23 机器学习前沿青年科学家

杜克大学鬲融：通往“Learning to learn”方法的理论理解

整理：智源编辑 许明英

近年来，人工智能的蓬勃发展促进了人们对人工智能理论的深入探索，人工智能理论的研究呈现出了 Artificial Intelligence --> Machine Learning --> Deep Learning --> Deep Reinforcement Learning --> Deep Learning to Learn 的趋势。Learning to Learn(学会学习) 已经成为继增强学习之后又一个重要的研究分支。

在 Machine Learning 时代，复杂的分类问题推动了人们对 Deep Learning（深度学习）的探索，深度学习的出现基本解决了一对一映射问题，然而深度学习在解决 Sequential decision making 问题上遇到了瓶颈，由此深度增强学习应运而生，并在序列决策问题上初显成效。但是，新的问题接踵而至，深度增强学习依赖于巨量的训练，并且需要精确的 Reward，对于现实世界的很多任务，没有好的 Reward，也没办法无限量训练。这就需要其能够快速学习。而快速学习的关键是具备学会学习的能力，能够充分的利用以往的知识经验来指导新任务的学习，因此 Learning to Learn 成为学者们新一轮攻克方向。

6 月 24 号，在第二届智源大会“机器学习前沿青年科学家”专题论坛上，杜克大学计算机科学系鬲融教授作为演讲嘉宾，带来了主题为《Towards a Theoretical Understanding of Learning-to-learn Methods》的精彩演讲。

鬲融在报告中，首先就深度学习中起核心作用的优化算法抛出第一个问题： 如何训练及优化网络，仅仅使用 SGD 或 Adam 足够吗？

他简单阐述了训练神经网络的一些技巧，例如可能需要设计步长、改变一些动量； 可能需要增加一些权重衰减，增加数据量； 可能需要利用各种各样的技巧去优化网络。然而，调整这些参数优化网络并不是一件容易的事情。

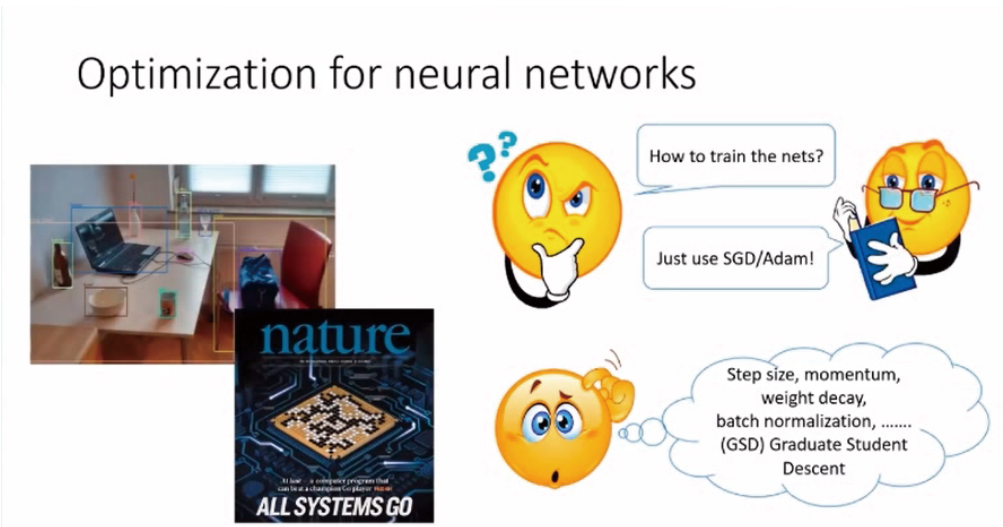


图 1： 神经网络优化

接下来，高融提到或许在调参过程中会非常沮丧，或许想摆脱这些繁杂的调参过程，自动找寻新的优化技巧。那么，这样做有没有可行性呢？答案是肯定的。

这方面的研究工作目前已经有很多，其中利用 Learning to learn 来设计更好的优化算法，从而来提高优化器性能是其中一个方向。高融以论文《Learning to learn by gradient descent by gradient descent》为例进行了介绍，这篇论文的主要思想是用 Learning to learn 方法学习一个新的优化器，目标是优化分配任务的目标函数 f(w)；具体则是，将优化算法抽象为具有参数Θ的优化器，然后通过各个分配任务优化参数Θ。

图 2：具有参数Θ的优化器

优化器可以是传统简单的优化器，也可以是神经网络优化器。训练优化器的步骤为： 进行 t 步优化、定义元目标、在优化器参数Θ做元梯度下降。事实上，这一个过程类似于循环神经网络 / 策略梯度。

然而这一过程会面临着诸多挑战，例如梯度消失或梯度爆炸问题、可能陷入较差的局部最优解、在具体任务上的泛化能力、没有理论保证等。高融在报告中谈到自己为二次目标分析了简单的优化器（包括梯度下降 GD 和随机梯度下降 SGD），并通过实践得出了一些结论如下：

1. 对于二次目标的梯度爆炸 / 梯度消失问题

(1) 传统的元目标对于所有步长都存在元梯度爆炸 / 消失问题；
(2) 可以设计一个更好的元目标，其元梯度保持多项式有界；
(3) 即使对于新目标，使用反向传播算法计算元梯度也会导致数值问题。

2. 最小二乘训练优化器的泛化能力

当样本数量较少时，需要在单独的验证集上定义元目标。当样本数量很大时，只需在训练集上定义元目标即可。

高融从步长和设计更好的目标两个方面入手探讨了应对梯度爆炸 / 梯度消失问题的策略。

一、为简单的二次目标优化步长

目标 :

$$minf\left(w\right)=\frac{1}{2}w^{T}Hw$$

算法 : 使用固定步长的梯度下降法:

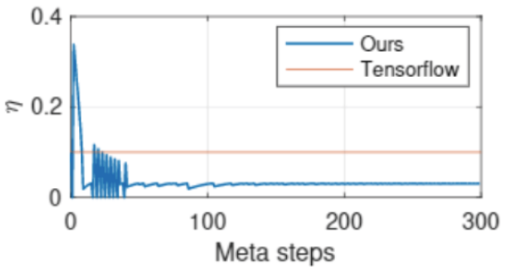
$$w_{t+1} = w_t - \eta \nabla f\left(w_t\right) = \left(I - \eta H\right) w_t$$

传统的元目标在最后一步的损失为:

$$F\left(\eta\right) = f\left(w_{\eta,\ T}\right)$$

定理: 对于η的几乎所有值，T 中的元梯度 $F'\left(\eta\right)$ 要么呈指数增长，要么呈指数下降。

鬲融通过实验展示了 TensorFlow 计算的实际元梯度与元梯度的训练轨迹（T = 80，初始步长为 0.1），以及成功学习不同迭代次数 T 时的最佳步长。如下图所示:



t	10	20	40	80
Ours	✓	✓	✓	✓
Tensorflow GD	×	×	×	×
Tensorflow RMSProp	✓	✓	×	×

图 3：训练优化器的泛化能力

二、设计一个更好的目标

思想: 因为目标在 T 中成倍地变大或变小，导致元梯度很大。因此设计一个新的目标如下:

$$G\left(\eta\right) = \frac{1}{T}\log f\left(w_{\eta,T}\right) = \frac{1}{T}\log F\left(\eta\right)$$

定理: 对于新目标，在所有相关参数中，元梯度 $G'\left(\eta\right)$ 总是多项式。此外，步长为 $1/\sqrt{k}$ 的元梯度下降收敛。然而，如果用反向传播计算 $F'\left(\eta\right)$，需要 $G'\left(\eta\right) = \frac{dF}{dG}$。$F'\left(\eta\right)$ 以及 $G'\left(\eta\right)$ 以指数形式变大或变小。

设置: 最小二乘问题

$$y = w_*^T x + \xi, w_* = 1, x \sim N\left(0, I_d\right), \xi \sim N\left(0, \sigma^2\right)$$

目标: 训练数据的平方损失

$$f(w) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - w^{\top}x_i)^2$$

算法：恒定步长的梯度下降（与 SGD 相似）

$$w_{t+1} = w_t - \eta\nabla f(w_t)$$

高融介绍了定义元目标的两个思想，给定 $w_{\eta,T}$ 为步长为η时迭代第 T 次的点。

(1) 使用 Train–by–Train(TyT) 方法，在训练集上定义原目标，例如，简单选择 F(η)=f($w_{\eta,T}$)。

(2) 使用 Train–by–validation 方法，在分开的验证集 $\left(x_1^{'}, y_1^{'}\right)\ldots\left(x_{n_2}^{'} \quad y_{n_2}^{'}\right)$ 上定义

$$G(\eta) = \frac{1}{2n_2}\sum_{i=1}^{n_2}\left(y'_i - w_{\eta,T}^{\top}x'_i\right)^2$$

高融就何时使用 Train–by–validation（TBV）方法总结如下：

定理： 当σ是一个足够大的常数，并且 n（样本数）是 d（维度）的恒定分数时，逐次验证效果更好；当 n（样本数）比 d（维度）大得多时，则逐列训练接近于最小 – 最大最佳解。

观察：神经网络经常被过度的参数化，这意味着样本数 n 小于维度 d。

高融给出了实验验证的结果如下，在原始最小二乘数模型上，高融分别比较了在使用 Train–by–Train(TBT) 方法和 Train–by–validation（TBV）两种方法时，不同步长下，在训练集和测试集上的均方根误差 RMSE。结果如下所示：

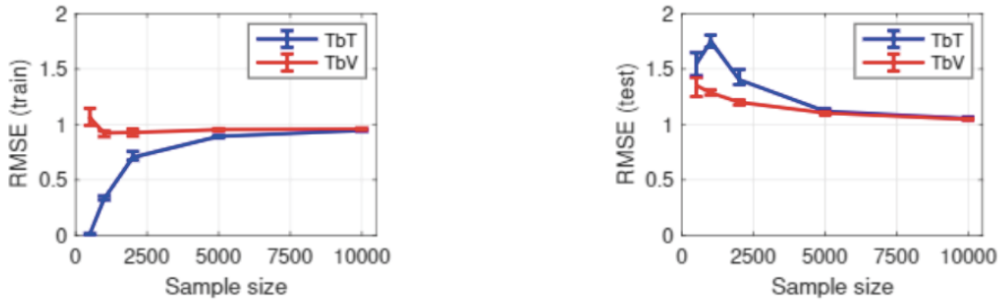


图 4：原始最小二乘数模型上的均方误差值

高融通过在合成数据如 MNIST 等数据上的简单实验验证阐述了使用 Train–by–Train(TyT)、Train–by–validation（TBV）训练优化器的结果。并得出结论，使用验证损失可以实现良好的泛化性能，而使用训练损失甚至对于简单的二次函数也可能过拟合。

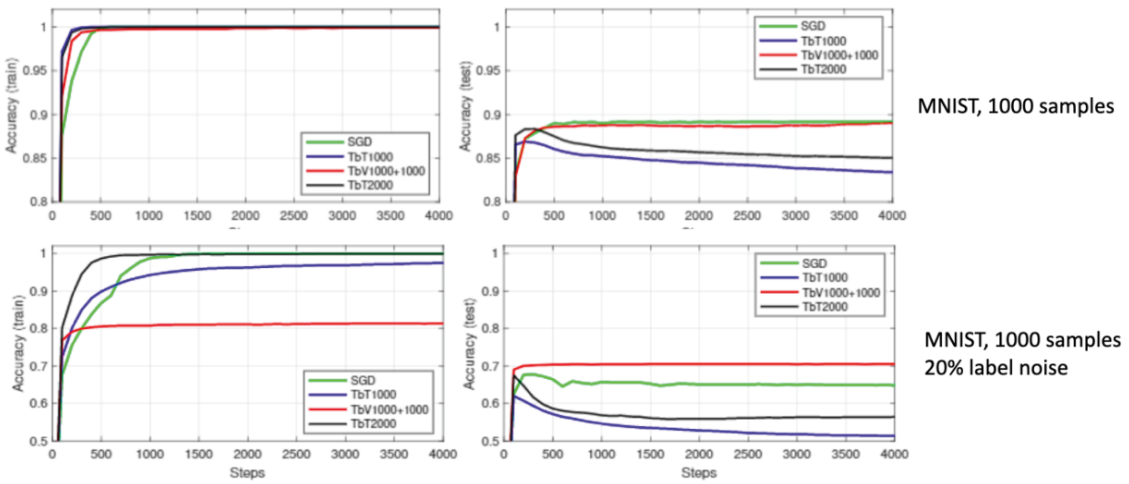


图 5：在网络优化器上的实验观测

鬲融总结了优化过程中的几个注意事项:

(1) 仔细选择元目标可以减轻梯度爆炸 / 消失的问题;
(2) 需要谨慎使用反向传播算法 ;
(3) 当样本较少或者噪声较大时，需要在单独的验证集上定义元目标。

最后，鬲融就进一步的研究提出了建设性的问题“是否可以为神经网络优化器缓解梯度爆炸、梯度消失问题以及数值问题呢?”“是否可以针对更复杂的目标为更复杂的优化器调整参数?”引发大家的进一步思考。

普林斯顿大学金驰：对强化学习算法复杂度的一种优化方法

整理：智源社区 何灏宇

在第二届北京智源大会“机器学习前沿青年科学家”专题论坛上，普林斯顿大学助理教授，青年科学家金驰做了题为《Near-Optimal Reinforcement Learning with Self-Play》的报告。

在报告中，金驰提到：低效是现有的大多数强化学习算法的瓶颈，也是制约强化学习大规模应用的一个关键点，这其中，有两种效率我们需要去考虑，一种是采样效率，即在训练时需要进行取样的样本数量；另一种是计算效率，即训练模型需要花费的时间和算力。

在本次报告中，金驰为我们展示了他在解决强化学习算法的低效性尤其是提高采样效率这方面的研究成果。以下是演讲全文，本文做了不改变原意的整理。

关于强化学习的一个里程碑事件是 Alpha Go 在人机大战中战胜围棋世界冠军，这也是机器第一次在围棋这个项目中战胜人类的最高水平。像 Alpha Go 这样的基于强化学习的人工智能有一个特点——它们并不是通过与人对战进行训练而是通过自己与自己对战进行学习的。这样的概念被称作自学习（Self-Play）。

现有的大多数强化学习算法都陷入了低效的瓶颈。比如在谷歌公司有充足算力资源的情况下，训练 Alpha Go Zero 需要采样一千万盘围棋比赛，用时一个多月。在像围棋或者扑克或者是星际争霸这些游戏中，玩家数量不是一个而是多个，且对手的策略会通过分析其他玩家的策略进行调整，这种形式叫做双（多）玩家零和博弈。零和博弈的意思是如果某一玩家获得奖励或者赢得博弈，那就意味着其他玩家获得惩罚或是输掉博弈。使用强化学习算法处理零和博弈问题时，也同样会遇到效率低下的问题。那么，对于双玩家零和博弈，怎样设计一种更有效的强化学习算法使得采样效率和计算效率更高呢？这正是本次报告的主要内容。

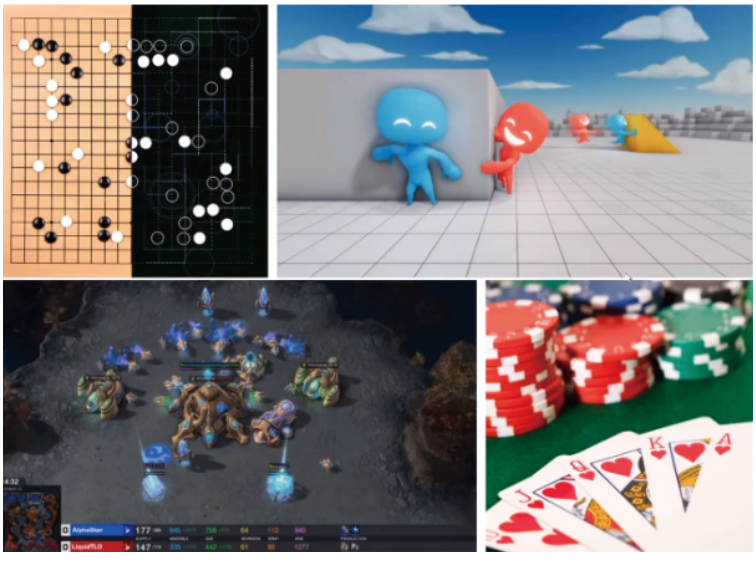


图 1：多玩家零和博弈在现实生活中的案例

一、极端还是均衡，哪个才是最优策略

在讨论怎样解决这个问题之前，首先需要说明我们使用的是马尔可夫对策模型（Markov Games），假设在该模型中有两个智能体 a 和 b，在一个状态下，a 和 b 会采取行动 a_1、b_1 或者 a_2、b_2，并且分别得到奖励值 r，同时，整个模型的状态 s 也会根据 a、b 的行动发生改变。那么，对于该模型，我们做一下参数的定义，S 代表模型所有状态的集合，A 代表智能体 a 所有行动的集合（$\{a_1, a_2, ..., a_n\}$），同理 B 代表智能体 b 所有行动的集合。H 代表整个博弈中视野的长度，比如星际争霸中一场游戏需要经过的时间。非常重要的一点是，两个智能体 a、b 是互相比赛的，其中一个智能体的目标是最大化系统的奖励值 r，另一个是最小化 r。

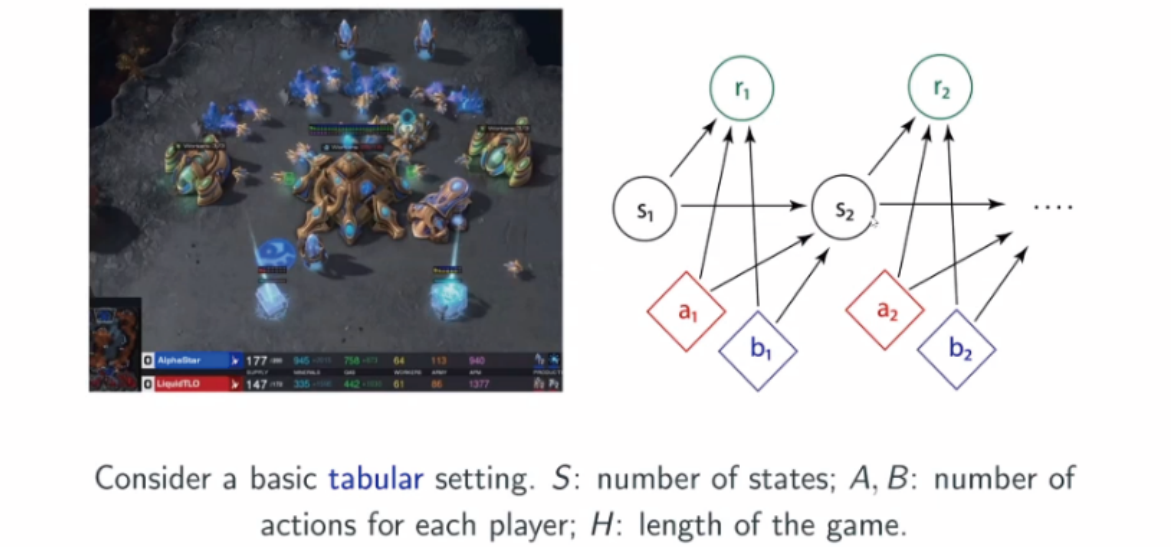


图 2：马尔科夫对策模型

在设计算法之前我们需要构建出一个对于双玩家零和博弈的最优策略。那么，怎样去理解最优策略呢？首先需要介绍几个概念，第一个概念称作最佳响应（Best response），在响应时会去尽可能探索对手的策略并且采取最优的行为。比如在剪刀石头布中，如果本玩家的策略是一直出石头，那么最佳响应策略就会一直出包袱来对抗本玩家。然而在实际的博弈过程中，任何一个玩家都不能准确的知道对手的策略是什么。第二个概念叫做纳什均衡（Nash Equilibria），纳什均衡的策略是，在任何时刻都做最坏的打算，不论其他玩家的策略如何改变，哪怕其他玩家采取针对本玩家的最佳响应策略 ，采用纳什均衡的玩家依然可以保证一定的收益。比如在剪刀石头布时均匀出拳，那么本玩家可以始终保持百分之五十的获胜概率。事实上，这种策略在一些场景下是非常强大的，想象一下如果你在玩星际争霸的时候与任何对手对战都能保证百分之五十的胜率，那么你就很有可能会是这个游戏世界中最好的玩家。因此我们的目标就是找到一个取样少采样效率高的纳什平衡，使得模型对于参数 S、A、B 的值有最优依赖。

二、基于置信度进行高效的搜索

现在我们明确了我们想要实现的目标，接下来就该开始设计算法了。用到的基础算法叫做 Nash Q-learning，对于熟悉 Q-learning 的人来说，可以把 Nash Q-learning 当作是双玩家版本的 Q-learning。基础的 Nash Q-learning 会做两件事情，一是随机更新 Q 值，二是根据预测的 Q 值重新计算纳什平衡，从而达到更新策略的目的。如果仔细思考一下 Nash Q-learning 的话，会发现它并没有详细说明怎样去对数据进行采样，这就引出

了 一个很重要的问题——搜索 (Exploration)。在强化学习中，不能只去利用 (Exploitation) 那些已经有的或者模型认为最优的数据，也要去探索那些模型认为并不是最优但实际可能就是最优的数据。这样做的原因是因为采样次数有限，模型根据学习到的策略计算出的最优样本和实际的最优样本之间会有偏差，通过搜索能够保证模型不仅局限于学习到的策略，而是对所有样本都进行尝试，这就是搜索的意义。𝜖–greedy 是一个常用的搜索方法，它表示每次采样数据时，模型有 𝜖 的概率去进行随机的搜索，其他时刻模型做贪婪搜索，从而保证所有可能的样本都能被取到。

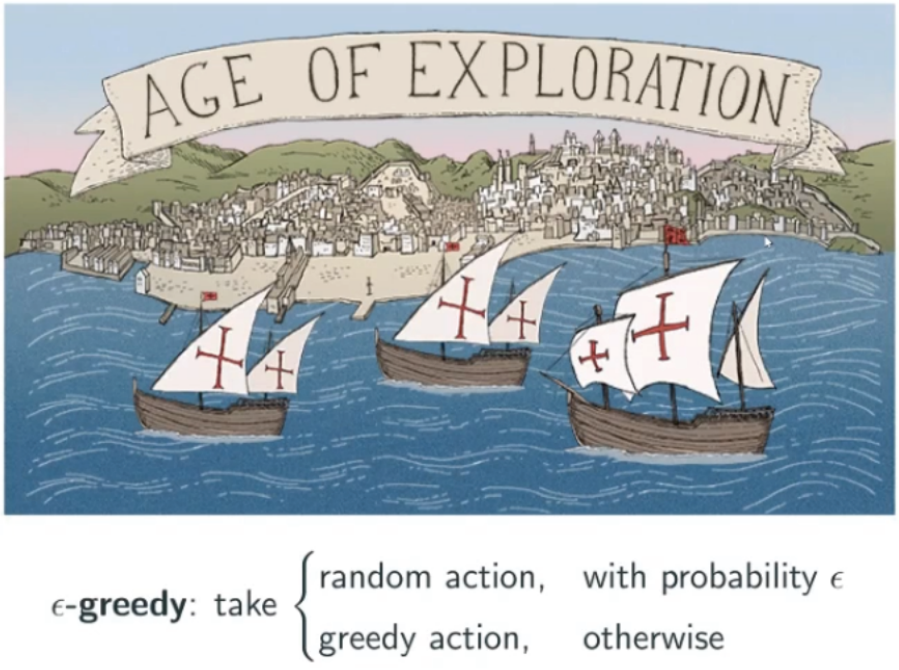


图 3： ϵ–greedy 的抽象表达 0

如果有无限数量的样本可以不停做搜索，ϵ–greedy 会非常有效，但是在实际的应用中样本数和时间都是有限的，导致 ϵ–greedy 搜索到的最优策略与真实值之间总会有一个差值，所以我们需要找到一个比 ϵ–greedy 更有效的算法——上置信界算法 (Upper Confidence Bound)。该算法的核心思想是假设每一个样本都有一个置信区间，这个置信区间代表模型对采样该样本时可能得到的奖励值的预估区间，每个样本都对应着一个奖励值的期望μ ，在下图中表示为μ_1 和μ_2。在每次采样时，取置信上界最高的样本，根据得到的奖励值更新 μ 。随着对样本的采样次数增多，模型对该样本的置信度增加，样本的置信区间会缩小。概括来说，模型会在每次采样的时候选择看起来有更高置信区间上界的样本，也就是更有可能拿到高奖励的样本，并且在每次采样之后对置信区间进行更新。该算法已被证明要比 ϵ–greedy 算法的效果好得多，对于强化学习的学习效果提升是巨大的。

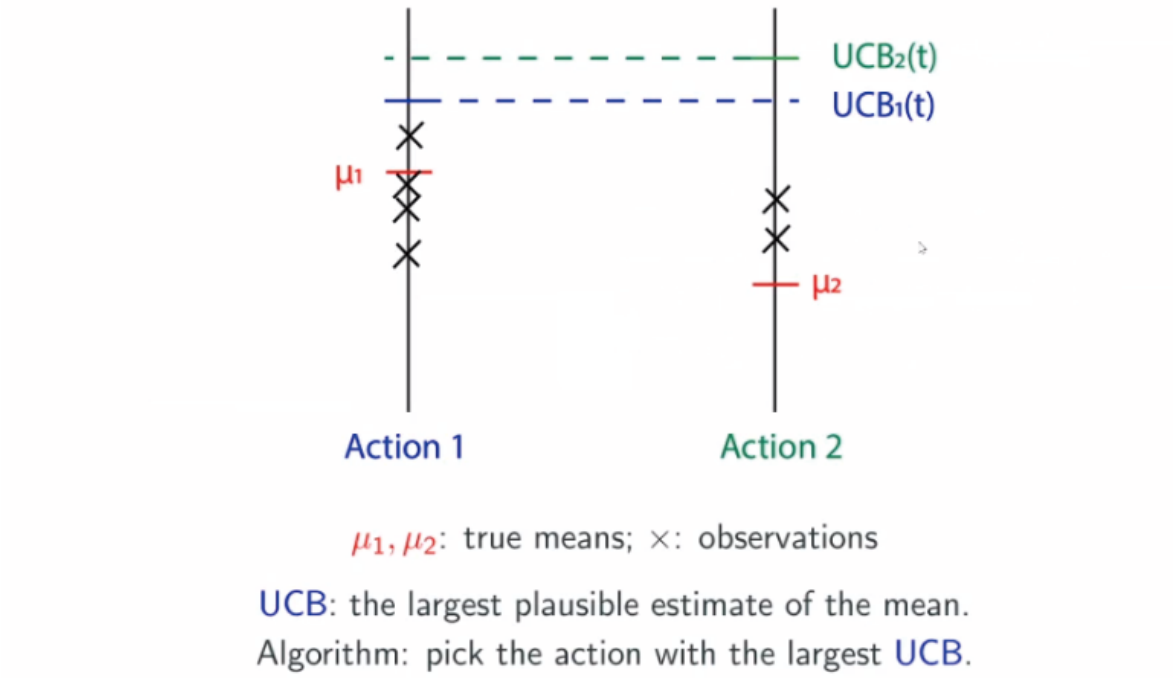


图 4：上置信区间算法

由于我们讨论的是双玩家的零和博弈，因此我们不仅使用上置信区间算法，也同时采用下置信区间算法（Lower Confidence Bound）。两个玩家分别使用两种算法进行采样。然而这会带来一个新的问题： 两个玩家采用不同的算法意味着算法中要计算两次纳什均衡，这会导致算法的计算复杂度为 PPAD（高于多项式级），显然这样的结果与我们提高算法运行效率的初衷不符。为了解决这个问题，我们采用了粗相关均衡（Coarse Correlated Equilibrium）代替纳什均衡从而达到更低的计算复杂度。

现在我们已经完成了大部分的算法设计，这样的算法已经能够在单玩家的场景下执行最佳策略。然而我们意识到一个问题，那就是在多玩家的前提下，这种优化后的 Nash Q-learning 算法只能保证在与采用纳什均衡策略的对手博弈时，取得良好的效果。如果对手采用最佳响应的策略，我们的算法策略不能保证取得好的效果，这样的一个策略不能算是一个优秀的策略。为了解决这个问题，我们在算法中加入了一个新的机制 Certified Policy，使得我们的策略能够保证在与最佳响应策略进行博弈时也能有好的表现。因为时间原因，我不会在这里讲解更多关于应用这个机制的原理。

三、算法复杂度分析

以上就是我们设计最优 Nash Q-learning 算法的全部过程，下图展现了该算法采样复杂度的上界和下界，可以看到参数 S、A、B 和 ϵ 对于该算法的依赖都是最优的（线性依赖）。也能看到我们设计的另一个算法——最优 Nash V-learning，在最优 Nash Q-learning 算法的基础上又进一步降低了采样复杂度。

Theorem [BJY20]

If we run optimistic Nash Q-learning for $\tilde{O}(H^5SAB/\epsilon^2)$ episodes, then the extracted policies will be an ϵ-approximate Nash equilibrium.

S: number of states, A, B: number of actions for each player, H: length of the game.

Lower bound: $\Omega(H^3S(A+B)/\epsilon^2)$ episodes.

Our new algorithm—optimistic Nash V-learning: $\tilde{O}(H^6S(A+B)/\epsilon^2)$.

图 5：Optimistic Nash Q–learning 的采样复杂度

我们设计的两种优化算法与其他算法的效率对比，可以明显的看到 Nash Q–learning 和 Nash V–learning 对于参数 S、A、B 的依赖都是最优的，达到了线性依赖。以实际情况为例做说明，在实际应用中，通常会有超过 1000 个状态 S，那么我们算法将采样复杂度对 S 的依赖从 S^2 降低到 S 则会使取样的时间比之前降低一千多倍，这种优化对于效率的提升是指数级的，这也是为什么我们希望能设计一个达到采样复杂度下界（Lower Bound）的算法的原因。

Algorithm	Sample Complexity	Runtime
Most Algorithms with ϵ-Greedy	$\Omega((A+B)^H)$	-
VI-ULCB [BJ20]	$\tilde{O}(H^4S^2AB/\epsilon^2)$	PPAD-complete
VI-explore [BJ20]	$\tilde{O}(H^5S^2AB/\epsilon^2)$	Polynomial
OMVI-SM [XCWY20]	$\tilde{O}(H^4S^3A^3B^3/\epsilon^2)$	
Optimistic Nash Q-learning	$\tilde{O}(H^5SAB/\epsilon^2)$	
Optimistic Nash V-learning	$\tilde{O}(H^6S(A+B)/\epsilon^2)$	
Lower Bound [BJ20]	$\Omega(H^3S(A+B)/\epsilon^2)$	-

Typical applications: $S \geq 1000; A, B \geq 10$.

图 6：优化算法之间的效率对比

四、更多发展方向

最后，对本次报告做一个总结，我们设计了一个在双玩家零和博弈中能够达到近似最优效率的自学习强化学习算法，基于这个算法我们还可以在未来有更多的研究方向:

1. 最重要的一点，虽然算法对于 S 的依赖是线性的，但如果 S 值特别大，算法的效率也不能得到保证。我们之前认为 S 是在 1000 这个数量级上变化的，因此从 S^2 到 S 的优化实际上是把采样复杂度对 S 的依赖从百万降低到千。但是在星际争霸这样的场景下，S 本身就有几百万个，这样的情况下，采样复杂度对于 S 的线性依赖还是会给计算带来灾难。这个问题是需要通过研究去解决的。

2. 尽管算法的复杂度对于 A、B、S、ϵ 的依赖都是最优的，但是对于 H 的依赖还不是最优，如何对 H 做优化是一个值得研究的方向。

3. 还需要将该算法应用于实际使用场景，以实际表现去评价算法的有效性。

宾夕法尼亚大学苏炜杰：隐私算法到底有多隐私？

整理：智源社区 熊宇轩

随着人工智能学科的蓬勃发展，以及深度学习等技术在社会生活中的广泛应用，算法的安全性问题又重新被人们所重视。在本届智源大会的“机器学习前沿青年科学家”专题论坛上，来自宾夕法尼亚大学的助理教授苏炜杰为我们带来了主题为「how PRIVATE are PRIVATE algorithms」的报告。该报告介绍了差分隐私保护的发展历史，并从信息量、复合、子采样这三个角度分析了他们近期提出的 f-DP 相对于传统差分隐私保护框架的优势。以下为苏炜杰演讲内容:

在今天的演讲中，我们将讨论一种新的隐私保护的机器学习框架。

在 George Orwell 著名的小说《1984》中，「Big Brother」是大洋国的独裁者，他可以窥探其国家内部所有人的隐私，而由于每个人都没有隐私，这个国家最后覆灭了！

不幸的是，George Orwell 在小说中描述的场景在今天正在变为现实。现在的大型企业（尤其是 IT 企业）可以获知用户的隐私数据，如果这些隐私数据被用于恶意用途，就会对每个人甚至是整个社会带来灾难。

那么我们如何应对这一问题呢？仅仅通过匿名化（从数据集中删除用户的名字）手段就能够保护隐私吗？很不幸，这还远远不够！因为属于某个人的数据可能出现在多个数据集中。

实际上，Narayanan 和 Shmatikov 等人于 2006 年发表的论文指出，可以通过将两个以上的数据集联系起来，从而识别出特定用户的身份。这也正是著名的 Netflix 竞赛被取消的原因。

另一方面，我们可以公开总结的统计量（样本均值）吗？很不幸，通常而言，这种数据也是存在泄露隐私的隐患。2008 年，Homer 等人指出，如果我们公开次等位基因频率（MAF）的均值，你可以确认判断某个人是否在这个数据集中。试想，如果某个数据集包含的是糖尿病人的数据，你可以通过判断某人是否在该数据集中从而获知他是否患有糖尿病，此时隐私也就荡然无存了。

一、差分隐私保护

那么，这就是我们的未来吗？好消息是，一些研究人员在隐私保护领域做出了积极探索。2006 年，计算机领域的研究人员提出了差分隐私保护技术（DP），将隐私保护与假设检验联系在了一起。

下面，我们对 DP 进行形式化定义。假设我们知道数据集的所有信息，并且知道数据集中存在 Jane、Ed、Bob 三名用户，但我们现在不知道 Anne 或 Eva 是否在数据集中。此时就有两种可能性，S={Anne,Jane,Ed,Bob} 和 S’={Eva,Jane,Ed,Bob}，由于这两个集合只有一个元素不同，我们将它们称为「相邻数据集」。

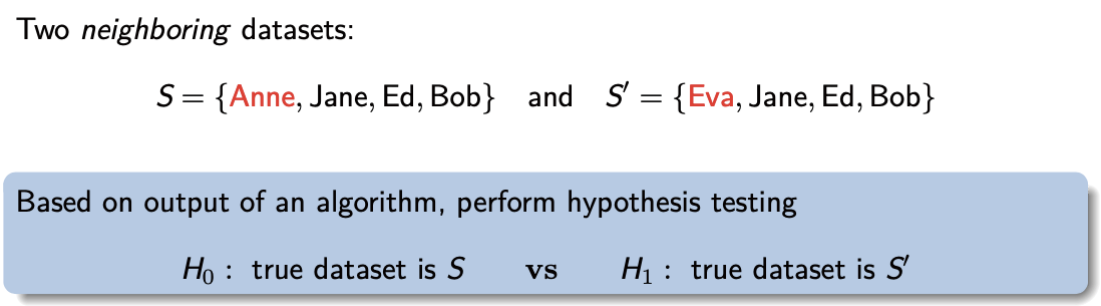


图 1：差分隐私保护

我们的问题是，是否能够基于某种算法，识别出这个不确定的元素究竟是 Anne 还是 Eva。我们将真实数据集 S 记为 H_0（对应于原假设），将真实数据集为 S’记为 H_1（对应于备择假设）。

实质上，H_0 代表 Anne 在数据集中，H_1 代表 Eva 在数据集中。那么，直观上来说，如果这种假设检验很难实现，那么 Anne 和 Eva 的隐私就得到了保护，这就是差分隐私保护的基本思想。

近年来，DP 技术带来了巨大的影响，包括 Google、Apple、微软在内的各大企业纷纷采用了这项技术。同时，美国人口普查局也承诺在调查中使用 DP 技术来保护最重要的统计数据。2017 年，四名计算机科学家也由于 DP 的相关工作获得了理论计算机科学界的最高奖项：哥德尔奖。

本次演讲将基于 4 篇论文展开，有兴趣的读者可以在 Arxiv 上找到其中 3 篇的原文，最后一篇也会很快与大家见面。这里需要提一下我的合作者董金硕。他是一名非常有创造力的学生，在这一系列工作中起到了极大的推动作用。

- Gaussian Differential Privacy. With Dong and Roth. JRSSB (with discussion)
- Deep Learning with Gaussian Differential Privacy. With Bu, Dong, and Long. In submission
- Sharp Composition Bounds for Gaussian Differential Privacy via Edgeworth Expansion. With Zheng, Dong, and Long. ICML 2020
- Central Limit Theorem and Uncertainty Principles for Differentially Private Query Answering. With Dong and Zhang. In submission

首先，我们将对比一下本次演讲中提出的新的隐私保护思路与前人的做法。在本次演讲中，我们将提出一种名为「f– 差分隐私」(f–DP) 的新型隐私保护框架，而 Dwork 等人于大约 13 年前提出的框架被称为「(ε , δ)– 差分隐私」。

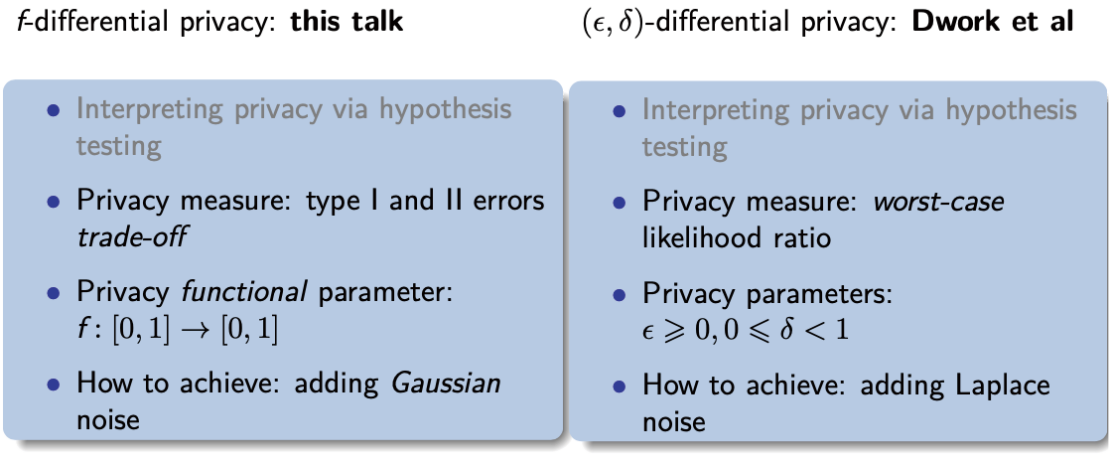


图 2：新的隐私保护思路

首先，它们的相似之处在于，它们都将隐私保护转化为了一个假设检验问题。但它们的差异体现在以下方面:

- f–DP 采用假设检验中的第一类错误（弃真错误）、第二类错误（取伪错误）作为隐私度量；而（ε，δ）–DP 则使用最差情况的似然比作为隐私度量，而这种最差情况在某种程度上说是过于悲观的。
- 由于我们的方法考虑弃真错误和取伪错误的折中，所以本质上说它是一种从区间 [0,1] 到 [0,1] 的函数映射。而在（ε，δ）–DP 中，他们使用仅仅使用了 ε 和 δ 两个数来定义最差情况下的似然比。
- 在 f–DP 框架下，典型的实现隐私保护的方式是加入高斯噪声（高斯分布），而（ε，δ）–DP 则是通过加入拉普拉斯噪声（双指数分布）实现隐私保护。

二、f–DP 简介

如今，市面上有各种各样的差分隐私保护方法，f–DP 也许是最新的一种差分隐私保护框架。在这里，我们考虑以下三种标准：(1) Informativeness (2) Composition (3) Subsampling。

	Informativeness	Composition	Subsampling
ϵ-DP	✗	✗	✓
(ϵ, δ)-DP	✗	✗	✓
Divergence based DPs	✗	✓	✗
f-DP	✓	✓	✓

图 3：差分隐私保护框架举例

我们的框架 f–DP 在上述三种评价标准上都取得了令人满意的性能。

$$H_0 : P \quad \text{vs} \quad H_1 : Q$$

For rejection rule $\phi \in [0,1]$, denote by $\alpha_\phi = \mathbb{E}_P[\phi]$ (type I error), and $\beta_\phi = 1 - \mathbb{E}_Q[\phi]$ (type II error)

Definition

For two probability distributions P and Q, define the trade-off function $T(P, Q) : [0,1] \to [0,1]$ *as*

$$T(P, Q)(\alpha) = \inf_\phi \{\beta_\phi : \alpha_\phi \leqslant \alpha\}$$

- Optimal ϕ given by the Neyman–Pearson lemma
- Trade-off functions are the most informative (Blackwell's theorem)

图 4：Trade–off 函数定义

在这里，我们令 H_0 为原假设，它表明真实情况为 S；H_1 为备择假设，它表明真实情况为 S'。我们用 P 表示真实值为「Anne」(S) 时算法输出结果的概率分布，P 表示真实值为「Eva」(S') 时算法输出结果的概率分布。此时的第一类错误由显著性水平 $\alpha_\phi = E_P[\phi]$ 定义，它代表原假设为真，但是我们拒绝了原假设，接受了备择假设。第二类额假设的概率 $\beta_\phi = 1 - E_Q[\phi]$，它代表备择假设为真，但是我们接收了原假设。

对于两个概率分布 P 和 Q 来说，我们定义其 trade–off 函数为一个从区间 [0,1] 到 [0,1] 的函数映射:

$$T(P, Q)(\alpha) = \inf_\phi \{\beta_\phi : \alpha_\phi \leq \alpha\}$$

其中，α 为 [0,1] 区间上的概率。该函数在 α 处的值是第二类错误可能的最小值，这样一来第一类错误就满足其概率 α_ϕ 小于 α。在满足该条件约束的情况下，你希望找到能够最好地使第二类错误概率最小的拒绝规则。此时，Neyman–Pearson 引理保证了始终存在最优的 ϕ，而根据 Blackwell 定理，我们的 trade–off 函数包含的关于假设检验问题的信息量最大。

f–DP 实际上反映了本次演讲的主题「隐私保护算法究竟有多隐私」。若某个随机的算法 M 对于所有的相邻数据集 S 和 S' 的 trade–off 函数满足:

$$T\big(\mathcal{M}(S), \mathcal{M}(S')\big) \geq f$$

我们则称该算法满足 f–DP。上式说明左侧函数的第二类错误始终大于等于右侧的 f。存在第二类错误意味着，将 Anna 与 Eva 区分开来并不比区分 P 和 Q 两种概率分布更简单。此时的随机性来自于算法，而并非来自于数据集，数据集是始终不变的。我们可以保证这里的 trade–off 函数 f 是关于第一象限的 45 度线 (y=x) 对称的，即 f(f(x))=x。

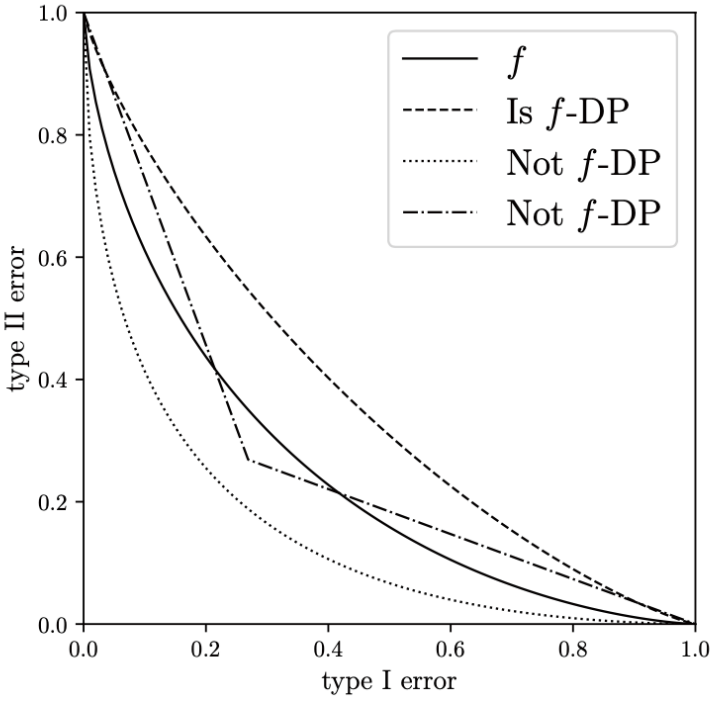


图 5：满足 f–DP 的算法示意图

如图 5 所示，黑色实线代表 f 的函数，只有“–––”的虚线满足 f–DP，因为根据定义，满足 f–DP 的算法的 trade–off 函数需要始终在 f 之上。

$$e^{-\epsilon}\,\mathbb{P}(\mathcal{M}(S')\in E)-e^{-\epsilon}\delta\leqslant\mathbb{P}(\mathcal{M}(S)\in E)\leqslant e^{\epsilon}\,\mathbb{P}(\mathcal{M}(S')\in E)+\delta$$

(ε , δ)–DP 需要满足如上图所示的不等式约束，当 ε 和 δ 都很小时，不等式最左边的一项和中间一项就相等了。实际上，通过定义 $f_{\epsilon,\delta}(\alpha)=\max\{0,1-\delta-e^{\epsilon}\alpha,\ e^{-\epsilon}(1-\delta-\alpha)\}$，我们发现 (ε , δ)–DP 是我们提出的 f–DP 的特例。

通过绘制这个函数的图形（分段函数），我们看到在 y=1 附近的截距为 δ，这是一个非常小的数，接下来的一段图像的斜率为 $-e^{x}$，然后在 y=x 另一侧的图像与之前绘制的图像对称。由此，我们就得到了四段函数图像。但是坚持使用 (ε , δ)–DP 方法所具有的局限性太大，使用概率 δ 也会带来一些不好的影响。这也正是我们提出 f–DP 的原因。

从更基础的「原始 – 对偶」的角度来看，我们可以通过使用不同的 (ε , δ) 对来描述算法的隐私性（ε 越小，则 δ 越大）。通过大量绘制各种 (ε , δ) 对的函数图像，我们得到了一条包络线，这条包络线代表 f–DP。

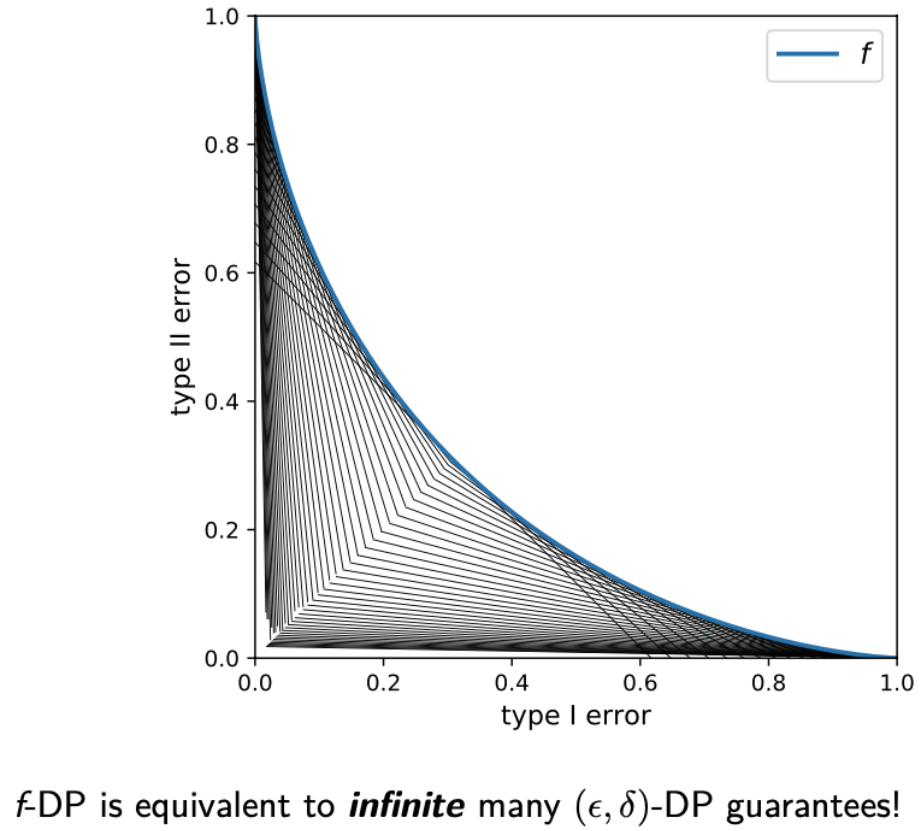


图 6：从原始对偶方法的角度看 f–DP

我们也可以反过来，对 f–DP 包络线取关于 y=x 对称的截距为 δ 的切线，不断重复这个过程就可以涵盖所有 (ε , δ)–DP 的情况。我们可以认为，f–DP 与 (ε , δ)–DP 等价当且仅当有无限对 (ε , δ)。

接下来，我们将考虑一种由正态分布得到的特殊的 trade–off 函数:

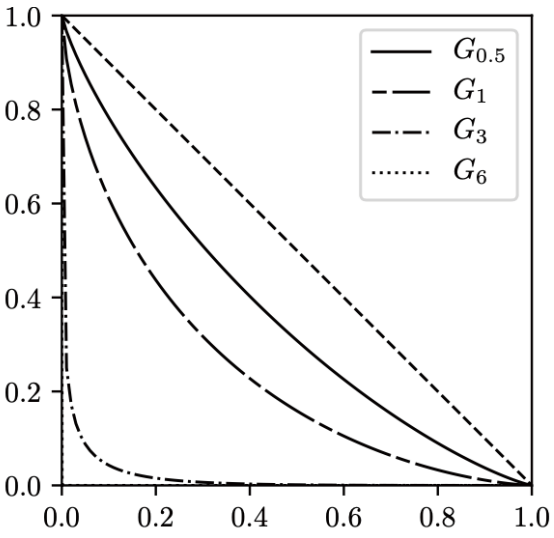
$$G_\mu := T\left(N(0,1),\quad N(\mu,1)\right)$$

其中，μ 为偏移后的均值。该函数的封闭解为 :

$$G_\mu(\alpha) = \Phi\left(\Phi^{-1}(1-\alpha) - \mu\right)$$

当 μ 小于完美隐私保护的要求时，你就无法区分 trade–off 函数中的两个分布，而如果 μ 过大 (例如超过 3 σ)，那么我们就可以区分出这两个分布，隐私就收到了威胁。

高斯差分隐私 (GDP) 是 f–DP 的一个子类，大体来说，对于所有的相邻数据集 S 和 S' 而言，该方法的 trade–off 函数值要一直大于等于由高斯 trade–off 函数给出的 G_μ，此时我们称算法 M 满足 μ –GDP。由于中心极限定理，以上结论对于 f–DP 来说是很重要的。



- Privacy amounts to distinguishing between $\mathcal{N}(0,1)$ and $\mathcal{N}(\mu,1)$
- $\mu \leqslant 0.5$: reasonably private; $\mu \geqslant 6$: blatantly non-private

图 7：对 GDP 中 μ 的解释

那么如何解释 GDP 中的参数（正态分布均值）μ 呢？当 μ 很小的时候（例如，μ =0.5），那么其曲线就与完美隐私下的状态十分接近；当 μ 很大时（例如，μ =6），其曲线就与坐标轴非常接近。此时，第一类错误与第二类错误都非常小，近乎于 0，毫无隐私可言。

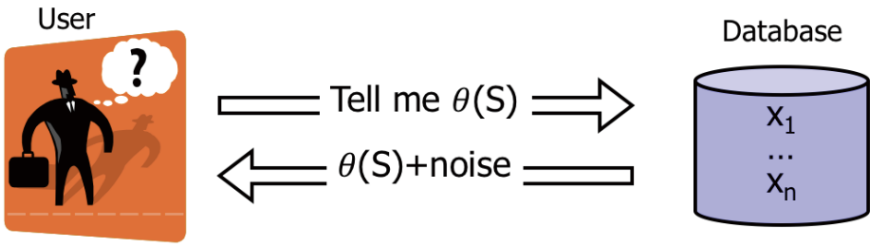


图 8：解决方案——加入噪声

一种减小 μ 的通用方法是加入噪声。如果你对统计量 θ (S) 的值感兴趣，为了保护隐私，我们可以向 θ (S) 加入隐私。对于 GDP 框架来说，我们向 θ (S) 加入高斯噪声。而至于加入多少高斯噪声则取决于我们希望有多大程度的隐私保护。另一方面，对于统计量 的灵敏度Δθ而言，我们可以将其定义为替换数据集中的某个元素后的最大扰动 $max_{S\sim S'}\left|\theta(S)-\theta(S')\right|$。根据定义，μ = Δθ / σ，因此如果我们希望 μ 较大，则添加的噪声 σ 较小；如果我们希望 μ 较小，则添加的噪声 σ 较大。

三、复合性与中心极限定理

大致说来，如果我们对同一个数据集进行多次查询，随着查询次数的增加，隐私性就逐渐下降了（一个拥有 n

条记录的数据库，在进行 $n \times \log_n 2$ 查询之后，就可以被重建出来）。那么问题来了，复合操作降低隐私性的速度有多快呢?

假设我们有两种隐私保护算法 M_1、M_2，且 M_1 为 M_2 的输入之一，其复合算法 M 为：

$$M : X \to Y_1 \times Y_2$$

在给定算法序列 $M_i : X \times Y_1 \times \ldots \times Y_{i-1} \to Y_i$ for $i \leqslant k$ 的情况下，我们可以将其复合函数递归定义为:

$$M : X \to Y_1 \times \ldots \times Y_k$$

为了定义 f–DP 的复合，我们首先将两个 trade–off 函数 f 与 g 的张量积⊗ 定义如下:

$$f \otimes g := T(P \times P', Q \times Q')$$

若真实值为 Anne 则第一个算法的概率分布为 P，第二个算法的概率分布为 P'； 若真实值为 Eva 则第一个算法的概率分布为 Q，第二个算法的概率分布为 Q'。

对于简单的 GDP 版本来说，k 个 GDP 的复合记为:

$$G_{\mu_1} \otimes G_{\mu_2} \otimes \cdots \otimes G_{\mu_k} = G_{\mu}$$

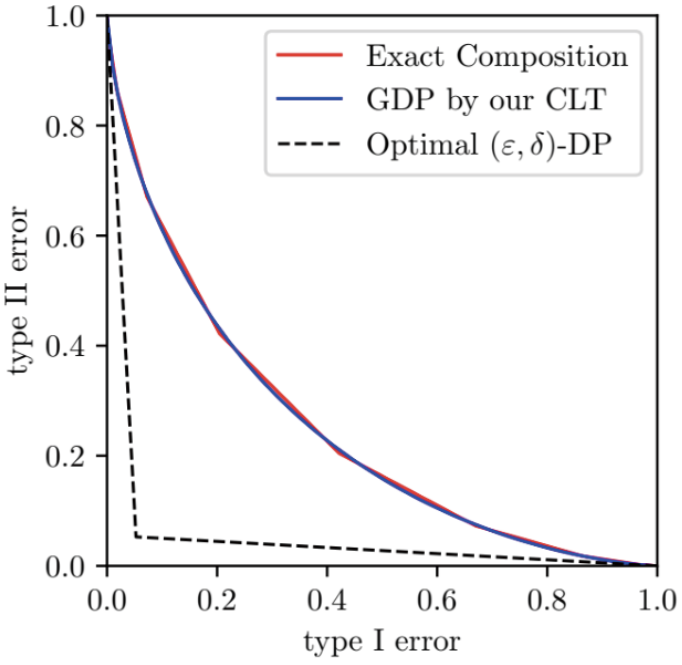
其中 μ 为参与复合的所有 GDP 算法 trade–off 函数中正态分布均值的二范数。

定理： 假设 $M_i(\cdot, y_1, \ldots, y_{i-1})$ 为第 i 个满足 f–DP 的算法（记为 f_i–DP），则其复合算法 $M : X \to Y_1 \times \ldots \times Y_k$ 是 trade–off 函数为 $f_1 \otimes \cdots \otimes f_k$ 的差分隐私算法，记为 $f_1 \otimes \cdots \otimes f_k - DP$。

f–DP 的中心极限定理是本次演讲中最重要的定理。令 $\{f_{ki} : 1 \leqslant i \leqslant k, k=1, 2, \ldots\}$ 为 trade–off 函数的一个三角形数组，其中每一个算法都接近完美隐私保护，当 k 趋近于正无穷时，f_{k1} 到 f_{kk} 的复合最终收敛到高斯分布上，它在 [0,1] 上一致收敛，而 μ 可以根据 f_{k1} 计算出来。

因此，若 M_{ki} 满足 f_{ki}–DP，则它们的复合近似满足 μ –GDP（通用性）。由于中心极限定理的存在，对一般的分布恰当地取平均最后都会渐进趋向于正态分布。此外，计算 trade–off 函数 $f_{k1} \otimes f_{k2} \otimes \cdots \otimes f_{kk}$ 是一个非常复杂的 #P 完全问题。我们可以通过 Edgeworth 展开进一步改进这个定理。

一般来说，为了应用中心极限定理，k 应该取一个较大的值。然而，在本例中，k 即使取 10 就已经足够了。



10-fold composition of $(1/\sqrt{10}, 0)$-DP. $\delta = 0.001$ in green curve

图 9：使用中心极限定理求得 GDP 的效果

如图 9 所示，红色的曲线代表精确的合成，由于 k 较小 (k=10)，所以我们可以将其计算出来；蓝色的曲线代表由我们的中心极限定理得到的 GDP，我们几乎无法区分红色和蓝色的两条曲线。而如果我们使用（ε，δ）–DP，它就不可能近似复合曲线，这是因为（ε，δ）–DP 包含了四段分段函数，我们无法用四段函数均匀地近似光滑的复合曲线。

在不考虑复合的情况下，我们重新考虑中心极限定理。在查询应答的场景下，如果查询的维度较高，无论我们将其中加入怎样的噪声符合怎样的分布，这个查询应答过程都近似满足 GDP。该定理还说明了差分隐私的不确定性定理。简单地说，「噪声变量」的值与「隐私成本」的平方的乘积要始终大于等于统计量 θ 的维度，因此我们无法同时令「噪声变量」与「隐私成本」都取较小的值，正所谓「鱼和熊掌不可兼得」。我们可以基于 Sudakov 定理对此进行证明。

四、通过子采样放大隐私

下面，我们将讨论最后一个性质：Subsampling。假设我们有一个非常大的数据集 S，我们通过子采样得到其中 10% 的数据，我们将算法 M 应用于采样得到的数据点上，记为：M。sub(S)。直观地说，这种做法为我们提供了更大的隐私性，因为 90% 的数据并不会暴露给算法。

假设我们从总数据量为 n 的数据集中均匀地采样出 m 个数据点的子集，令 p ：= m/n。给定任意 trade–off 函数 f 时，算子 C_p (f) 定义如下：

$$C_p(f) := Conv\left(\min\left\{f_p, f_p^{-1}\right\}\right) = \min\left\{f_p, f_p^{-1}\right\}^{**}$$

对于凸组合 f_p=pf+(1−p)Id 而言，由于 p 是由我们采样得到的样本计算而来，(1−p) 取决于我们没有采样的样本，因此它满足完美隐私保护。为了令该操作对称，我们考虑了 f_p 的反函数，为了令其为凸组合，我们使用了 $\min\{f_p, f_p^{-1}\}$ 的二次共轭形式。因此，子采样机制可以保护隐私，而其隐私程度取决于算子 C_p (f)。相比之下，Renyi DP 是一种非常复杂的自采样定理。

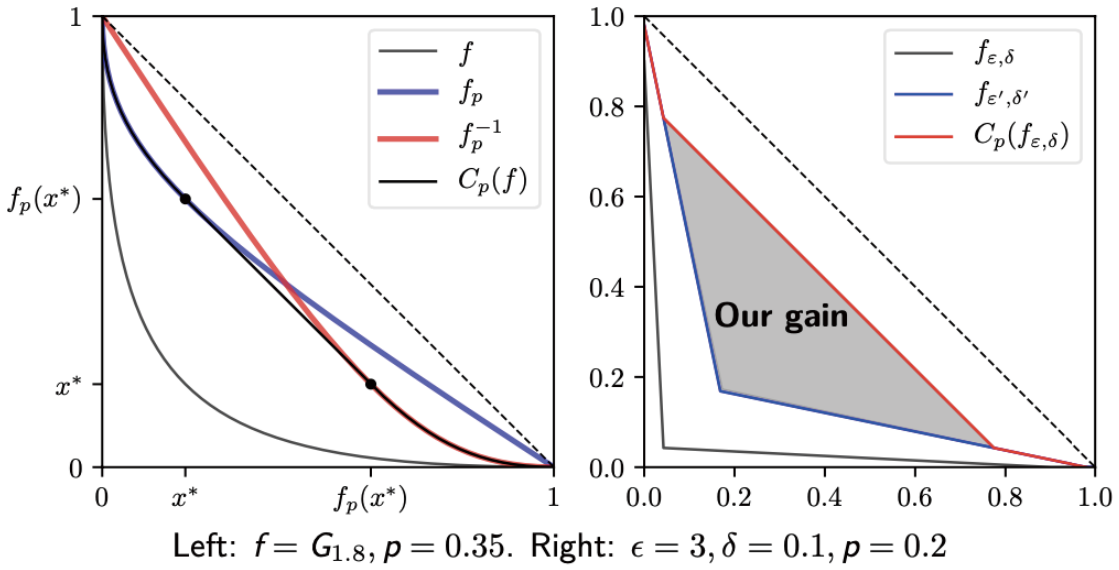


图 10：增益效果示意图

图 10 显示了子采样操作的增益效果。在右图中，黑色实线代表初始的隐私程度，其隐私性较低（十分接近原点）。当我们通过子采样得到数据集中 20% 的样本点 (p=0.2) 时，根据之前的子采样定理得到的隐私程度如蓝色实线所示，我们提出的新子采样定理得到的隐私程度则如红色实线所示。因此，如图中灰色的部分所示，我们对于隐私性的增益是非常大的。

五、f-DP 在深度学习中的应用

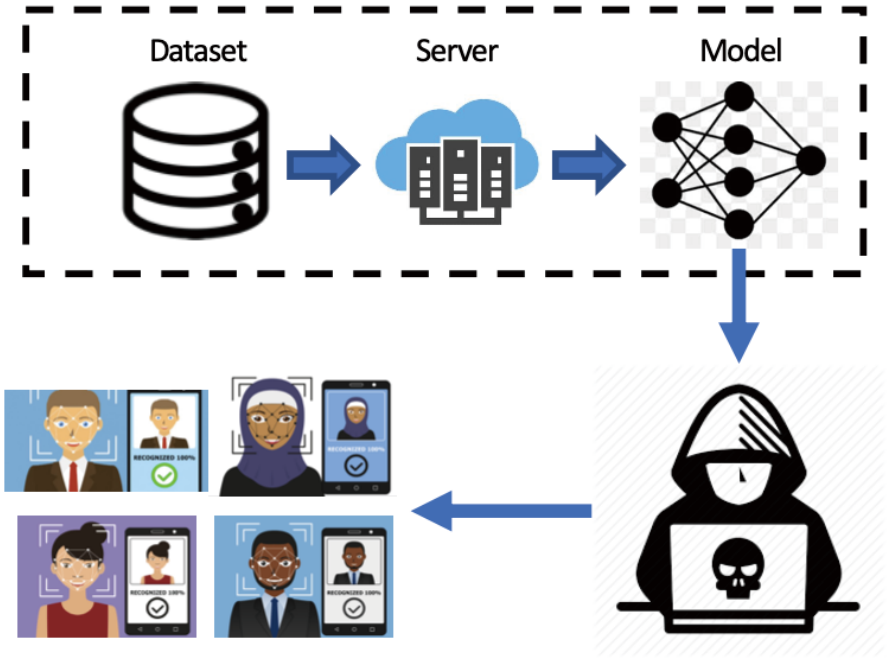


图 11：深度学习中的隐私保护问题

如今，深度学习技术无处不在。相应的，其隐私保护问题也凸显了出来。而由于深度学习架构的非凸性和许多复杂的天然特性，所以在之前很难将隐私保护引入到深度学习中。大约 3 年前，Google Brain 团队使用了一种名为「Moments accountant」的技术在 (ε , δ)-DP 的框架下来分析深度学习模型的隐私性。

Input: Dataset $S = \{x_1, \ldots, x_n\}$, loss function $\ell(\theta, x)$.
Parameters: initial weights θ_0, learning rate η_t,
subsampling probability p, number of
iterations T, noise scale σ, gradient norm bound R.

for $t = 0, \ldots, T-1$ **do**
Take a Poisson subsample $I_t \subseteq \{1, \ldots, n\}$ with subsampling probability p
for $i \in I_t$ **do**
$v_t^{(i)} \leftarrow \nabla_\theta \ell(\theta_t, x_i)$
$\bar{v}_t^{(i)} \leftarrow v_t^{(i)} / \max\left\{1, \|v_t^{(i)}\|_2 / R\right\}$ ▷ **Clip gradient**
$\theta_{t+1} \leftarrow \theta_t - \eta_t \cdot \frac{1}{|I_t|}\left(\sum_{i \in I_t} \bar{v}_t^{(i)} + \sigma R \cdot \mathcal{N}(0, I)\right)$ ▷ **Gaussian mechanism**

Output θ_T

图 12：具有隐私保护的深度学习

在随机梯度下降（SGD）的基础之上，我们通过「梯度截断」(clip gradient) 和「高斯机制」(Gaussian mechanism) 来保证深度学习训练过程的隐私性。「梯度截断」指的是当梯度过大时，我们通过缩放梯度使其变小。「高斯机制」指的是我们向平均梯度中加入高斯噪声。

我们可以通过 f-DP 改进深度学习的隐私性分析。SGD 实际上也是一种子采样策略，而深度学习本质上就是子采样与复合操作的结合。在训练过程中，我们子采样一个 mini-batch，然后在一轮轮的迭代中进行复合。这正是我们在所有 f-DP 框架中要重点考虑子采样和复合操作的原因。

根据本文之前介绍的各种特性，我们得到以下定理：当 $\mu = \frac{m}{n}\sqrt{T(e^{\frac{1}{\sigma^2}} - 1)}$ 时，具有隐私保护的深度学习 $M(S) = (\theta_1, \theta_2, \cdots, \theta_T)$ 渐进地满足 μ-GDP。其中 m 为 mini-batch 的大小，n 是样本的总数，T 为迭代轮次，而 σ 为我们向梯度中加入的噪声。

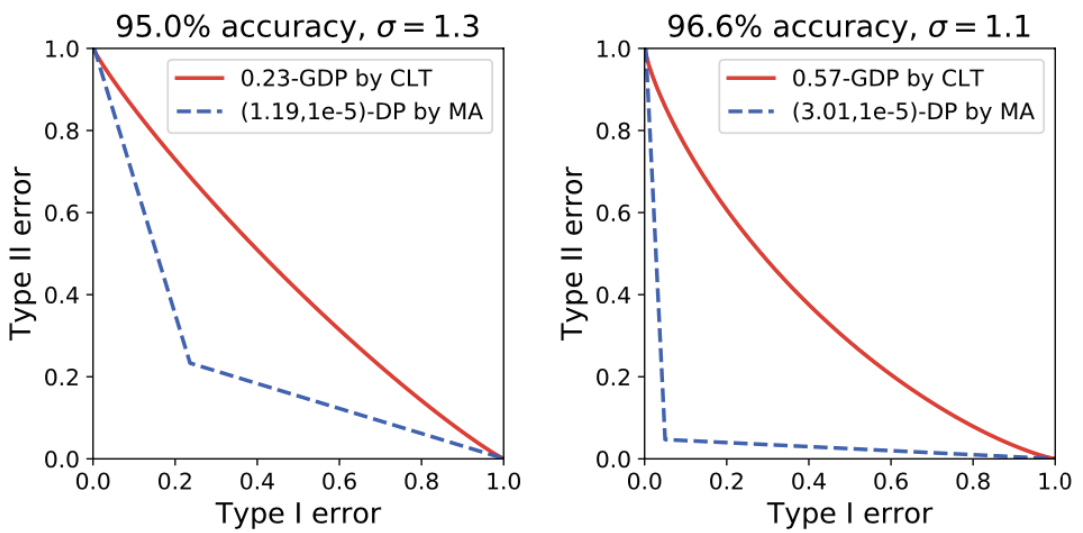
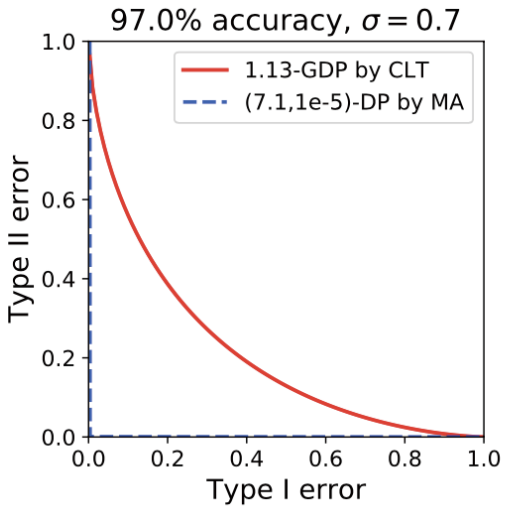


图 13：与 Google Brain 方法的隐私分析对比

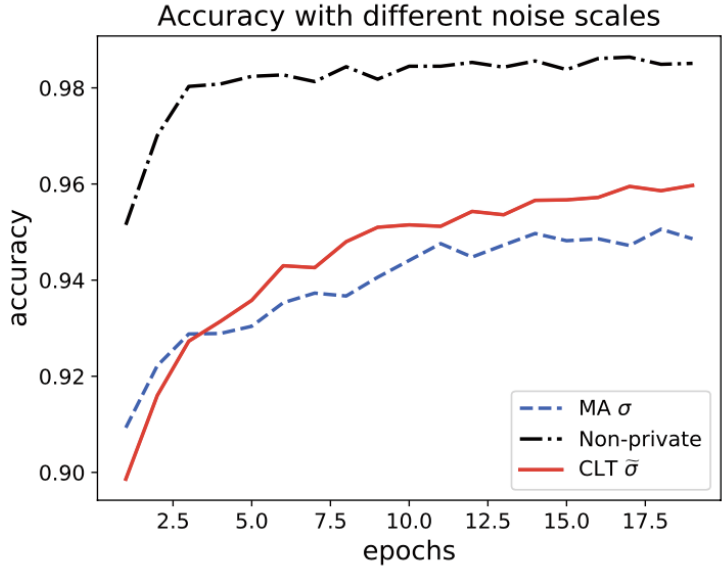
蓝色虚线代表 Google Brain 通过「Moments accountant」得到的隐私性分析结果，红色实线代表我们使用中心极限定理得到的 GDP 分析结果。我们的红色实线始终位于蓝色虚线的上，由于 trade-off 函数值越大，则隐私性越高，因此我们的方法更加能够保证隐私性。



Our f-DP interpretation is $\mathcal{N}(0,1)$ vs $\mathcal{N}(1.13,1)$; while MA gives $(7.1, 10^{-5})$-DP, noting $e^{7.1} = 1212.0$

图 14：与 Google Brain 方法的隐私分析对比

图 15 显示了最为显著的性能对比结果，其中 Google Brain 的方法分析结果认为其毫无隐私性可言（第一类错误和第二类错误都为 0），而我们的方法却认为这里仍然存在一些隐私性。此时，隐私性由 μ =1.13 决定，这相对来说还是具有一定隐私性的，因为我们不能很轻易地区分这两个分布。而在「Moments accountant」的设定下，ε 为 7.1，其似然比甚至超过了 1,200，人们会认为此时毫无性可言。



CLT uses $\tilde{\sigma} = 1.06$ and MA uses $\sigma = 1.3$, both giving $(1.34, 10^{-5})$-DP

图 15：我们基于中心极限定理的方法可以加入更小的噪声

当我们拥有了隐私性后，我们就可以利用这一性质，向梯度中加入更小的噪声。红色实线代表使用我们的新框架时的测试准确率，而蓝色虚线代表使用 Google Brain 方法时的测试准确率。需要强调的是，我们并没有损失任何性能，因为我们保留了相同的算法，我们仅仅减小了向梯度中加入的噪声的大小，同时也提升了测试时的准确率。

六、结语

未来，我们还有很多有待探索的研究方向:

- 首先，对于某些具体的算法来说，我们希望研究使用 f–DP 可以获得多大的隐私性增益。
- 此外，f–DP 是否能被应用于联邦学习呢？这是一个非常有前景的研究方向。
- f–DP 与神经网络的架构有何联系呢？也许我们可以考虑更多网络架构的信息，从而在 f–DP 框架下保证算法的隐私性。

首先，trade–off 函数可以为隐私损失提供有效的信息。f–DP 可以提供紧密（tight）的复合。f–DP 可以通过平均和凸化方法进行更加激进的子采样。以上三种性质使 f–DP 成为了目前性能最佳的深度学习隐私保护方法。

根据中心极限定理，无论是对于我们提出的 f–DP 框架，还是（ε，δ）–DP 框架，只要参与复合的元素过多，所有的问题都会被归结为高斯差分隐私（GDP）问题，它是我们所有研究的中心。这也许就是数学王子的王者归来吧！

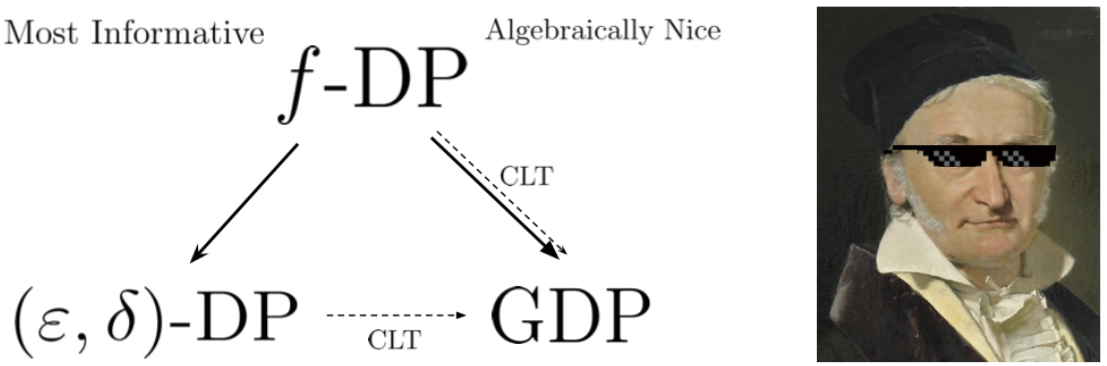


图 16：数学王子王者归来

斯坦福大学雷理骅：反事实与个体处理效应的共形推理

整理：智源社区 熊宇轩

因果推理是下一轮人工智能革命的重中之重。在本届智源大会“机器学习前沿青年科学家”论坛上，来自斯坦福大学统计系的雷理骅博士为我们带来了题为「反事实与个体处理效应的共形推理」的主题报告，深入浅出地介绍了因果推理的必要性，以及如何通过共形推理研究个体处理效应。

以下为演讲整理。

雷理骅：今天，我很高兴介绍自己近期关于因果推理的一些工作。具体而言，我将谈谈「反事实与个体处理效应的共形推理」。

目前，人工智能领域正面临着诸多巨大的挑战。事实上，Michale Jordan 两年前曾撰写了一篇非常重要且辛辣的文章「人工智能革命尚未发生」，讨论有关人工智能面临的挑战与人工智能的未来。他列举出了许多的挑战，并重点强调了「因果关系」和「量化不确定性」。如他所述，以上两者都是仿人人工智能领域的经典目标，但却常被当下的人工智能革命所忽略，而它们至今却还没有被很好地解决。

在本次演讲中，我将在这个方面展开一些讨论。具体而言，我将介绍如何围绕机器学习算法，将它们应用到因果推理中，从而试图实现可靠的不确定性量化。

一、个体处理效应

「个体处理效应」是本次演讲的主题之一。Seth Morgan 有一句名言：「传统研究的前提是，将治疗放在考虑的中心，并决定这种治疗是否对一名“典型”的病例有效？问题是，有很多患者并不是『典型』病例，我也像大多数人一样，并不是『典型』病例」。

这句名言出自两年前美国医学研究院举办的一场重要的研讨会。这场研讨会的重要主题之一正是研究异质处理效应。在介绍完该问题的重要性之后，我们需要开始解决这一问题。在本次演讲中，我将重点关注潜在结果 (Potential outcome)，尽管我们的研究也可以泛化到 Judea Pearl 在本届智源大会上介绍过的因果图上面。

下面，对于那些不太熟悉潜在结果的人，我们将给出一个简单的入门示例：

图 1：平行宇宙的两只荷兰猪

假如我们有两个平行宇宙，在每个平行宇宙中都有一只荷兰猪，这两只荷兰猪一模一样。在其中一个平行宇宙中，我们对荷兰猪施以治疗措施，而在另一个平行宇宙中则并不进行治疗。通过观测实验结果，我们发现在其中一个平行宇宙中，荷兰猪存活了下来，而在另一个平行宇宙中，荷兰猪则死掉了。在这里，以上两种情况都是潜在结果，但是在现实世界中我们只能观测到其中一种结果，这取决于我们是否对荷兰猪施以治疗或控制。

因此，在这里，潜在结果指的就是如果某人接受了治疗，他会有什么反应。

Inference of individual treatment effects?

- Potential outcome (PO) framework (Neyman, '23; Rubin, '74)
  - $T \in \{0, 1\}$ binary treatment
  - $Y(1), Y(0)$ potential outcomes
  - X covariates
- Individual treatment effects (ITE): $Y(1) - Y(0)$ (random)

Find interval estimate $\hat{C}_{\text{ITE}}(X)$ s.t. $\mathbb{P}(Y(1) - Y(0) \in \hat{C}_{\text{ITE}}(X)) \geq 90\%$

- For subjects **in** the study, only one PO missing $\Longrightarrow$ **counterfactual inference**
- For subjects **not in** the study, both POs missing $\Longrightarrow$ pure **ITE inference**

图 2：对个体处理效应进行推理

在这个框架之下，我们约定 T 为 0 或 1 中的某一个值，这个二值变量代表是否进行治疗；Y(1) 和 Y(0) 则代表潜在结果；X 是协变量。我们将个体处理效应 (ITE) 定义为：Y(1)–Y(0)。请注意，原则上，由于 Y(1) 和 Y(0) 可以是随机变量，因此二者之差也是随机变量，而不是一个目标策略（确定性目标，Deterministic target）。

因此在这个任务中，我们的目标是找到以下个体处理效应为真的置信度高于 90% 的区间估计 $\hat{C}$：

$$\hat{C}_{ITE}(X)\ s.t.\ \mathbb{P}\big(Y(1)-Y(0)\in \hat{C}_{ITE}(X)\big)\geq 90\%$$

为了实现这一目标，我们至少需要考虑两种场景：

(1) 当研究中有实验主体 (subject) 存在，只缺失了其中一种潜在结果。例如，如果我们在治疗组中有一个主体，根据定义，我们可以观测到其 Y(1)，而缺失了 Y(0)。因此在这种情况下，只缺失了一个潜在结果，我们需要对其进行推理。这种情况可以被归结为「反事实推理」。
(2) 当研究中不存在实验主体 (subject) 时，两种潜在结果都缺失了。这种情况可以被归纳为「纯 ITE 推理」，这更加困难。

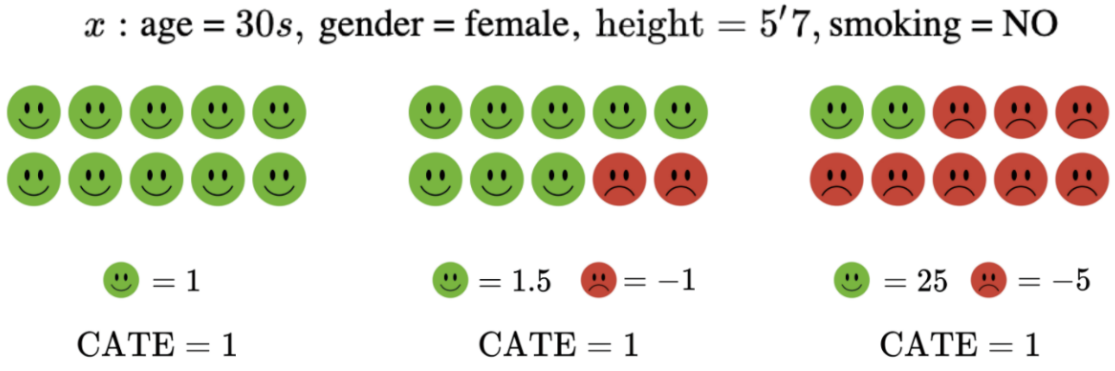


图 3：与条件平均因果效应的对比。

处理这种差异化处理效应问题的经典方法是： 估计条件平均因果效应 (Conditional average treatment effects, CATE)。即给定协变量值的条件下的 ITE 的期望：

$$\tau(x) \triangleq \mathbb{E}\left[\,Y(1)-Y(0)\,|\,X=x\,\right]$$

根据定义，这里还存在着很多问题。假设我们有三个实验组，它们的协变量值都相同，其 CATE 值也相同。但是在第一个实验组中，所有人都具有正向的 ITE； 在第二个实验组中，80% 的人具有正向的效应； 在第三个实验组中，只有 20% 的人拥有正向 ITE，但是他们的正向效应非常大。

我们当然会推荐第一个实验组的情况，因为每个人的情况都变得更好了； 但是如果我们现在处于第三个实验组中，我们就要考虑是否要接受这种处理方案了。

前面的例子说明了，在实际中进行 ITE 推理比进行 CATE 推理要更有道理。那么，如上图所示，我们接下来的工作则是以下面三个因果推理领域最标准的假设为基础的:

- 超总体假设 (super–population assumption):

$$(X_i, T_i, Y_i(1), Y_i(0)) \overset{i.i.d.}{\sim} (X, T, Y(1), Y(0))$$

即所有的样本相互独立，且于总体同分布。

- 个体处理稳定性假设 (SUTVA):

$$Y^{\text{obs}} = \begin{cases} Y(1) & \text{if } T = 1 \\ Y(0) & \text{if } T = 0 \end{cases}$$

即如果主体得到了处理，观测到的结果就是 Y(1)，否则观测到的结果就是 Y(0)。

- 强可忽略性假设:

$$(Y(1), Y(0)) \perp\!\!\!\perp T \mid X$$

即潜在结果与给定协变量时的处理方案分配相独立。

二、反事实推理

我们的第一个任务是进行反事实推理。

The counterfactual inference problem

- For a testing control unit X, $T = 0$, $Y^{\text{obs}} = Y(0)$, $Y(1) = ?$

$$\mathbb{P}(Y(1) \in \hat{C}_1(X) \mid T = 0) \geq 90\% \Longrightarrow \mathbb{P}(Y(1) - Y(0) \in \hat{C}_1(X) - Y(0) \mid T = 0) \geq 90\%$$

- For a testing treated unit X, $T = 1$, $Y^{\text{obs}} = Y(1)$, $Y(0) = ?$,

$$\mathbb{P}(Y(0) \in \hat{C}_0(X) \mid T = 1) \geq 90\% \Longrightarrow \mathbb{P}(Y(1) - Y(0) \in Y(1) - \hat{C}_0(X) \mid T = 1) \geq 90\%$$

$$\mathbb{P}(Y(t) \in \hat{C}_t(X) \mid T = 1 - t) \geq 90\%$$

$$\Longrightarrow \quad \hat{C}_{\text{ITE}}(X; T, Y^{\text{obs}}) = \begin{cases} \hat{C}_1(X) - Y(0) & \text{if } T = 0 \\ Y(1) - \hat{C}_0(X) & \text{if } T = 1 \end{cases} \quad \text{valid}$$

图 4：反事实推理问题定义

假设我们有一个控制组 X，如前所述，我们观测到其输出结果为 Y(0)，那么当对其施加处理时，Y(1) 将是多少呢?

在这种情况下，假设我们可以得出$\hat{C}_1(\mathrm{x})$，在 T =0 的条件下，Y(1) 有 90% 的置信度落在其中。接着，我们可以通过用$\hat{C}_1(\mathrm{x})$减去 Y(1) 构造一个新的区间。此时 ITE 有至少 90% 的概率属于这个新构造的区间。反之，我们也可以对干预组做相同的操作。

从某种程度上来说，如果我们根据观察的是控制组还是干预组将$\hat{C}_{\mathrm{ITE}}$定义为

$$\hat{C}_{ITE}(X;T,Y^{obs}) = \begin{cases} \hat{C}_1(X) - Y(0) & if\ T=0 \\ Y(1) - \hat{C}_0(X) & if\ T=1 \end{cases}$$

我们就可以得到一个有效区间（Valid Interval）

此时，我们可以将任务归纳如下：

$$\mathbb{P}(Y(1) \in \hat{C}_1(X)) \geq 90\% \quad \text{where} \quad (X, Y(1)) \sim P_{X|T=0} \times P_{Y(1)|X,T=0}$$

我们希望找到一个$\hat{C}_1(\mathrm{X})$使得概率声明$\mathbb{P}\left(\mathrm{Y}(1) \in \hat{C}_1(\mathrm{X}) | \mathrm{T}=0\right) \geq 90\%$成立。它等价于，我们认为该概率声明在 (X,Y(1)) 服从给定 T=0 时 (X,Y(1)) 的联合概率分布的情况下成立。

$$\mathbb{P}(Y(1) \in \hat{C}_1(X)) \geq 90\% \quad \text{where} \quad (X, Y(1)) \sim P_{X|T=0} \times P_{Y(1)|X}$$

$$\text{Strong ignorability assumption} \Longrightarrow Y(1) \mid X, T=0 \overset{d}{=} Y(1) \mid X$$

对上面的联合分布进行简单的分解后，我们希望得到给定 T=0 时 X 的边缘概率；以及给定 X 的值且 T = 0 时的 Y(1) 的条件概率分布。

根据强可忽略性假设，我们可以直接将第二个因式中的 T 移除，

得到其简化形式：$\mathrm{Y}(1) | X, T=0 \overset{\text{def}}{=} Y(1)|X$。这就是我们的目标分布。

$$\mathbb{P}(Y(1) \in \hat{C}_1(X)) \geq 90\% \quad \text{where} \quad (X, Y(1)) \sim P_{X|T=0} \times P_{Y(1)|X}$$

$$\text{The observed treated units: } (X_i, Y_i^{\mathrm{obs}})_{T_i=1} \overset{i.i.d.}{\sim} P_{X|T=1} \times P_{Y(1)|X}$$

$$\text{We have } P_{X|T=1} \times P_{Y(1)|X}$$

$$\text{We want } P_{X|T=0} \times P_{Y(1)|X}$$

但在现实中，我们能观测到的是干预组，即从另一个给定 T=1 时 X 的条件分布以及给定 X 时 Y(1) 的条件分布中采样得到的独立同分布的样本$\left(\mathrm{X_i}, Y_i^{obs}\right)_{T_i=1}$。

总而言之，如上图所示，我们拥有的是红色的概率分布 $P_{X|T=1}$，而我们希望得到蓝色的概率分布 $P_{X|T=0}$。而这两个概率分布后的条件分布是相同的，其不同之处在于协变量分布。

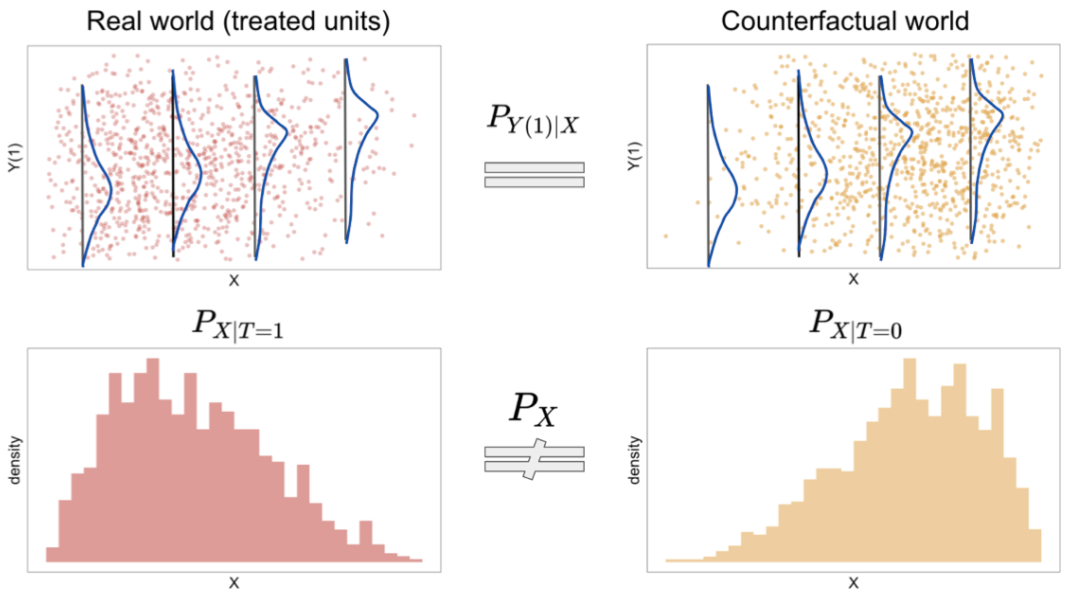


图 5：反事实推理与协变量偏移图解

图 5 是对上文所述的各种假设与我们的目标的图解。在图的左侧，我们展示了真实世界中的观测结果（干预组），右侧则是反事实世界中的观测结果。

在上面一行，我们给出了横坐标为 x、纵坐标为 Y(1) 的散点图。由强忽略性假设可知，左右两侧的条件分布是相同的。而真正使这两个散点图不同的原因则是它们的协变量分布。对于真实观测结果来说，X 的概率密度为粉红色的分布，而我们的目标则是右侧橙色的分布。

Use i.i.d. samples from $P_{X|T=1} \times P_{Y(1)|X}$ to construct $\hat{C}_1(X)$ with

$$\mathbb{P}(Y(1) \in \hat{C}_1(X)) \geq 90\% \quad \text{under } P_{X|T=0} \times P_{Y(1)|X}$$

Covariate shift $\quad w(x) \triangleq \dfrac{dP_{X|T=0}}{dP_{X|T=1}}(x) \propto \dfrac{1-e(x)}{e(x)}$

$$e(x) \triangleq \mathbb{P}(T=1 \mid X=x) \quad \text{propensity score}$$

图 6：问题重述

现在，我们的目标可以改写为：使用从 $P_{X|T=1} \times P_{Y(1)|X}$ 中采样得到的独立同分布的样本，构建 $\hat{C}_1(X)$，并且满足 Y(1) 落在其区间内的置信度大于 90%，它需要适应协变量变化后 $P_{X|T=0} \times P_{Y(1)|X}$ 的情况。

这是一类被称作「协变量偏移」的问题，机器学习领域的研究人员十多年前对此进行了深入研究。而事实上，统计学领域的科学家们数十年前在进行抽样调查时就已经涉足这一问题。在这种情况下，我们可以将协变量偏移写作两个概率分布之比：$w(x) \triangleq \frac{dP_{X|T=0}}{dP_{X|T=1}}(x)$，而根据简单的贝叶斯公式，我们可以推导出该比值正比于 $\frac{1-e(x)}{e(x)}$，其中 e(x) 为倾向指数（propensity score）$e(x) \triangleq \mathbb{P}(T=1 \mid X=x)$，即给定协变量时某实验对象得到处理的概率。可以证明，这里的倾向指数是因果推理中最基本、最重要的对象之一。

以上分析说明，我们面对的是一种存在协变量偏移的预测性推理问题。幸运的是，前人已经对此有所研究。

有一类工作被称为「共形推理」，这种技术十分神奇。下面，我将对论文「Weighted split conformalized Quantile Regression」中的处理过程进行说明，并展示这一过程的效果。

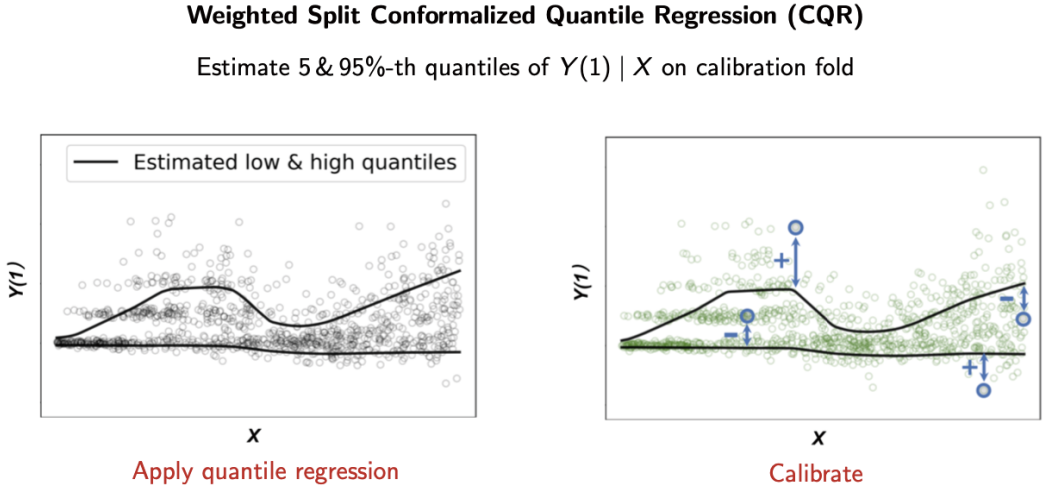


图 7：协变量偏移情况下的共形推理（conformal inference）；右图是校正后的概率分布。

第一步，我们将随机地将 $(X_i, Y_i^obs)_{(T_i=1)}$ 划分为两堆数据。

在第一堆数据中，我们可以使用任意方法去拟合 Y(1)|X 的 5% 和 95% 分位数，即我们可以使用分位数回归 (quantile regression)、分位数随机森林、分位数 Boosting、分位数神经网络等任意的方法进行拟合。

接着，我们将上面的估计结果应用于校正概率分布。之后，我们将计算每个点到这两个包络面的符号距离：

$$V_i \triangleq \max\left\{\hat{q}_{0.05}(X_i) - Y_i(1), Y_i(1) - q_{0.95}(X_i)\right\}$$

具体而言，我们首先计算每个点到每个包络面的距离，如果该点在包络面的外部，我们给这个距离赋予一个「+」号，而如果该点在包络面内部，我们给这个距离赋予一个「−」号。该直方图等价于符号距离的经验累积分布函数。

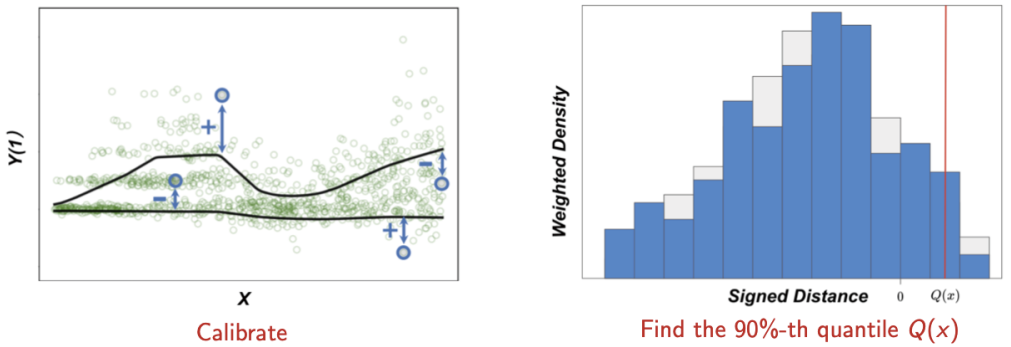


图 8：右图中为概率直方图，蓝色为重新加权后的概率直方图，红线为通过加权划分共形分位数回归方法找到的 90% 分位数。

当我们计算出符号距离后，我们可以绘制出其概率直方图，其中「0」两侧的距离符号相反。

$$\sum_{i=1}^{n} p_i(x)\delta_{V_i} + p_\infty(x)\delta_\infty \text{ where } p_i(x) = w(X_i) / \left(\sum_{i=1}^{n} w(X_i) + w(x)\right)$$

在这里，我们使用根据似然比 / 协变量偏移得来的权值 w(x)，对直方图重新加权。如上图所示，蓝色的部分是重新加权之后的直方图。

接着，我们找出加权直方图的 90% 分位数（红色的刻度）。

$$Q(x) \triangleq \texttt{Quantile}\left(90\%, \sum_{i=1}^{n} p_i(x)\delta_{V_i} + p_\infty(x)\delta_\infty\right)$$

最后，我们将输出置信区间：

$$\hat{C}_1(x) = [\hat{q}_{0.05}(x) - Q(x), \hat{q}_{0.95}(x) + Q(x)]$$

假设倾向指数已知，我们将权值 w(x) 设为 w(x)=(1−e(x))/e(x)，我们有：

$$90\% \le \mathbb{P}(\mathrm{Y}(1) \in \hat{C}_1(\mathrm{X}) | \mathrm{T} = 0) \le 90\% + \mathrm{c}\, n^{-1/2}$$

即在有限样本中，该置信度始终大于 90%，而不需要任何假设。也就是说，该结论对于任意的条件分布 $P_{Y(1)|X}$（例如，柯西分布、正态分布等）、任意的样本量大小都成立，它也适用于拟合任意的条件分位数。无论这种估计效果多差，我们始终都可以保证上述置信度约束成立。

另一方面，该约束在非常宽宽松的条件下也有一个上界 90%+$cn^{-1/2}$，条件为:

假设符号距离是连续随机变量且 $\mathrm{E}\left[\frac{1}{\mathrm{e}(\mathrm{X})^2}\right]<\infty$（即倾向指数不会取太极端的值）。该上界也对于任意的条件分布 $\mathrm{P}_{Y(1)|X}$（例如，柯西分布、正态分布等）、任意的样本量大小都成立，它也适用于拟合任意的条件分位数。

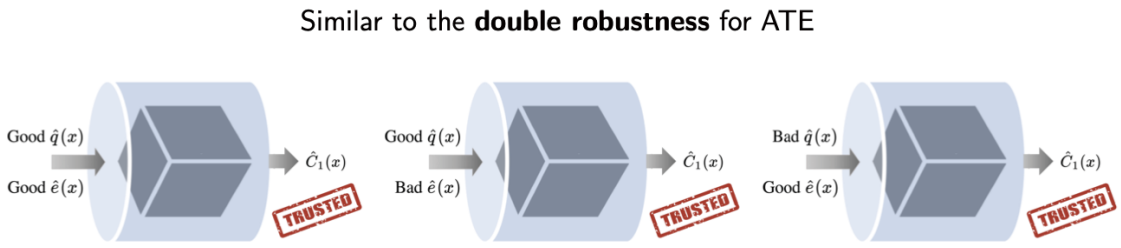


图 9：近似反事实推理

大体说来，这个神奇的过程如右下角的示意图所示： 一旦我们得到了一个倾向指数，我们可以对模型进行任意的估计，这就好比一个黑盒。当我们将倾向指数和模型估计输入这个封装器处理后，它会输出一个有限样本上的可信的区间估计。具体而言，这种方法适用于完全随机 / 分层的实验，具有良好的依从性。因为根据我们的设计，在本例中，倾向指数是已知的。

另一方面，当我们并不知道倾向指数时，观测性研究的结果将会如何呢？结果表明，在至少满足以下两种条件之一的情况下，我们仍然可以近似保证 90% 的置信度:

(1) $\hat{e}(x)\approx e(x)$，即倾向指数可以被很好地估计。

(2) $\hat{q}_{0.05/0.095}(x)\approx \mathrm{q}_{0.05/0.095}(x)$，即条件分位数可以被很好地估计。

当条件 (2) 可以被满足时，我们可以得到一个更强的结论:

$$\mathbb{P}\left(\mathrm{Y}(1)\in\hat{C}_1(\mathrm{X})\middle|\mathrm{T}=0,\mathrm{X}\right)\approx 90\%\quad\text{with high probability}$$

这与经典因果推理领域中的平均因果效应（ATE）的双重鲁棒性相类似。

综上所述，假设我们可以同时很好地估计倾向指数和条件分位数，那么我们就可以得到可信的区间估计 $\hat{C}_1(x)$ ； 但是如果我们无法很好地估计其中一个值，最终也能得到可信的区间估计，这就是所谓的「双重鲁棒性」。

三、纯 ITE 推理

接下来，我们将要讨论更为激进的纯 ITE 推理。

最朴素的方法是通过加权划分共形分位数回归（CQR）方法得出 $\hat{C}_1(x)$ 和 $\hat{C}_0(x)$，然后直接对这两个区间估计进行比较，从而得到 ITE 的区间估计。

$$\hat{C}_{ITE}(x) = C(x) - C(x)$$

但是，这里存在两个问题：（1）潜在结果被解耦了（2）为了保证有效性，我们需要对潜在结果进行 Bonferroni 校正。以上两个问题都会使得这一过程非常保守。

在这里，我将提出一种新的嵌套方法（Nested approach）。我们使用反事实推理作为一个中间步骤为实验主体生成 ITE 区间，然后我们试图将生成的区间泛化到实验中未考虑的主体上。这种方法可以显著降低朴素方法的保守性。

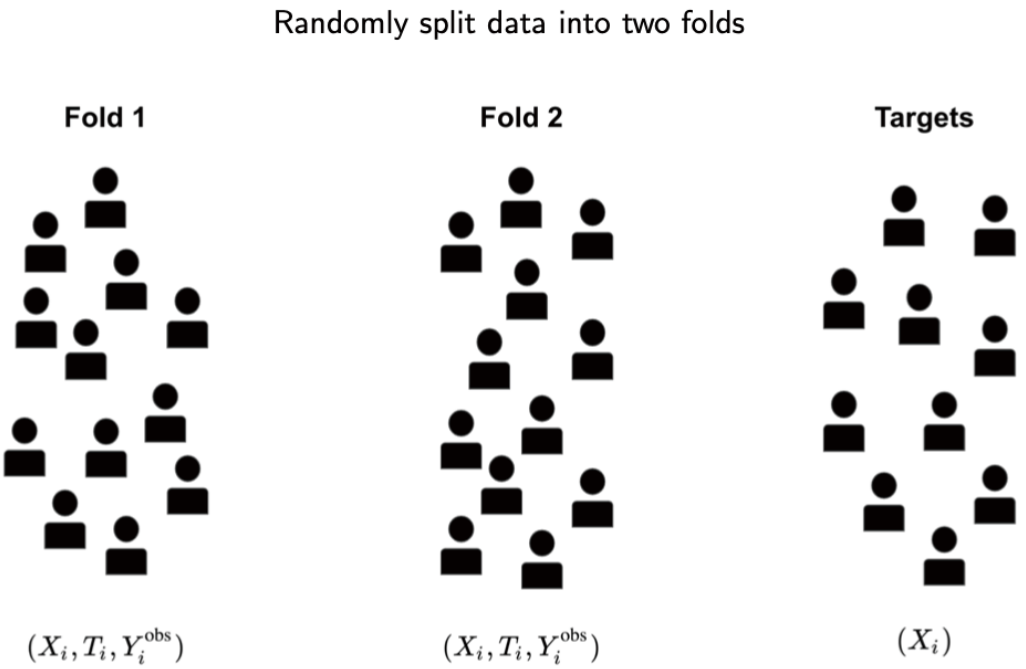


图 10：通过 Nested 方法进行 ITE 推理的图解

首先，我们也将随机地将数据分为两堆（Fold 1 和 Fold 2），我们将会把推理结果应用到图 10 中的 Targets 组中。在 Fold 1 和 Fold 2 中，我们可以看到协变量、处理方案分配、观测结果，而在 Target 种群中，我们只能看到协变量。

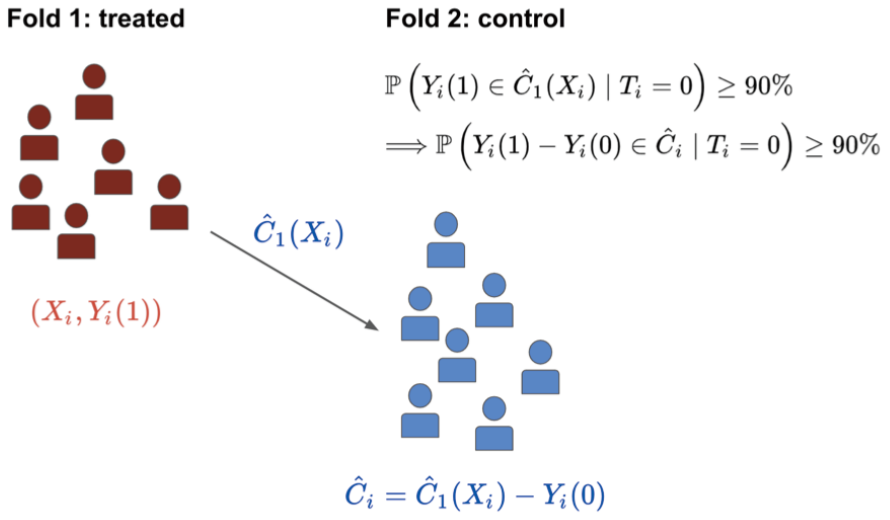


图 11：将反事实推理应用于 Fold 1 中的干预组，估计 Fold 2 中的置信区间

首先，我们取 Fold 1 中的干预组，并且对其进行反事实推理（例如，加权划分 CQR），从而得到 $\hat{C}_1(x)$。然后我们在 Fold 2 的控制组中估计 $\hat{C}_1(X_i)$。

由于我们可以观测到 Fold 2 中的 $Y_i(0)$，我们可以通过以下方式计算 $\hat{C}_i$：

$$\hat{C}_i(x) = \hat{C}_1(X_i) - Y_i(0)$$

由此，我们可以得到 $\hat{C}_i$ 有至少 90% 的置信度覆盖 ITE。

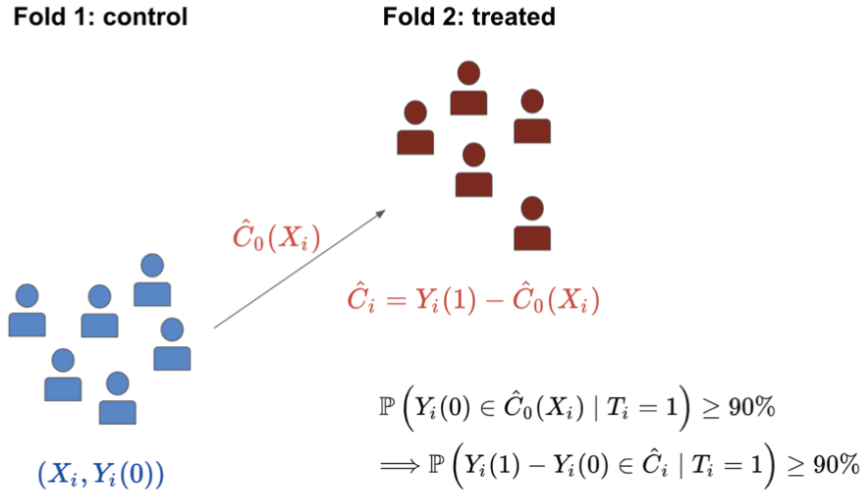


图 12：根据 Fold 1 中的控制组估计 Fold 2 中的干预组，并为 Fold 2 中的干预组构建 ITE 区间

类似地，我们也可以将这个过程反过来，我们可以根据 Fold 1 中的控制组估计 Fold 2 中的干预组。

同样地，我们可以构建满足条件

$$\mathbb{P}\left(Y_i(0) \in \hat{C}_0(X_i) \mid T_i = 1\right) \geq 90\%$$
$$\Longrightarrow \mathbb{P}\left(Y_i(1) - Y_i(0) \in \hat{C}_i \mid T_i = 1\right) \geq 90\%$$

的区间 $\hat{C}_i = Y_i(1) - C_0(X_i)$。

在进行上述计算后，我们最终得到了包含 i 个观测结果的集合 $(X_i, \hat{C}_i)$，其中 X_i 为协变量，而 $\hat{C}_i$ 为置信区间，而 $\hat{C}_i$ 有 90% 的置信度覆盖真实未知的 ITE。$\hat{C}_i$ 事实上也是 ITE 的不确定性度量。

接下来，我们拟合某种某型，使用 X_i 作为协变量，学习区间 $\hat{C}_i$ 的左右端点。

最后我们会将这里的学习器应用于 Targets 组，为每个测试点生成 ITE 区间。

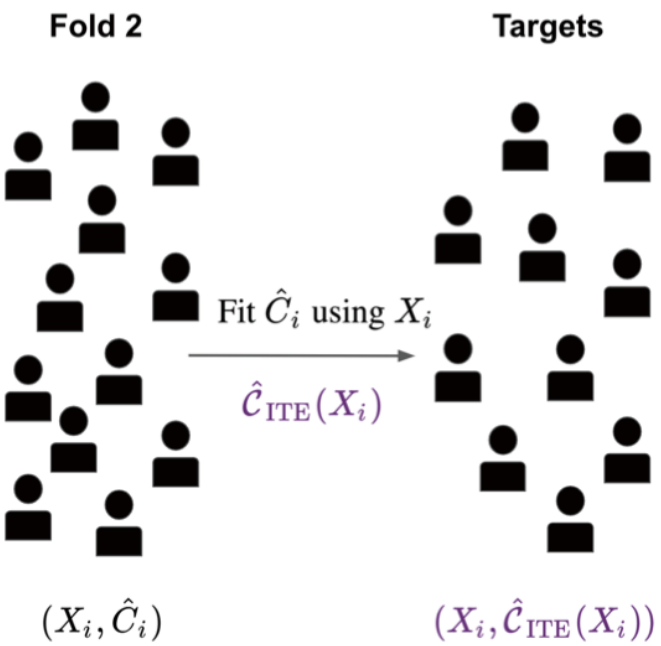


图 13：为每个测试点生成 ITE 区间，Fold 2 的区间可能在测试样本上仍有效

由于我们知道 Fold 2 上的区间有一个对 ITE 的最小置信度，当学习过程不太差时，Fold 2 的区间在 Target 组上仍然有效。我们可以通过 $(X_i, \hat{C}_{ITE}(X_i))$ 得到一个近似的最小置信度。

四、实证结果

接下来，我将展示一些实证结果。

首先，我将介绍一个仿真实验结果。该实验是 Wager 和 Athey 2018 年的一份工作的变体。

在这里，我将跳过这个仿真实验的细节，主要介绍定性的部分。协变量 X 属于一个多元高斯分布，协变量之间是相互独立或相关的，其维度为 10 或 100。我们将潜在结果的基线设置为 0，这意味着 ITE 推理就约简为了反事实推理。

接着，我们将对 Y(1)|X~N(μ (x), σ (X)^2) 进行仿真，其中均值 μ (x) 均匀地依赖于 X_1 与 X_2，方差可以是同方差的，也可以是异方差的。倾向指数 e(X) ∈ [0.25,0.5]，并均匀地依赖于 X_1。

我们尽可能地简化这个仿真实验，关键在于，在这种简单的模型下，所有的方法都应该有效。

我们开发了名为「cfcausal」的 R 语言程序包 (github.com/lihualei71/cfcausal) 来实现仿真实验。

在这个仿真实验中，我们将评估三种版本的 CQR 方法与三种对比基准方法。如前文所述，我们可以将任意的算法封装到这个 CQR 黑盒中。具体而言，我们封装了随机森林、Boosting、BART（贝叶斯可加回归树，一种非常流行的因果推理算法）三种算法。我们还考虑了三种基线算法：因果森林、X-learner、BART，这三种算法都非常流行，在过去经过了深入研究。这三种方法可以通过不同的方式来量化不确定性。

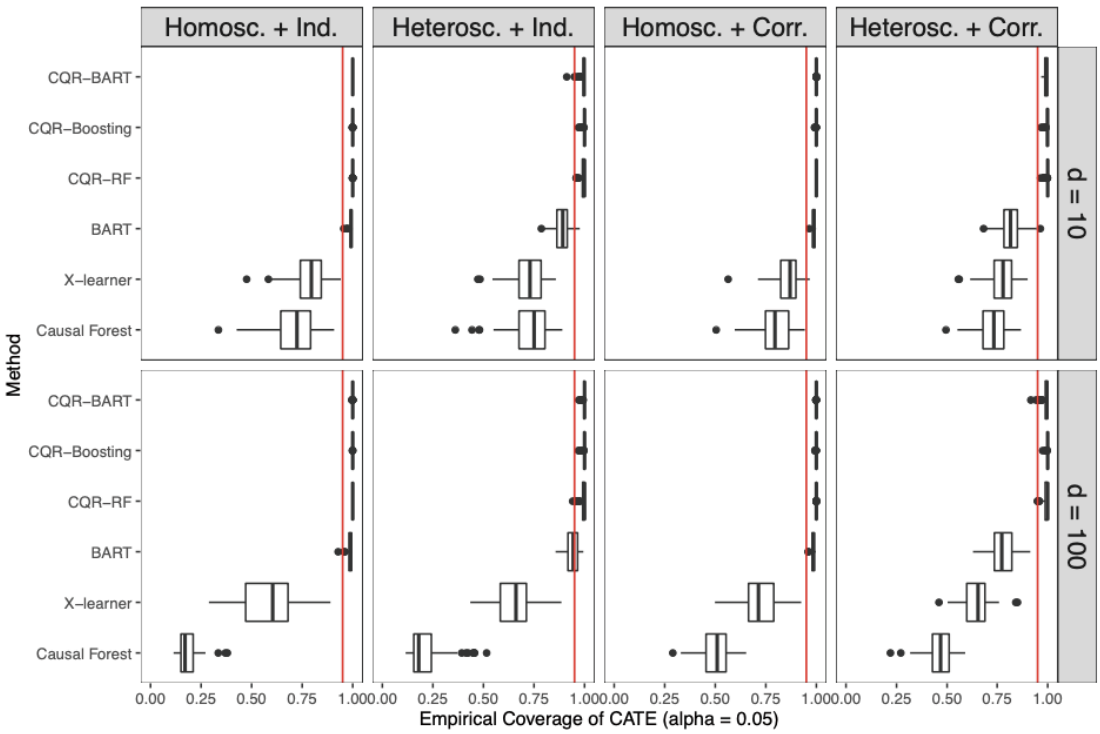


图 14：CATE 的边缘覆盖率（合理性检验）

图 14 显示了 CATE 的边缘覆盖率。我们的方法并不能保证覆盖 CATE，因为我们的方法的设计目标是覆盖 ITE。但我们可以将这种方法用于合理性检验，因为其余三种基线的设计目标都是覆盖 CATE。

即使在拥有 10 个协变量 (d=10) 时，我们可以看到这三种对比基线都没有达到最小的保险覆盖率 (在本例中目标覆盖率为 95%)，而在某些设定下，它们的覆盖率低至 75%，从统计意义上说这是一个非常低的覆盖率。当 d=100 时，某些算法 (因果森林) 的覆盖率甚至低至 25%。然而，我们的算法在所有的情况下都有非常保险的覆盖率 (基于高至 1)。

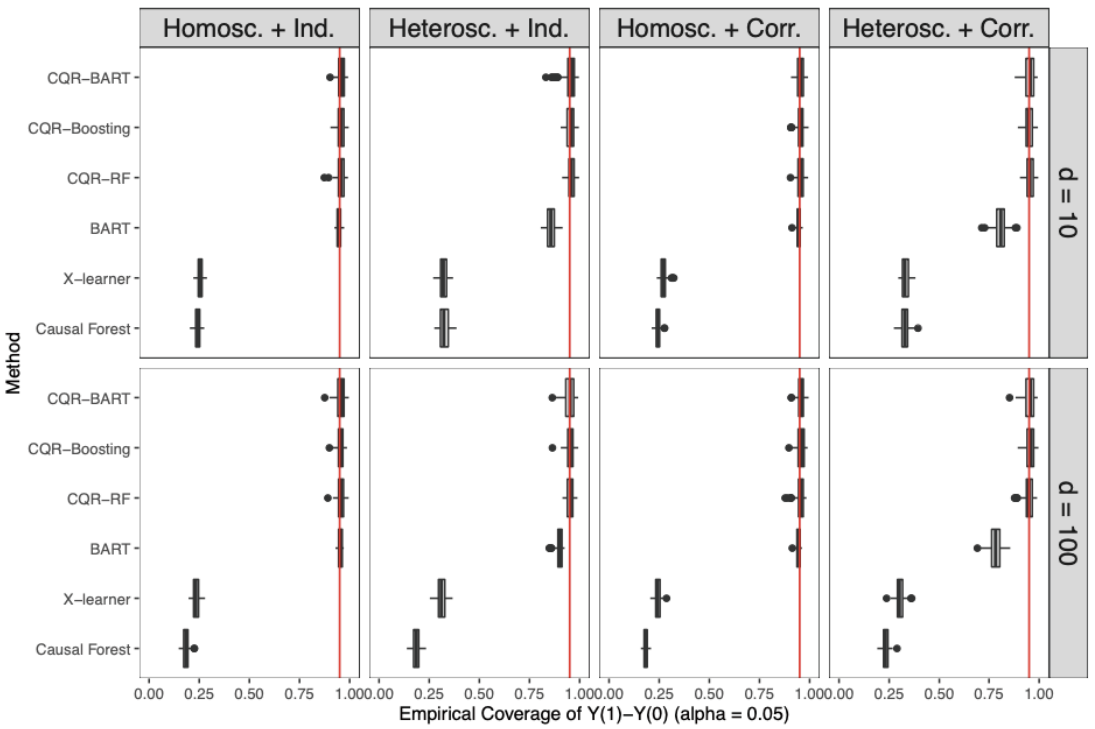


图 15：ITE 的边缘覆盖率

图 15 中，我们展示了 ITE 的边缘覆盖率。同样地，我们的方法旨在覆盖 ITE。令人十分惊讶的是，我们的方法的覆盖率恰好为 95%，我们的定理也指出了其覆盖率不仅高于 95%，而且误差不会太大。事实上，在所有的实验设定下都出现了这种情况。同时，我们也可以看到，其它的对比基线方法的覆盖率则往往很低，没有达到 95%。

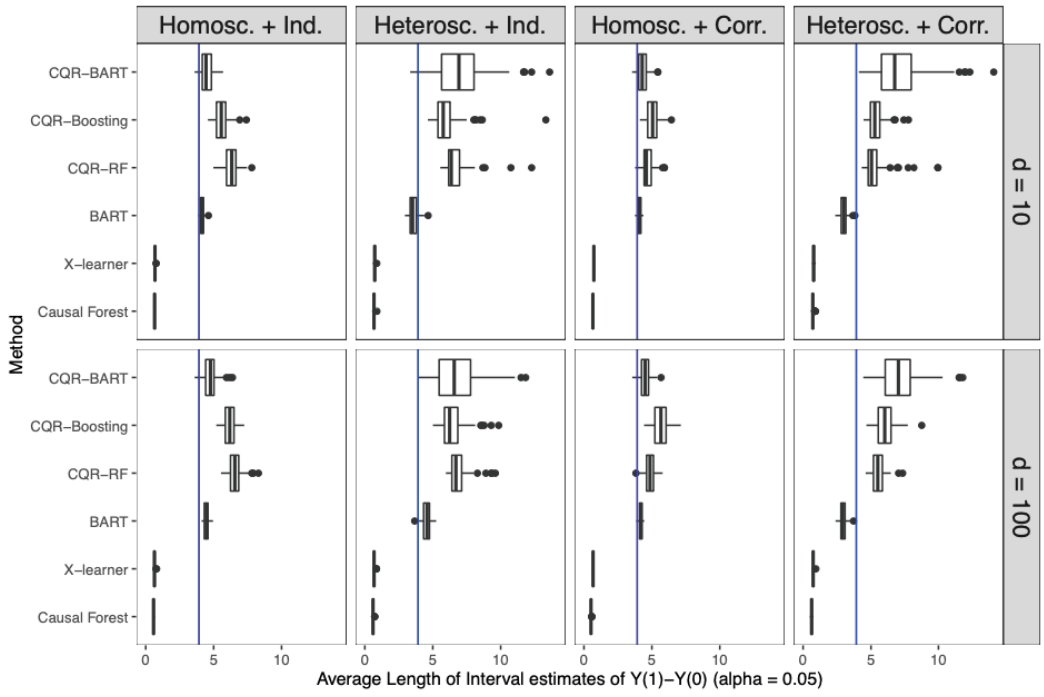


图 16：ITE 置信区间的平均长度

你可能会认为，为了保证覆盖率，你可能会使置信区间的长度非常宽。但是，在这里我们将所有的方法与标准方法（oracle）进行了对比。我们在样本量无限、模型十分理想的情况下求出了这种标准方法的置信区间的长度。但是，我们在这里其实只有 1,000 个样本。

如你所见，我们的方法的置信区间长度与标准方法十分相近。当然，其它方法的置信区间长度非常短，但这并不意味着它们是有效的（简而言之，他们的覆盖率非常低）。

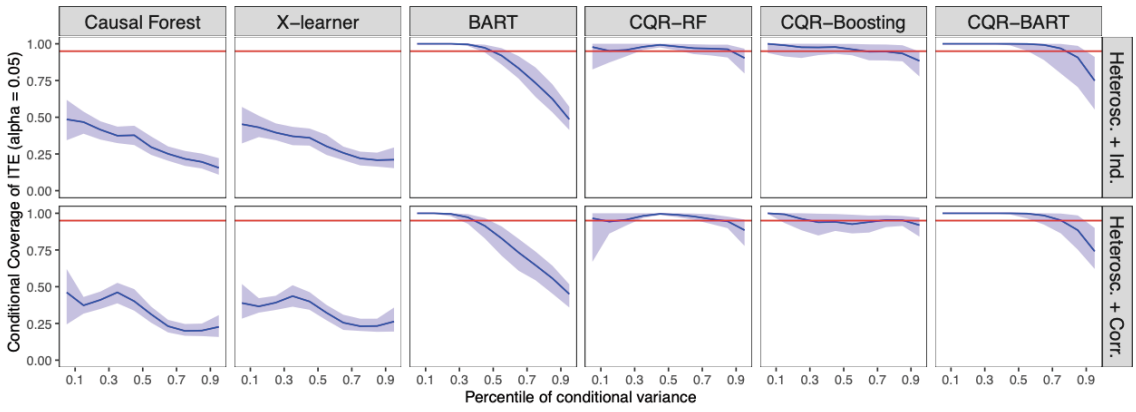


图 17：ITE 的条件覆盖率

由于我们的理论只保证在边缘覆盖率上成立，但是也很有必要看看条件覆盖率的表现。如上图所示，我们的方法有非常高的条件覆盖率，而其它的方法则不能做到这一点。

这些数据是基于 NLSM 获得，它于 2018 年最大的因果推理学术会议 (ACIC) 所使用，我们会基于这个数据集生成一些合成数据。

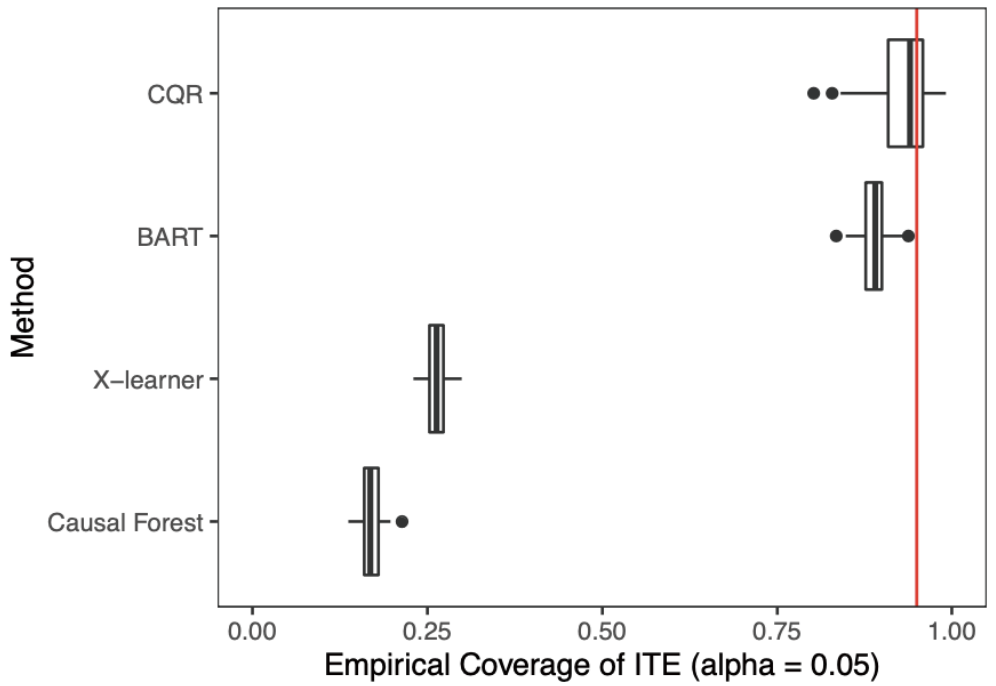


图 18：ITE 的边缘覆盖率与三种基线的对比

通过将我们的方法与其余三种基线对比，与前文所述的图表相类似，我们的方法保证了其它方法无法达到的覆盖率。

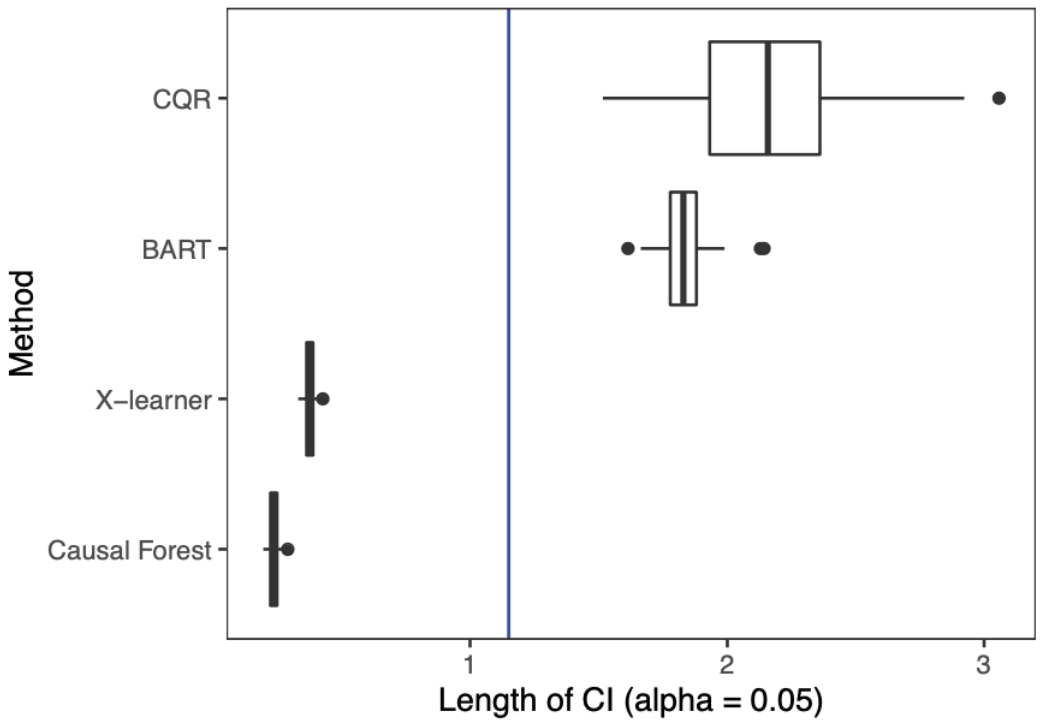


图 19：ITE 置信区间的平均长度

我们方法得出的置信区间的长度也很好。

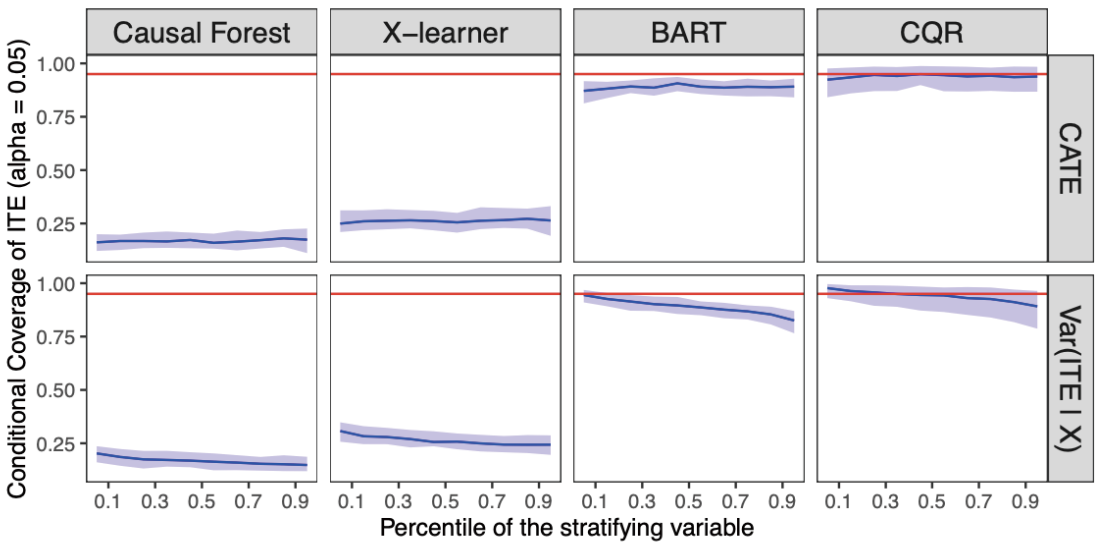


图 20：ITE 的条件覆盖率

最后，在本例中，我们的方法也取得了很好的条件覆盖率。请注意，在这里，我们需要同时对两种潜在结果进行推理，因此它是一个纯 ITE 推理问题，而不是一个反事实推理问题。这是一个困难得多的问题，但是我们的方法仍然取得了很好的效果。

五、总结

综上所述，在这份工作中，我们提出了一种将共形推理用于反事实和个体处理效应的方法。重要的是，它十分可靠，这体现在：在随机实验中，对于有限样本而言，使用任何的黑盒算法都可以取得接近精确的覆盖率。

而在观测性研究中，我们的方法拥有双重鲁棒性，它确保了最小的覆盖率（置信度）。而在我们的仿真实验和真实数据上的研究中，我们确实在实际中观测到了这种性质，它不仅仅是一种理论学说！

斯坦福大学马腾宇：理解噪声协方差的隐式偏差

整理：智源社区 熊宇轩

在 2020 年北京智源大会的「机器学习前沿青年科学家专题论坛」中，来自斯坦福大学的助理教授马腾宇针对噪声对深度学习隐式正则化的影响带来了题为「理解噪声协方差的隐式偏差」的演讲。马腾宇的主要研究领域包括机器学习和算法，如非凸优化、深度学习及其理论，以及强化学习、表示学习、高维统计等。他在国际顶级会议和期刊上发表了系列的高质量论文，同时还获得了 2018ACM 博士论文荣誉奖等诸多奖项。

演讲全文如下:

一、深度学习时代的优化器

首先，我想简要地介绍一下最近深度学习理论研究领域一个非常热门的话题： 深度学习算法中的隐式正则化。在这之前，我将简要地回顾一下传统的机器学习理论，然后讨论为什么说「这些新的隐式正则化技术改变了我们对机器学习理论的理解」。

十到二十年以前，当我们讨论机器学习理论时，一种简化的视角是： 将统计数据和优化方法解耦开来。对于统计数据而言，我们需要设计合适的损失函数。在损失函数中，我们需要考虑数据和正则项。而优化方法旨在为损失函数寻找合适的优化器。

在机器学习理论中，对于统计数据来说，我们认为： 训练的正则化后的损失函数的全局最小值具有很小的测试误差。而优化方法的任务是为正则化后的损失函数找到一个优化器。结合以上两个步骤，我们可以在多项式时间内寻找到一个具有较小测试误差的解。往往，我们在损失函数为凸函数的情况下，全局最优解是唯一的，此时的优化工作便是凸优化。简而言之，这就是经典的机器学习理论。

然而，到了深度学习时代，很多事情都发生了改变。最终要求的一点就是： 最小化训练误差并不再是优化器唯一的职责。

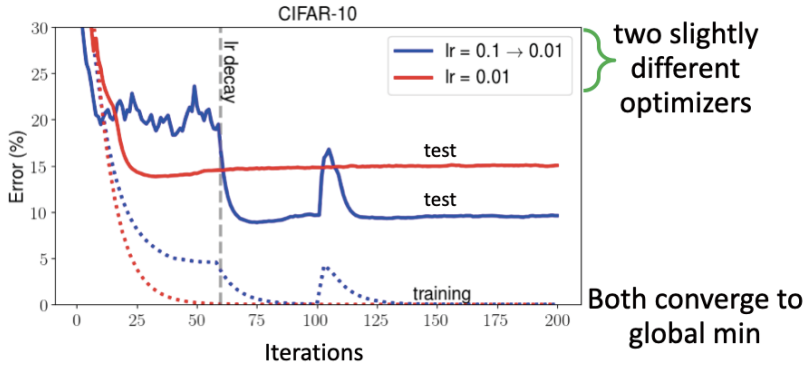


图 1：在 CIFAR-10 数据集上进行的实验。两种具有相同目标函数的算法，使用了不同的学习率，在训练损失都收敛至 0 的情况下，得到了差异很大的测试误差。

如图 1 所示，这是一个在 CIFAR-10 数据集上进行的简单的实验。图中涉及到了两种算法，他们之间仅有的差异就是其学习率。对于第一个算法（算法 1，蓝色曲线）来说，其初始学习率为 0.1，随后其学习率衰减为 0.01；第二个算法（算法 2，红色曲线）的学习率则一直为 0.01。所以，这两种算法有相同的目标函数，唯一的区别是其优化器有微小的差异。在这两种情况下，当经过了足够多轮的迭代训练后，训练误差都收敛到了 0。然而，关键的问题是，最终的测试误差则有着显著性的差异。算法 2 最后的测试误差要高于算法 1 的测试误差。

这个例子告诉我们，优化器不仅仅要做到最小化训练损失，因为最小化训练损失并不能保证测试误差也很小。不知为何，上面的算法 1 的测试误差较小，但算法 2 则不然。

从另一个角度来看，造成这种状况的原因是该目标函数是非凸函数，它有多个全局最优解 。此外，模型存在过参数化（over-parameterization）的问题（注： 测试时过拟合，但是训练较为容易收敛），参数的数量要远远大于数据点的数量。

二、具有多个全局最优解的损失函数

在这里，我使用了一个一维空间下的示例来解释非凸函数和过参数化。

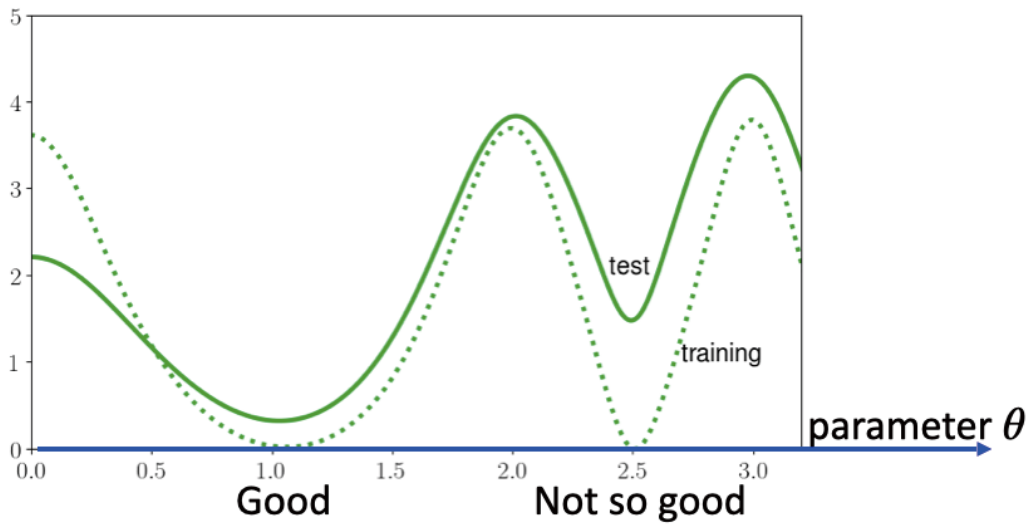


图 2：训练误差处于全局最小值、测试误差差异较大的两种参数化情况

在如图 2 所示的训练损失函数（虚线）中，我们有两个全局最小值。然而，左侧的训练损失全局最小值是较好的（测试误差较小），而右侧的全局最小值则较差（测试误差较大），这是因为训练误差和测试误差的关注点并不都是一致的，所以有一些训练误差处于全局最小值的参数情况要优于其它处于全局最小值时的情况。此时，优化方法的职责并不只是找到某一个全局最小值，而是需要找出许多潜在可能的全局最小值，进而找出「正确」（测试误差等指标也同时优异）的全局最小值。

下面，我们举一个例子来说明这种情况。在我的脑海中，我想到了我第一次来美国时，我去一个非常大的滑雪度假村游玩时的场景。在山峰之间有很多的峡谷，当你来到度假村时，你会将车停在其中的一个停车场中。而

当你最后一次滑下山坡时，你不仅仅要随便找到一个峡谷谷底（全局最小值）然后回家，还需要找到你停车的那个「正确」的峡谷。而实际上，我滑雪结束时，没有找到正确的停车场。

三、深度学习理论研究新范式

从某种程度上说，深度学习的学习理论框架需要进行革新。进行这种改变的方式是，对于统计数据来说，我们认为有一些训练函数处于全局最小值处的情况会有很小的测试误差，而不是所有的全局最小值都会有相同小的测试误差；另一方面，对于优化方法而言，优化器不仅仅需要找到损失函数的某一个全局最小值，还需要找到「正确」（性能优异）的全局最小值点。

那么什么是「正确」的全局最小值呢？我们在这里提出一个猜想：通常而言，常用的优化器（即随机梯度下降，SGD）具有隐式的对于某些简单解（从而得到简单的模型）的偏置 / 偏好，而这正是我们想要找到的「正确」的损失函数的全局最小值。另一方面，对于统计数据而言，这些特定的模型往往拥有较小的测试误差。

在某种意义上，以上就是我眼中的深度学习领域的新型学习理论范式。

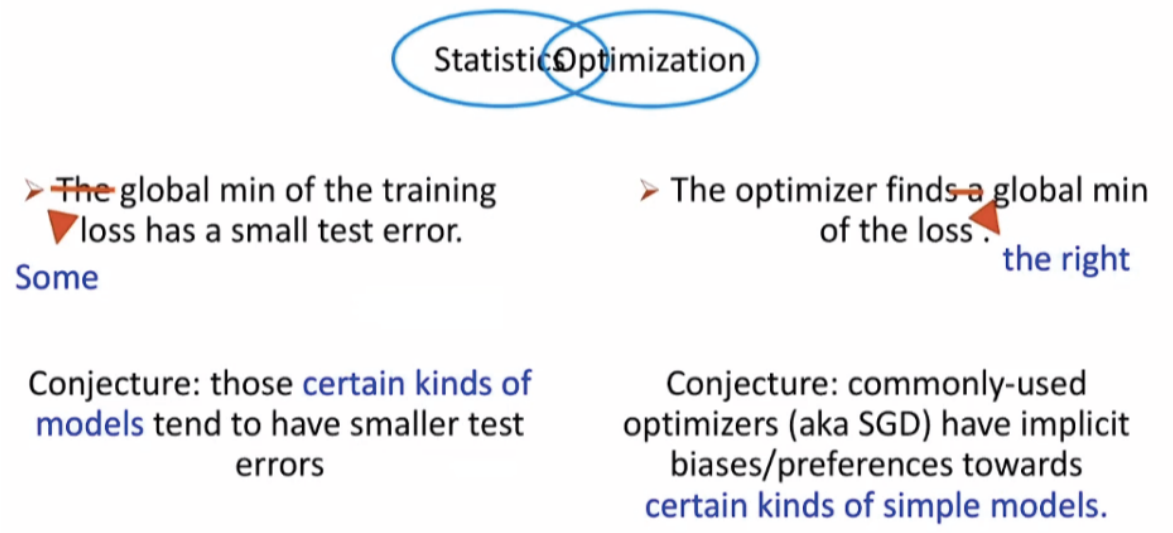


图 3：数据与优化方法趋向于融合

我们可以看到，统计数据和优化方法的结合和交互正越来越紧密。

四、优化器的隐式偏置

在过去的几年中，我认为深度学习领域最热门的一个研究方向就是，理解不同的优化器的隐式偏置。事实上，如果我们仔细想想，几乎所有优化器之间的微小差别都会引起不同的隐式偏置。而这涉及到算法中几乎所有你可以使用的不同的超参数，以及你在执行算法时所做的所有决定。

示例 1：初始化是其中的一种超参数。我们知道，使用较小的初始化参数往往偏向于得到低范数解，而另一方面，如果你使用较大的初始化，往往会出现过拟合的现象。

示例 2：使用较大的初始学习率（或较小的批处理大小），往往会学习到较为简单的模式。我们知道，如果你使用较大的学习率，并希望性能更好，那么在一开始就需要使用大学习率。在我和 Li、Wei 等人的论文中，我们对此进行了解释。

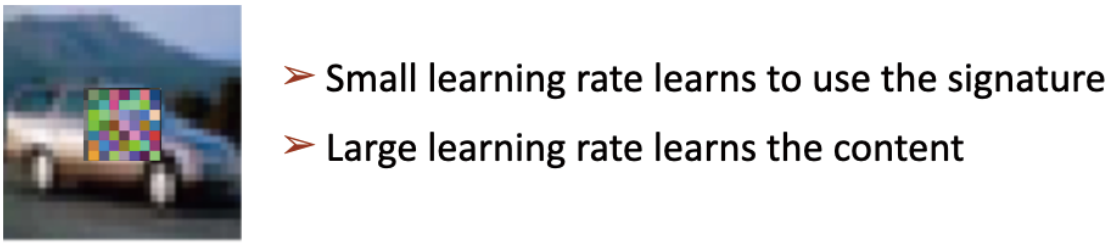


图 4： 使用不同的学习率会得到差别很大的训练结果。

在图 4 中，我们列举出了一个简单的可视化例子来说明学习率对优化器的影响。在如图所示的汽车图片中，我们加入了一些硬编码模式（hard pattern）/ 签名（signature）。由于每种图片你有一个这样的签名（signature），所以这种签名也可以被用来预测图片的类别。如果我们使用较小的学习率，那么模型将会学着使用这种签名来预测分类； 而如果我们使用较大的学习率，那么模型将会忽略签名，学习到图片的内容。这说明使用不同的学习率会学习到不同的模式，这意味着模型会收敛到损失函数不同的全局最小值处。

示例 3：Dropout 过程中也存在着一些隐式偏置。在你使用 Dropout 时，你需要同时考虑显式正则化和隐式正则化。

我们定义 Dropout 损失，即同时对数据示例和 Dropout 掩模取期望，我们将其记为：

$$L_{\text{drop}}(F) \triangleq \mathbb{E}_x \mathbb{E}_\eta[\ell(F(x,\eta))] \neq L(F) \triangleq \mathbb{E}_x[\ell(F(x))]$$

其中，η 代表 dropout 掩模，F 代表模型，l 代表损失函数。显然，这与最原始的损失函数 L(F) 是不同的，其中 L(F) 代表标准训练目标的群体样本损失（population loss）。

$$R_{\text{drop}}(F) = L_{\text{drop}}(F) - L(F).$$

L_{drop} (F) 和 L(F) 之间的差别被称为显式正则化，因为当我们使用 Dropout 时，我们实际上优化的是 L_{drop} (F) 而不是 L(F)，这种差别可以被用作一种正则项。

然而，故事到这里并没有结束，你需要使用专门优化了 L_{drop} (F) 的 Dropout 方法，如果你使用其它任意的方法来优化 L_{drop}，你的模型泛化能力可能会较差。这说明 Dropout 中的噪声也起了很大的作用，它们不仅仅改变了损失函数，还在优化过程中引入了一些零均值随机性，而这改变了隐式偏置。

五、噪声协方差的隐式正则化效应

至此，我们简要地举出了几个关于优化器的隐式偏置的现有工作的示例。接下来，我将谈谈我的新工作： 来自噪声协方差的隐式正则化效应。优化器中的噪声协方差在隐式正则化中也起到了很重要的作用，我们将试图理解怎样的噪声会对于提升模型泛化能力起到正向的作用，并探究其背后的原因。

下面，我们首先介绍一下该研究的背景。

根据经验，我们认为验证误差越小越好。众所周知，当我们使用完整的 batch 进行训练时，我们使用的是完全的梯度下降，我们将无法进行隐式正则化，模型的性能将会很差。从很多论文中我们都可以看到，如果我使用完整的 batch 进行训练，我们必须通过一些额外的方法来提升 full batch 梯度下降的泛化性能。

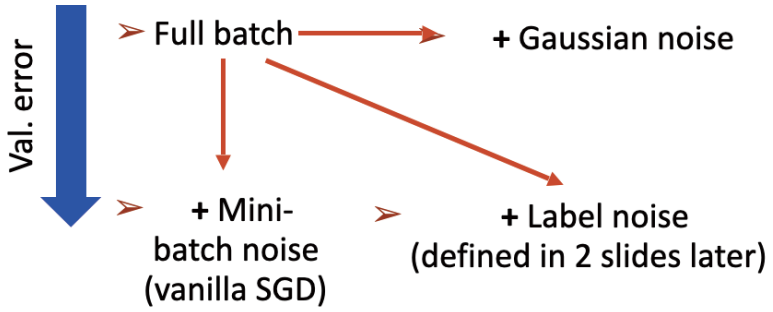


图 5：噪声协方差

我们知道，解决该问题的方法之一是： 在梯度中引入额外的噪声。Mini-batch 随机梯度下降算法（原始的 SGD）就是实现这一思想的具体的一种方法。如果我们将这种噪声加入到完整梯度中，就可以得到更小的泛化误差。如果引入人们在分布式计算领域使用的标签噪声（Label Noise）技术，也可以得到与 Mini-batch 方法相近的性能。然而，如果我们向 full batch 梯度中加入了一些错误的噪声（例如高斯噪声），那么可能我们无法获得性能的提升，或者提升幅度很小（这取决于你如何对高斯噪声的权重进行调优）。

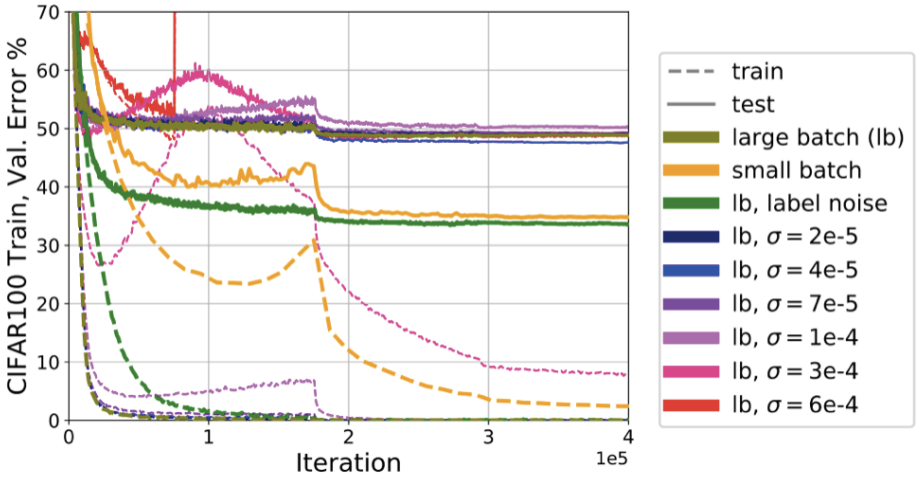


图 6：噪声协方差对验证误差的影响

图 6 显示了一个我们在 CIFAR-100 数据集上做的实验，y 轴代表验证误差（越小越好），x 轴代表训练的迭代次数。我们尝试使用了很多种算法，使用了较大 batch 规模的算法（对应于图中的「large batch」）性能较差，我们用它来代表使用完整 batch 时的情况（因为使用完整 batch 的运行时间过长）。我们可以看到使用较小 batch 的「small batch」（黄色曲线）和使用标签噪声的「label noise」（绿色曲线）时，模型的泛化误差要比使用「large batch」时好很多。如果我们在「large batch」的情况下加入高斯噪声，泛化误差将减小 1 个百分点左右。 而，使用「small batch」或「label noise」时，模型性能将相较于使用「large batch」时提升至少 10 个百分点。

这个实验说明，特定的噪声协方差确实对模型的泛化误差有影响，但高斯误差似乎作用并不大。

接下来，我们将研究为什么会出现上述情况。我们知道研究深度学习是十分困难的，其中有各种各样的非线性变化。因此，在这里我们使用了一个简化的模型来研究噪声的影响。该模型由 Vaskevicius 和 Woodworth 等人于 2019 年提出，并用于研究学习率、噪声等因素对隐式正则化的影响。

Introduced and studied by [Vaskevicius et al.'19, Woodworth et al.'19]

- Input $x \in \mathbb{R}^d$ from spherical Gaussian
- Output $y = \langle v^\star \odot v^\star,\ x \rangle$
- Parameterization $f_v(x) = \langle v \odot v,\ x \rangle$
- Given n samples where $n \ll d$
- Sparse ground truth: $v^\star$ is r-sparse
- Similar phenomenon: label noise or mini-batch noise generalizes better than Gaussian noise or no noise

图 7：研究噪声协方差的简化模型

在这里，我们将这个简单的模型重参数化后用于了线性模型。当然，你需要将这种线性模型的情况推广到非凸函数的情况下，这样你就可以看到隐式正则化的作用。

假设，该模型的输入是一个采样自球形高斯函数的 d 维向量 $x \in R^d$；输出是关于 x 的基本线性函数，记作 $y = v^* \odot v^*, x$。其中，v^* 是一个代表真实值（ground truth）的向量，$\odot$ 代表点乘（element-wise product）。将 $v^* \odot v^*$ 的结果与 x 做内积，从而得到输出 y。与对真实值的假设相同，在参数化的过程中，我们使用了参数 v，而模型的输出为 $f_v(x) = v \odot v, x$。当 v 与 v^* 相等时，我们就早找到了最优解。

在这里，我们假设发生了过参数化现象，我们假设数据点 x 的维度 d 要远大于样本的数量 n，我们没有足够的样本量来学出较好的参数 v。此时，我们就需要依靠隐式正则化技术去学习真实值的参数。同时，我们还假设真实参数向量 v^* 是「r- 稀疏」（r-sparse）的。

从信息论的角度来说，你可以使用这种信息去学习真实参数。但是在算法中，我们并不会使用任何的正则化技术。所以，我们可以使用不同的参数化方法，例如，令 $\mathbf{u} = \mathbf{v} \odot \mathbf{v}$，对于一个 u 的线性函数而言，我们可以使用 Lasso 方法找到可行的稀疏解。

然而，此时的关键之处并不在于找到最佳的方法求解问题，而是认识到我们可以通过一种简单的模型理解算法的隐式正则化。如果我们不加任何的正则化处理，而在 v 的空间中使用这种标准的 L2 损失，你会看到一种非常相似的现象：使用标签噪声或 mini-batch 噪声比使用高斯噪声或不适用噪声的泛化性能更好。而这正是我选择这种简化模型研究噪声协方差的原因。下面我们将重点讨论这一现象。

六、当噪声被引入优化器

经验损失 L(v) 是一种标准的 L2 损失，显然 L(v) 中并没有正则项，该损失有很多的全局最小值点，x 的维度要远大于样本量，自由度很大。

- Empirical loss

$$L(v) = \frac{1}{n} \cdot \sum_{i=1}^{n} (y^{(i)} - f_v(x^{(i)}))^2$$

- Gradient descent

$$v \leftarrow v - \eta \nabla L(v)$$

- SGD

$$v \leftarrow v - \eta \nabla \left(y^{(i)} - f_v\left(x^{(i)}\right) \right)^2$$

- SGD with label noise

$$v \leftarrow v - \eta \nabla \left(y^{(i)} + \xi - f_v\left(x^{(i)}\right) \right)^2$$

- SGD with spherical Gaussian noise / Langevin Dynamics

$$v \leftarrow v - \eta \nabla L(v) + \xi$$

- All updates are unbiased estimator of the gradient $\nabla L(v)$

图 8：带有各种噪声的正则项

图 8 中的梯度下降指的就是标准的梯度下降算法 $v \leftarrow v - \eta \nabla L(v)$。而 SGD 则是我们取 $\left(y^{(i)} - f_v\left(x^{(i)}\right) \right)$，计算出其梯度，用 $v \leftarrow v - \eta \nabla \left(y^{(i)} - f_v\left(x^{(i)}\right) \right)^2$ 来更新 v。而我们都可能对标签噪声很熟悉，此时我们引入了一个具有零均值的随机变量 ξ，将其与标签 $y^{(i)}$ 相加，计算出其梯度后，通过 $v \leftarrow v - \eta \nabla \left(y^{(i)} + \xi - f_v\left(x^{(i)}\right) \right)^2$ 更新 v。ξ 对于噪声的影响是二阶的，但是如果我们在此使用的是其梯度，那么 ξ 的影响就是一阶的。所以这个带有噪声的损失函数的梯度实际上仍然是真实梯度的一个无偏估计量。

为了对比，我们也考虑了向 SGD 加入球形高斯噪声 / 郎之万动力学的情况。在这里，我们需要做的是，我们取完整的梯度，然后向其中加入球形高斯噪声 ξ (与之前的 ξ 不同，这里的 ξ 是一个向量)，通过 $v \leftarrow v - \eta \nabla L(v) + \xi$ 更新参数 v。以上这些算法更新的损失函数的梯度都是对梯度 $\nabla L(v)$ 的无偏估计。所以，这意味着它们都可以解决训练中存在的 (隐式偏置) 问题。

然而，你仍然能看到，在许多情况下，以上的算法的泛化性能是不同的。下面我们将展示我们主要的理论研究成果，它们展现了我们前面讨论的各种优化算法的性能。

假设 n 远大于 r^2 但是远小于 d。这意味着我们面临过参数化的情况，但是我们有 n 远大于 r^2。所以从信息论的角度来说，我们可以恢复出真实值 (ground truth)。

之前的一些研究工作介绍了这种研究梯度下降的模型。它们研究了不同的初始化情况，以及这些初始化情况下的隐式偏置。

众所周知，如果你使用较大的初始化宽度和足够小的学习率，那么通常而言，你所做的事情基本上与神经切线核 (NTK) 类似。这意味着，以试图在核空间 (kernel space) 中寻找最小范数解。在这种情况下，如果你使用较大的初始化宽度 (CNN 中卷积核的通道数，或者 FC 层的神经元数)，会发生过拟合现象，这是因为你没有足够的数据。而当 n 的阶为 O(d) 时，核函数的泛化能力可以得到提升。而如果我们使用极小的初始值，我们可以恢复出真实值 (ground truth)。使用较小的初始值往往会得到较小的范数解。

接下来，我们将讨论带有标签噪声的 SGD 优化器。

当我们使用带有标签噪声的 SGD 方法时，无论初始值如何，使用该优化器优化的参数都会收敛到真实值 v* 上，这意味着这种方法对于较大的初始值并不敏感。即使我们使用较大的初始值，噪声会帮助我们降低解的范数，v* 会收敛到稀疏的解上，这对于数据来说是过拟合的。

另一方面，加入我们并不使用标签噪声，转而使用带有高斯噪声的 SGD (郎之万动力学)。可以说明，郎之万动力学并没有一个具备有限配分函数的固定的吉布斯分布，这说明这种分布并不存在，因为配分函数并不是有限的。因此，郎之万动力学并不会收敛到一个固定的分布上。

在这里，带有标签噪声和高斯噪声的 SGD 方法形成了对比。加入了标签噪声的 SGD 往往会收敛到真实值的稀疏解 v* 上，而带有高斯噪声的 SGD 则不具备这一性质。而我们没有分析 mini-batch SGD 和原始的 SGD 的原因是，它们对初始化非常敏感，这会使得分析十分困难。如果我们恰巧在初始情况下陷入了过拟合解，那么 SGD 算法则不会在损失函数上移动。从某种程度上来说，标签噪声要比 SGD 噪声更好一些，因为即使我们应陷入了过拟合状况，也可以得到一些噪声。

标签噪声和高斯噪声的对比说明噪声协方差确实很重要，接下来我们会讨论为什么它之所以重要的原因是什么。

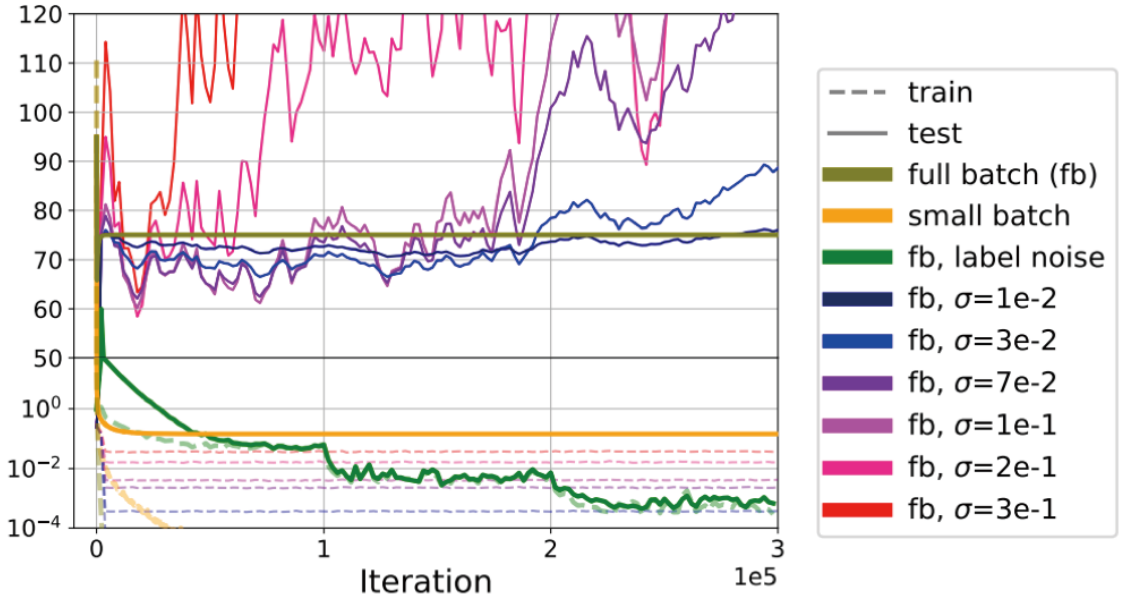


图 9：带噪声 SGD 的仿真实验

在进行理论分析之前，我们先迅速地来看一下对于这个简化案例的仿真实验，从而证明我们的理论描述 / 理论结论与实证研究观测结果相符。如上图所示，通过运行上述的算法，我们可以看到，带有标签噪声的 SGD（绿色曲线）或 mini-batch SGD（黄色曲线）确实对于性能提升有所帮助，它们的泛化误差均收敛到接近于 0 的值，而此时带有标签噪声的 SGD 的泛化性能比 mini-batch SGD 还要更好。

另一方面，如果我们不使用噪声（fb）或使用高斯噪声，会发生显著的过拟合现象，测试误差会很高。如果我们使用较大的高斯噪声，泛化误差会立刻爆炸式增长；而如果我们使用较小的高斯噪声，在迭代初期似乎相较于 fb 并没有太大的影响，但最后当郎之万动力学中的因素相互作用（mixing，微小的误差在长时间的过程中可能被不停积累和放大）时，泛化误差也会爆炸增长。对于以上所有算法而言，它们的训练误差都可以收敛到很小的值上，但是其测试误差却有时很大，这是一个重要的泛化问题。

七、研究动机与理论分析

关于「为什么这些算法会有区别」的问题，根据直觉，传统的看法是：SGD 往往会找到最「平坦」的局部最小值。我始终认为这种看法是正确的，它在所有的情况下都起到了很重要的作用。例如，在加入高斯噪声的情况下，郎之万动力学会收敛到吉布斯分布上。随着郎之万动力学中的温度 T 收敛到 0，郎之万动力学算法可以被支撑在全局最小值的流形上。在下图所示的简单的可视化结果中，白色的明亮的部分就是全局最小值。

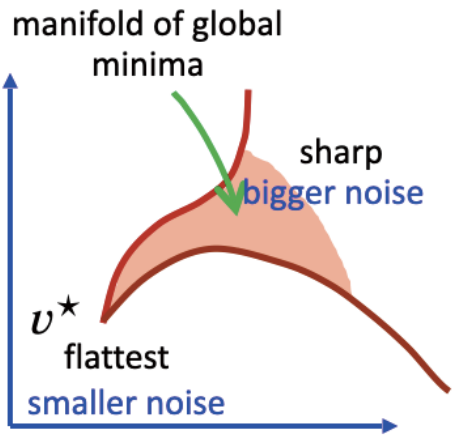


图 10： SGD 会找到最「平坦」的局部最小值

现在的问题是，如何在温度趋向于 0 的过程中度量全局最小值的流形。根据郎之万动力学的定义，吉布斯分布将更多的质量分配在最平坦的全局最小值处，在本例中这一点就是 v*。那这样做为什么不起作用呢？事实上，所有其它的全局最小值都要比 v* 处的全局最小值在流形上更加尖锐，而这样尖锐的性能很差的全局最小值点太多，他们都会导致流形的尖锐度（sharpness）增加。所以，有时我们的算法会收敛到这种尖锐的全局最小点上，这会导致在迭代训练的过程中泛化误差爆炸增长。

而另一方面，为什么在 SGD 中加入标签噪声是有效的呢？在本文中，我们指出存在另一个隐式偏置的源头：SGD 也偏向于找到具有较小噪声的局部最小值点。在使用高斯噪声时，这种情况是不存在的，因为那时每一点上的噪声是相等的，即使算法试图收敛到最平坦的局部最优解，也不会偏向于收敛到具有较小噪声的局部最优解。

我们认为，当我们使用带有标签噪声的 SGD 时，存在两个隐式偏置的源头：（1）曲率 （2）噪声大小的变化。在本例中，以上两点足够使我们收敛到真实值处，但是只有「曲率」是不够的。因此，与其它的全局最小值点相比，v* 不仅仅是最平坦的，也具有较小的噪声。此时，标签噪声可以写作：

$$\text{Label noise} = \eta \cdot \xi \cdot \nabla f_v\left(x^{(i)}\right) = \eta \cdot \xi \cdot v \odot x^{(i)}$$

根据上面的公式，我们可以看出标签噪声不仅仅取决于数据的规模，也取决于参数的规模。标签噪声的大小与参数规模的大小成正比，这也正是 v* 的噪声在所有的全局最小值最小的原因。

以上是我们最主要的研究直觉，我也介绍了我们如何对算法之间的差异进行分析、如何形式化定义这个问题、如何对其进行数学证明。

我们发现，在前文中，标签噪声在某种意义上类似于一种乘性噪声。下面我们将展示一个简化的场景，其中我们只考虑 1 维的随机游走，并不涉及任何梯度。你可以对一下的两种情况进行对比：（1）高斯噪声 （2）乘性噪声。其中，「高斯噪声」指高斯噪声随机游走（一种布朗运动）。「标签噪声」对应于乘性噪声，其噪声的大小与

参数相关。对于「高斯噪声」而言，我们在每次更新 v 时，加上一个η · ξ；对于「乘性噪声」，我们在每次更新 v 时，加上一个η · ξ · v。这里的差别是：如果 v 很小，那么你将得到一个较小的乘性噪声，但高斯噪声则不会受影响。在上图中的轨迹图中，高斯噪声（蓝色曲线）的 v 会增长，它可以由布朗运动预测得到。当我们使用乘性噪声（黄色曲线）时，尽管曲线也有所波动，但是随着随机游走步数的增加，它会收敛到 0。有趣的是，这两种随机游走的均值都是 1，而它们的方差都会增长。尽管方差不断增大，但是对于乘性噪声来说，v 有很大的概率收敛到 0。另一方面，对于高斯噪声来说，它的方差、v 都在增长。

为了对乘性噪声进行证明，我们使用的证明方法是：选择一种势函数（potential function）ϕ，它在 [–C, C] 的区间内是非负的凹函数。这一方法也同样适用于高维空间的情况。

首先，我们的随机游走不会超出 [–C, C] 的范围。随着我们进行这种乘性随机游走，势函数逐渐减小。在这里，我们写出更新后的势函数的期望的泰勒展开式：

$$\begin{aligned}\mathbb{E}[\phi(v+\eta\xi v)] &\approx \mathbb{E}[\phi(v)+\phi'(v)\eta\xi v+\phi''(v)\eta^2\xi^2v^2] \\ &= \mathbb{E}[\phi(v)]+\mathbb{E}[\phi''(v)\eta^2v^2] < \mathbb{E}[\phi(v)]\end{aligned}$$

在泰勒展展开式 $\mathbb{E}\left[\phi(v)\right]+\mathbb{E}\left[\phi''(v)\eta^2v^2\right]$ 中，由于第一项是零均值项，它可以被消掉。而在第二项中，由于势函数是一个凹函数，ϕ ''(v) 为负。所以第二项对于整体期望的贡献是负的，随着我们进行随机游走，势函数会递减。实际上，不仅仅势函数会递减，随机变量很可能也会递减，v 有很大概率会收敛到 0，v 确实偏向于收敛到噪声最小的点上。

请注意，这种证明方法并不适用于高斯噪声随机游走，上面的不等式此时就不成立了，因为我们无法将高斯噪声限制在区间 [–C, C] 内，我们无法在实数域上找到合适的全局非负凹势函数。

总而言之，噪声协方差确实对隐式正则化有很大影响，而 SGD 偏向于收敛到具有较小噪声的参数上，而不仅仅偏向于流形上较小的曲率。

我们提出一个开放性问题：我们是否能够理解 mini–batch SGD？正如前文所述，此时最大的难题是：如果我们刚好进行了错误的初始化，或者在算法进行的过程中我们刚好陷入了过拟合现象，噪声突然变为 0，此时损失就不会改变。这会使我们非常难以进行进一步的分析，所以我们对于初始化是非常敏感的。因此，我们需要说明这种算法在使用随机初始化的情况下不会非常快地收敛到某一点。如果收敛过程非常快，那么噪声就会迅速减少，算法就会失效。以上就是我针对隐式正则化展开的讨论。

八、显示正则化——解耦统计数据与优化方法的另一个视角

最后，我想从另外的一个视角来谈谈如何通过研究「显式正则化」将统计数据和优化方法解耦。这与我们之前讨论的的内容有很大的区别。

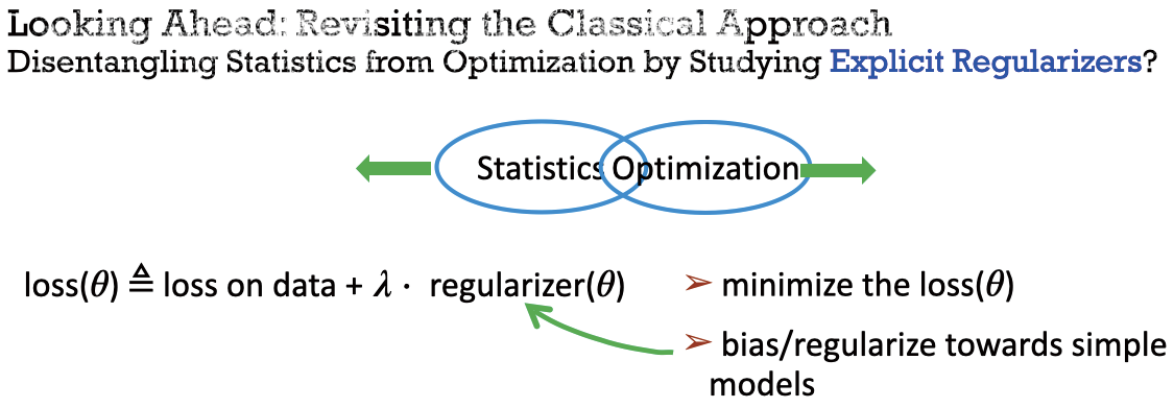


图 11：显示正则化——解耦统计数据与优化方法的另一个视角

从某种意义上说，经典的方法的性能仍然是很理想的。它们可以很「干净」地区分对于统计数据和优化方法的分析。但是在深度学习时代，所有的事情都变得非常复杂，我们需要同时考虑统计数据和优化方法。将统计数据和优化方法解耦的方式之一就是，使用新的算法，因为现在的算法将两者混合了起来。而我们可以通过显式正则化技术对其进行解耦。

当我们考虑统计数据时，我们需要设计损失函数，而在使用现在的算法时，优化方法有两个功能：（1）最小化损失（2）给出正确的对于简单模型的偏置 / 正则化过程。然而，如果我们将（2）的工作更多地转移到对统计数据的处理过程中，那么我们就可以减轻优化方法的负担。我们希望，假如我们有一个很强的正则项，那么这个正则项就有可能控制算法的偏置。我们就不需要过度依赖于算法的偏置，从而进行正则化。这些正则项试图替代隐式正则化过程，他们也可以在隐式正则化效果不佳的情况下更好地起作用。例如，对于不平衡的数据集来说，隐式正则化并不能让模型得到很好的泛化性能，我们需要采取一些更为主动的行动。