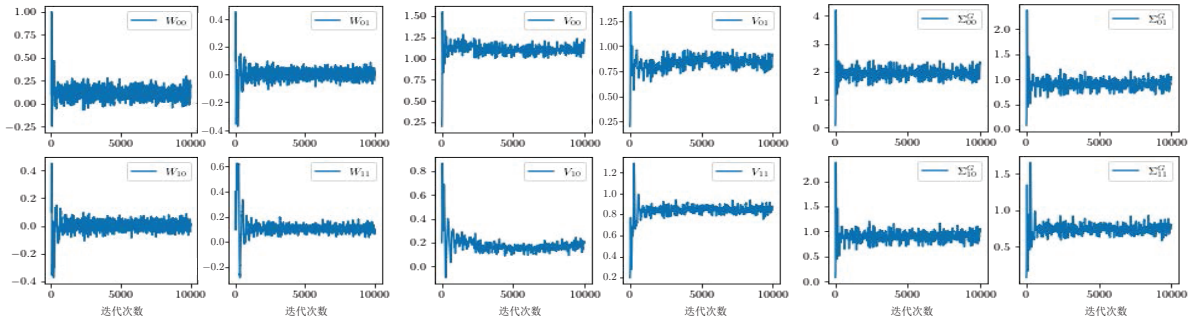


(a) 判别器权重 (b) 生成器权重 (c) 生成的协方差矩阵

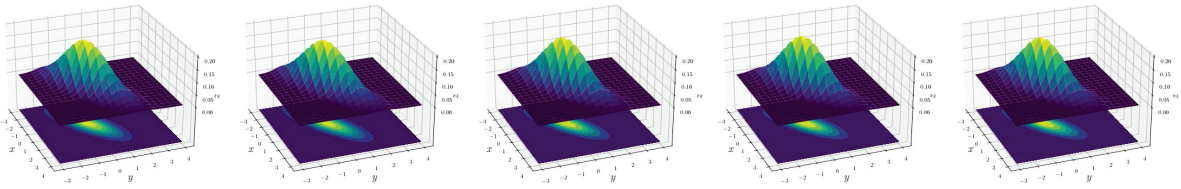


(d) 真实分布 (e) 迭代 $T-50$ 次 (f) 迭代 $T-35$ 次 (g) 迭代 $T-20$ 次 (h) 迭代 T 次

图 12 (网络版彩图) (a)–(c) 为用 SOGDA 训练 Wasserstein GAN 学习二维 Gauss 分布 ($d=2$) 协方差矩阵的迭代过程图, 其中, BS $=50$, $\alpha=0.02$, $T=10{,}000$, $\lambda=0.1$, 且判别器使用权重裁剪技巧, 将其权重裁剪到 $[-1,1]$; (b)–(f) 分别为真实分布和接近训练结束时各点生成器分布 [12]

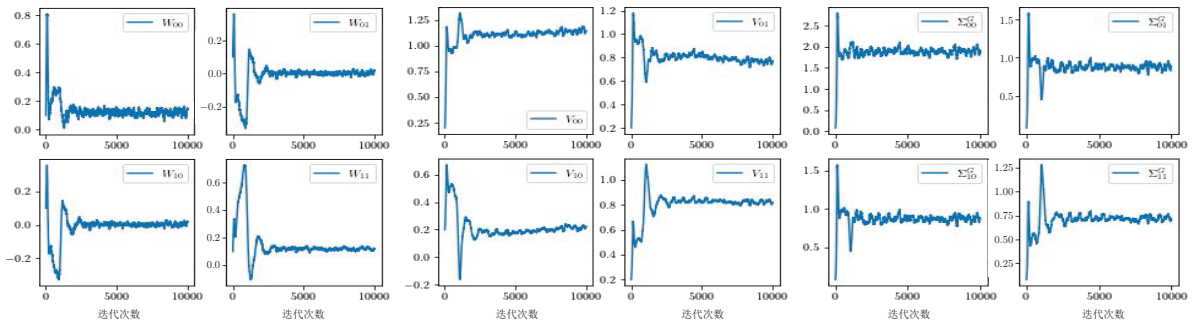


(a) 判别器权重 (b) 生成器权重 (c) 生成的协方差矩阵

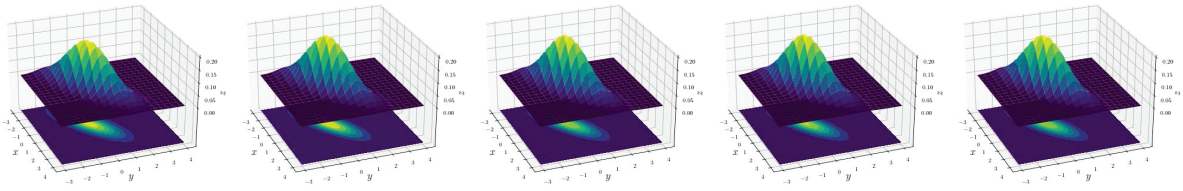


(d) 真实分布 (e) 迭代 $T-50$ 次 (f) 迭代 $T-35$ 次 (g) 迭代 $T-20$ 次 (h) 迭代 T 次

图 13 (网络版彩图) (a)–(c) 为用 SMPM 训练 Wasserstein GAN 学习二维 Gauss 分布 ($d=2$) 协方差矩阵的迭代过程图, 其中, BS $=50$, $\gamma=0.002$, $\alpha=0.02$, $T=10{,}000$, $\lambda=0.1$, 且判别器使用权重裁剪技巧, 将其权重裁剪到 $[-1,1]$; (d)–(f) 分别为真实分布和接近训练结束时各点生成器分布



(a) 判别器权重　(b) 生成器权重　(c) 生成的协方差矩阵



(d) 真实分布　(e) 迭代 $T-50$ 次　(f) 迭代 $T-35$ 次　(g) 迭代 $T-20$ 次　(h) 迭代 T 次

图 14　(网络版彩图) (a)–(c) 为用 SPCA 训练 Wasserstein GAN 学习二维 Gauss 分布 ($d=2$) 协方差矩阵的迭代过程图, 其中, BS $=50$, $\gamma=0.002$, $\alpha=0.006$, $\beta=0.02$, $T=10{,}000$, $\lambda=0.1$, 且判别器使用权重裁剪技巧, 将其权重裁剪到 $[-1,1]$; (d)–(f) 分别为真实分布和接近训练结束时各点生成器分布

表 1　CIFAR10 实验所用的 DCGAN 网络架构 (参见文献 [20, 表 2])

生成器			
输入层: $z\in\mathbb{R}^{128}\sim N(0,I)$			
线性层 $128\to4\times4\times512$		批规范化	ReLU
4×4 conv.T 步幅 $=2$ 填充 $=1$	$512\to256$	批规范化	ReLU
4×4 conv.T 步幅 $=2$ 填充 $=1$	$256\to128$	批规范化	ReLU
4×4 conv.T 步幅 $=2$ 填充 $=1$	$128\to3$		Tanh
判别器			
输入层: $x\in\mathbb{R}^{3\times32\times32}$			
4×4 conv. 步幅 $=2$ 填充 $=1$	$1\to64$		lReLU
4×4 conv. 步幅 $=2$ 填充 $=1$	$64\to128$	批规范化	lReLU
4×4 conv. 步幅 $=2$ 填充 $=1$	$128\to25$	批规范化	lReLU
线性层 $128\times4\times4\times4\to1$			

注: ReLU 表示整流线性单位函数, conv.T 是转置卷积, conv. 表示卷积.

表 2　CIFAR10 和 CelebA 数据集上的实验环境

实验环境		
CPU: Inter Core i7-10700	GPU: RTX A4000 16G	RAM: 64 G
Python: Version 3.8.0	Pytorch: Version 1.9.0	

表 3 CIFAR10 数据集上实验超参数设置

(DCGAN) WGAN-GP 超参数		
WGAN-GP	BS = 64	最大迭代次数 $T = 200{,}000$
Adam $\beta_1 = 0.5$	Adam $\beta_2 = 0.9$	梯度罚 $= 10$
生成器学习率	(Extra-Adam) $\gamma = 5\text{E}{-4}$	
	(PCA-Adam) $\gamma = \beta = 5\text{E}{-4}$,	$\alpha = 4.5\text{E}{-4}$
判别器学习率	(Extra-Adam) $\gamma = 5\text{E}{-4}$	
	(PCA-Adam) $\gamma = \beta = 5\text{E}{-4}$,	$\alpha = 4.5\text{E}{-4}$

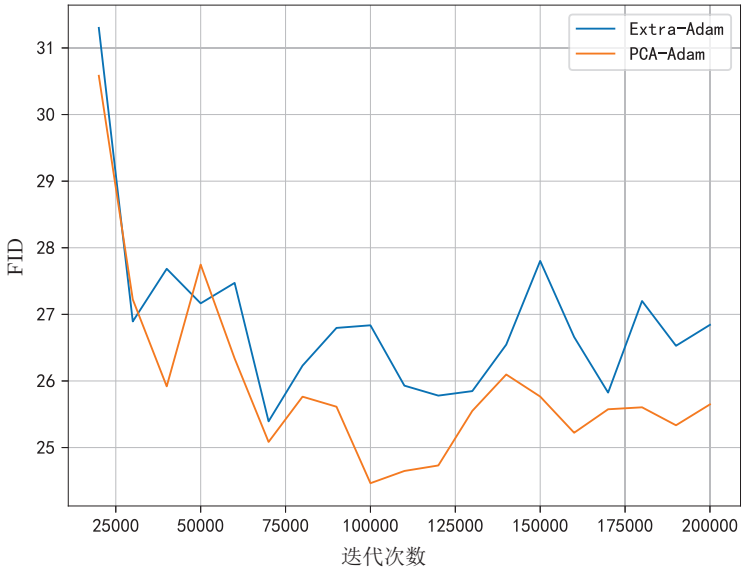


图 15 (网络版彩图) CIFAR10 数据集上算法 PCA-Adam 和 Extra-Adam 训练 DCGAN 的 FID 值对比图. 横坐标表示迭代次数, 纵坐标表示 FID 值的大小

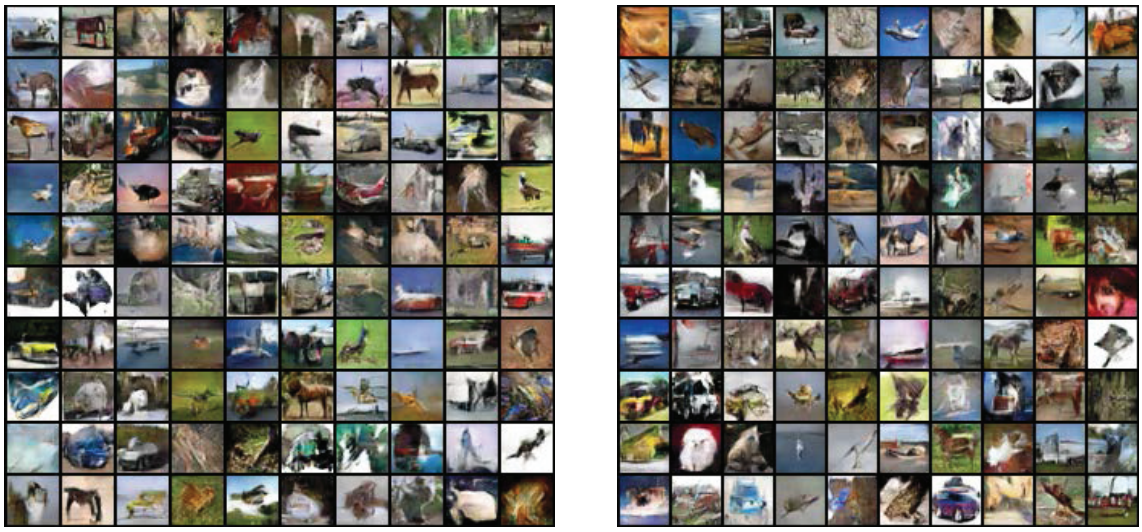
图 15 为 PCA-Adam 和 Extra-Adam 训练 DCGAN 生成数据与真实数据在特征层面距离远近的比较. 由图 15 可知, 在训练过程的中后期, PCA-Adam 训练 DCGAN 比 Extra-Adam 训练 DCGAN 持续具有更小的 FID 值. 因此, PCA-Adam 比 Extra-Adam 训练 DCGAN 生成数据的分布更接近真实分布. Extra-Adam 和 PCA-Adam 训练 DCGAN 迭代 200,000 次后生成的图片见图 16, 对比两张图片可知 Extra-Adam 训练该 DCGAN 生成的图片相对更模糊一些.

分析图 15 和 16 的实验过程可知, 向心加速思想的引入不仅稳定了 DCGAN 的训练过程, 而且相对提高了其生成数据的质量. 因此, 图 15 和 16 的结果表明 PCA-Adam 训练 GAN 的效果明显优于 Extra-Adam.

6.2.2 PCA-Adam 在 CelebA 数据集上训练 GAN 生成图片

本文借助文献 [39] 的网络架构, 且在算法相应位置参数相同条件下, 通过 FID 值说明 PCA-Adam 训练 DCGAN 的有效性. 实验环境同表 2, DCGAN 网络架构和参数设置见表 4 和 5. 对比实验结果详见图 17 和 18.

由图 17 可知, 向心加速思想的引入使得在 DCGAN 的整个训练过程中使用 PCA-Adam 比 Extra-Adam 能持续获得更小的 FID 值, 故 PCA-Adam 比 Extra-Adam 训练 DCGAN 生成数据更接近真实数据. Extra-Adam 和 PCA-Adam 训练 DCGAN 迭代 200,000 次后生成的图片见图 18, 结合 FID 值分析两幅图片, 整体而言由 Extra-Adam 训练 DCGAN 生成的图片相对更模糊一些.



(a) (b)

图 16 (网络版彩图) (a) Extra-Adam 在 CIFAR10 数据集上训练 DCGAN 迭代 200,000 次生成器生成的图片; (b) PCA-Adam 在 CIFAR10 数据集上训练 DCGAN 迭代 200,000 次生成器生成的图片

表 4 CelebA 数据集上 DCGAN 网络架构 (参见文献 [39, 表 1])

生成器			
输入层: $z \in \mathbb{R}^{100} \sim N(0, I)$			
全连接层	$4 \times 4 \times 512$	批规范化	ReLU
4×4 conv.T 步幅 $= 2$	256	批规范化	ReLU
4×4 conv.T 步幅 $= 2$	128	批规范化	ReLU
4×4 conv.T 步幅 $= 2$	64	批规范化	ReLU
3×3 conv.T 步幅 $= 1$	3	权重规范化	Tanh
判别器			
输入层: $x \in \mathbb{R}^{3 \times 32 \times 32}$			
3×3 conv. 步幅 $= 1$	64		lReLU
3×3 conv. 步幅 $= 2$	128		lReLU
3×3 conv. 步幅 $= 1$	128		lReLU
3×3 conv. 步幅 $= 2$	256		lReLU
3×3 conv. 步幅 $= 1$	256		lReLU
3×3 conv. 步幅 $= 2$	512		lReLU
3×3 conv. 步幅 $= 1$	512		lReLU
全连接层	1		

注: ReLU 表示整流线性单位函数, conv.T 是转置卷积, conv. 表示卷积.

表 5 CIFAR10 数据集上的实验超参数设置

(DCGAN) WGAN-GP 超参数		
WGAN-GP	BS = 64	最大迭代次数 $T = 200{,}000$
Adam $\beta_1 = 0.0$	Adam $\beta_2 = 0.9$	梯度罚 $= 1.0$
生成器学习率	(Extra-Adam) $\gamma = 1\text{E}{-4}$	
	(PCA-Adam) $\gamma = \beta = 1\text{E}{-4}$,	$\alpha = 0.8\text{E}{-4}$
判别器学习率	(Extra-Adam) $\gamma = 1\text{E}{-4}$	
	(PCA-Adam) $\gamma = \beta = 1\text{E}{-4}$,	$\alpha = 0.8\text{E}{-4}$

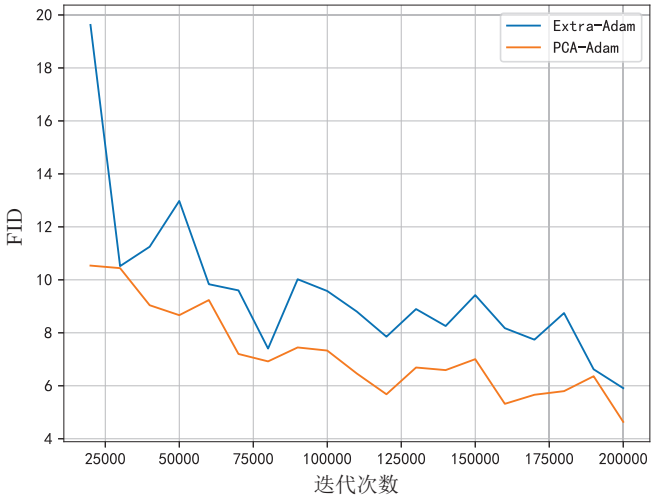


图 17 (网络版彩图) CelebA 数据集上 PCA-Adam 和 Extra-Adam 训练 DCGAN 的 FID 值对比图



(a) (b)

图 18 (网络版彩图) (a) Extra-Adam 在 CelebA 数据集上训练 DCGAN 迭代 200,000 次后生成器生成的图片; (b) PCA-Adam 在 CelebA 数据集上训练 DCGAN 迭代 200,000 次后生成器生成的图片

图 17 和 18 的实验结果表明, 向心加速思想的引入使 DCGAN 的训练过程更稳定, 且获得的 FID 值更小, 生成图片更清晰. 因此, PCA-Adam 训练 GAN 的效果明显优于 Extra-Adam.

7 结论

基于文献 [28] 提出的如何使用优化技巧消除零和博弈中的极限循环行为这一公开问题, 受 OGDA、GOGDA、Grad-SCA 和 MPM 思想的启发, 本文提出了 PCA 用于缓解 GDA 求解可微二人零和博弈时表现出的极限循环行为. 本质上训练一般的 GAN 需要用优化算法并借助深度学习框架求解一个非凸非凹零和博弈. 从目前国际上对非凸非凹零和博弈的研究现状来看, 无论是理论方面还是算法方面都有许多重要且难度大的问题需要解决 [57]. 本文得到了 PCA 在二元线性博弈上的最后一次迭代全局收敛性, 那么能否得到 PCA 在一般非凸非凹零和博弈上的最后一次迭代全局收敛性呢? 这是一个值得继续思考的问题.

致谢 衷心感谢审稿专家提出的宝贵意见, 同时也感谢编委在处理本文时所付出的辛勤劳动.

参考文献

1 Abernethy J, Lai K A, Wibisono A. Last-iterate convergence rates for min-max optimization: Convergence of Hamiltonian gradient descent and consensus optimization. In: Algorithmic Learning Theory. New York: PMLR, 2021, 3–47

2 Alawieh M B, Li W, Lin Y, et al. High-definition routing congestion prediction for large-scale FPGAs. In: 2020 25th Asia and South Pacific Design Automation Conference. New York: IEEE, 2020, 26–31

3 Anagnostides I, Penna P. Solving zero-sum games through alternating projections. arXiv:2010.00109, 2020

4 Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. In: International Conference on Machine Learning. New York: ICML, 2017, 214–223

5 Azizian W, Mitliagkas I, Lacoste-Julien S, et al. A tight and unified analysis of gradient-based methods for a whole spectrum of differentiable games. In: International Conference on Artificial Intelligence and Statistics. New York: PMLR, 2020, 2863–2873

6 Bailey J P, Gidel G, Piliouras G. Finite regret and cycles with fixed step-size via alternating gradient descent-ascent. In: Conference on Learning Theory. New York: PMLR, 2020, 391–407

7 Balduzzi D, Racaniere S, Martens J, et al. The mechanics of n-player differentiable games. In: International Conference on Machine Learning. New York: ICML, 2018, 354–363

8 Berard H, Gidel G, Almahairi A, et al. A closer look at the optimization landscapes of generative adversarial networks. In: International Conference on Learning Representations. Washington: ICLR, 2020

9 Brock A, Donahue J, Simonyan K. Large scale GAN training for high fidelity natural image synthesis. In: International Conference on Learning Representations. Washington: ICLR, 2018

10 Chavdarova T, Jordan M I, Zampetakis M. Last-iterate convergence of saddle point optimizers via high-resolution differential equations. arXiv:2112.13826, 2021

11 Chavdarova T, Pagliardini M, Jaggi M, et al. Taming GANs with lookahead-minmax. In: International Conference on Learning Representations. Washington: ICLR, 2021

12 Daskalakis C, Ilyas A, Syrgkanis V, et al. Training GANs with optimism. In: International Conference on Learning Representations. Washington: ICLR, 2018

13 Daskalakis C, Panageas I. The limit points of (optimistic) gradient descent in min-max optimization. In: Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2018, 9236–9246

14 Deng Q, Gao J J, Ge D D, et al. Modern optimization theory and applications (in Chinese). Sci Sin Math, 2020, 50: 899–968 [邓琪, 高建军, 葛冬冬, 等. 现代优化理论与应用. 中国科学: 数学, 2020, 50: 899–968]

15 Fang S, Han F, Liang W Y, et al. An improved conditional generative adversarial network for microarray data. In: International Conference on Intelligent Computing. Berlin: Springer, 2020, 105–114

16 Farnia F, Ozdaglar A. Do GANs always have Nash equilibria? In: International Conference on Machine Learning. New York: ICML, 2020, 3029–3039

17 Fiez T, Sim R, Skoulakis S, et al. Online learning in periodic zero-sum games. In: Advances in Neural Information Processing Systems, vol. 34. Cambridge: MIT Press, 2021, 10313–10325

18 Flokas L, Vlatakis-Gkaragkounis E V, Piliouras G. Poincaré recurrence, cycles and spurious equilibria in gradient-descent-ascent for non-convex non-concave zero-sum games. In: Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2019, 10450–10461

19 Gidel G. Multi-player games in the era of machine learning. PhD Thesis. Montréal: Université de Montréal, 2021

20 Gidel G, Berard H, Vignoud G, et al. A variational inequality perspective on generative adversarial networks. In: International Conference on Learning Representations. Washington: ICLR, 2019
21 Gidel G, Hemmat R A, Pezeshki M, et al. Negative momentum for improved game dynamics. In: International Conference on Artificial Intelligence and Statistics. New York: PMLR, 2019, 1802–1811
22 Gidel G, Jebara T, Lacoste-Julien S. Frank-Wolfe algorithms for saddle point problems. In: International Conference on Artificial Intelligence and Statistics. New York: PMLR, 2017, 362–371
23 Goodfellow I. NIPS 2016 tutorial: Generative adversarial networks. arXiv:1701.00160, 2016
24 Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2014, 2672–2680
25 Grnarova P, Kilcher Y, Levy K Y, et al. Generative minimization networks: Training GANs without competition. arXiv:2103.12685, 2021
26 He H, Zhao S F, Xi Y Z, et al. AGE: Enhancing the convergence on GANs using alternating extra-gradient with gradient extrapolation. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications. Cambridge: MIT Press, 2021
27 Heusel M, Ramsauer H, Unterthiner T, et al. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2017, 6629–6640
28 Hsieh Y P. Convergence without convexity: Sampling, optimization, and games. PhD Thesis. Lausanne: École Polytechnique Fédérale de Lausanne, 2020
29 Jin C. Machine learning: Why do simple algorithms work so well? PhD Thesis. Berkeley: University of California Berkeley, 2019
30 Jin C, Netrapalli P, Jordan M I. What is local optimality in nonconvex-nonconcave minimax optimization. In: International Conference on Machine Learning. New York: ICML, 2020, 4880–4889
31 Kingma D P, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014
32 Lei N, An D S, Guo Y, et al. A geometric understanding of deep learning. Engineering, 2020, 6: 361–374
33 Liang T Y, Stokes J. Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In: International Conference on Artificial Intelligence and Statistics. New York: PMLR, 2019, 907–915
34 Lin T Y, Jin C, Jordan M I. On gradient descent ascent for nonconvex-concave minimax problems. In: International Conference on Machine Learning. New York: ICML, 2020, 6083–6093
35 Liu M R, Rafique H, Lin Q H, et al. First-order convergence theory for weakly-convex-weakly-concave min-max problems. JMLR, 2021, 22: 1–34
36 Lorraine J, Acuna D, Vicol P, et al. Complex momentum for learning in games. arXiv:2102.08431, 2021
37 Lv W, Xiong J, Shi J, et al. A deep convolution generative adversarial networks based fuzzing framework for industry control protocols. J Intell Manuf, 2021, 32: 441–457
38 Mertikopoulos P, Papadimitriou C, Piliouras G. Cycles in adversarial regularized learning. In: ACM-SIAM Symposium on Discrete Algorithms. Philadelphia: SIAM, 2018, 2703–2717
39 Mertikopoulos P, Zenati H, Lecouat B, et al. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In: International Conference on Learning Representations. Washington: ICLR, 2019
40 Mescheder L, Geiger A, Nowozin S. Which training methods for GANs do actually converge? In: International Conference on Machine Learning. New York: ICML, 2018, 3481–3490
41 Mescheder L, Nowozin S, Geiger A. The numerics of GANs. In: Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2017, 1825–1835
42 Mishchenko K, Kovalev D, Shulgin E, et al. Revisiting stochastic extragradient. In: International Conference on Artificial Intelligence and Statistics. New York: PMLR, 2020, 4573–4582
43 Mokhtari A, Ozdaglar A, Pattathil S. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In: International Conference on Artificial Intelligence and Statistics. New York: PMLR, 2020, 1497–1507
44 Nagarajan V, Kolter J Z. Gradient descent GAN optimization is locally stable. In: Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2017, 5585–5595
45 Nedić A, Ozdaglar A. Subgradient methods for saddle-point problems. J Optim Theory Appl, 2009, 142: 205–228
46 Nouiehed M, Sanjabi M, Huang T J, et al. Solving a class of non-convex min-max games using iterative first order methods. In: Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2019, 14934–14942
47 Odena A. Open questions about generative adversarial networks. Distill, 2019, 4: e18
48 Peng W, Dai Y H, Zhang H, et al. Training GANs with centripetal acceleration. Optim Methods Softw, 2020, 35: 955–973
49 Pinetz T, Soukup D, Pock T. What is optimized in Wasserstein GANs. In: Proceedings of the 23rd Computer Vision

Winter Workshop. New York: IEEE, 2018
50 Qu Y Y, Zhang J W, Li R D, et al. Generative adversarial networks enhanced location privacy in 5G networks. Sci China Inf Sci, 2020, 63: 220303
51 Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434, 2015
52 Ryu E K, Yuan K, Yin W T. ODE analysis of stochastic gradient methods with optimism and anchoring for minimax problems and GANs. arXiv:1905.10899, 2019
53 Salimans T, Goodfellow I, Zaremba W, et al. Improved techniques for training GANs. In: Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2016, 2234–2242
54 Saxena D, Cao J. Generative adversarial networks (GANs): Challenges, solutions, and future directions. ACM Comput Surv, 2021, 54: 1–42
55 Shen J Y, Chen X H, Heaton H, et al. Learning a minimax optimizer: A pilot study. In: International Conference on Learning Representations. Washington: ICLR, 2020
56 Vondrick C, Pirsiavash H, Torralba A. Generating videos with scene dynamics. In: Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2016, 613–621
57 Xu Z, Zhang H L. Optimization algorithms and their complexity analysis for non-convex minimax problems (in Chinese). Oper Res Trans, 2021, 25: 74–86 [徐姿, 张慧灵. 非凸极小极大问题的优化算法与复杂度分析. 运筹学学报, 2021, 25: 74–86]
58 Xu Z B, Yao X. Advances in Data Intelligence, 2nd ed (in Chinese). Shanghai: Shanghai Jiao Tong Univ Press, 2021 [徐宗本, 姚新. 数据智能研究前沿 (第二版). 上海: 上海交通大学出版社, 2021]
59 Yazdandoost-Hamedani E, Jalilzadeh A, Aybat N S, et al. Iteration complexity of randomized primal-dual methods for convex-concave saddle point problems. arXiv:1806.04118, 2018
60 Yuan Y X, Bai Y Q, Chen J W, et al. Chinese Discipline Development Strategy·Mathematical Optimization (in Chinese). Beijing: Science Press, 2020 [袁亚湘, 白延琴, 陈加伟, 等. 中国学科发展战略 · 数学优化. 北京: 科学出版社, 2020]
61 Zhang G J. Understanding minimax optimization in modern machine learning. PhD Thesis. Ontario: University of Waterloo, 2021
62 Zhang G J, Yu Y L. Convergence behaviour of some gradient-based methods on bilinear zero-sum games. In: International Conference on Learning Representations. Washington: ICLR, 2020
63 Zhang M, Lucas J, Ba J, et al. Lookahead optimizer: k steps forward, 1 step back. In: Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2019, 9597–9608

Training GAN with predictive centripetal acceleration

Keke Li, Xinmin Yang & Ke Zhang

Abstract To alleviate the issue of limit cycle behavior in training generative adversarial networks (GAN), in this paper, we draw inspiration from the centripetal acceleration algorithm and the modified predictive method (MPM) by Liang and Stokes (2019). Building upon geometric observation of uniform circular motion, we propose the predictive centripetal acceleration algorithm (PCA). First and foremost, we prove the last-iterate convergence of the PCA on the bilinear game, which is a special case of the GAN. Besides, by combining PCA with the stochastic gradient descent (SGD) algorithm and adaptive moment estimation (Adam) algorithm, we propose two variants, which are called stochastic PCA (SPCA) and PCA-Adam, for the practical training GAN. Last but not least, experiments conducted on the bilinear game, multivariate Gaussian distribution, and CIFAR10 and CelebA datasets validate the effectiveness of the proposed algorithm.

Keywords generative adversarial networks, predictive centripetal acceleration algorithm, bilinear game, convergence

MSC(2020) 90C26, 90C29, 90C30, 90C46

doi: 10.1360/SSM-2022-0237