高斯分数匹配的变分推断

摘要

变分推断（VI）是一种近似计算贝叶斯统计中难以处理的后验分布的方法，通常通过最小化某个目标（如证据下界 ELBO）来找到一个简单的参数化分布，以近似目标后验分布。本研究提出了一种基于分数匹配原则的新型变分推断方法。分数匹配的核心是：如果两个分布相等，它们的分数函数（即对数密度的梯度）在每个点上也应相等。分数匹配的变分推断（score matching VI）是一种迭代算法，通过最小化变分近似与精确后验分布分数之间的差异来优化变分分布。本文证明，当变分族为高斯分布时，这个优化问题有一个闭式解，称为高斯分数匹配变分推断（GSM-VI)。GSM-VI 是一种“黑箱”变分算法，因为它只需要一个可微的联合分布，无需对具体模型做特别的假设，适用范围非常广泛。将 GSM-VI 与另一种优化目标是 ELBO 的黑箱变分推断方法 BBVI 进行比较，实验结果表明，GSM-VI 表现出更高的效率和更低的计算开销，不仅速度更快，而且不牺牲准确性，在相同的近似质量下，所需的梯度评估次数更少。

关键词：变分推断；高斯分布；贝叶斯推断

1 引言

变分推断（Variational Inference, VI）是一种用于概率模型中近似推断的方法，广泛应用于贝叶斯机器学习和统计领域。在复杂的概率模型中，直接计算后验分布往往是不可能的，因为后验分布可能没有解析解，或者求解的计算成本过高。这种困难通常出现在模型具有大量隐藏变量或高维参数的情况下，例如主题模型、隐马尔科夫模型以及深度生成模型中。变分推断通过将复杂的后验推断问题转化为优化问题，以一种可控且高效的方式找到后验分布的近似解。

变分推断的核心思想是用一个可调节的简单分布族来逼近复杂的后验分布。具体而言，变分推断引入一个参数化的变分分布，并通过最小化变分分布与真实后验分布之间的差异来完成近似。这种差异通常用 Kullback-Leibler（KL）散度来量化。通过优化这个目标函数，即将最小化 KL 散度等价于最大化证据下界（Evidence Lower Bound, ELBO），变分推断在近似分布中找到最接近后验分布的那一个。这种方法避免了直接计算后验的高维积分问题，将其转化为梯度优化问题，因此可以利用现代优化工具和硬件加速实现高效推断。

变分推断的意义在于，它为复杂概率模型提供了一种通用且灵活的推断方法，使得研究者可以在高维复杂模型中进行贝叶斯推断，进而得到参数的不确定性估计和预测分布。相比于传统的采样方法如马尔可夫链蒙特卡罗（MCMC），变分推断在计算效率上具有显著优势，尤其适用于大规模数据和实时应用。此外，它还与深度学习方法高度兼容，例如变分自编码器（VAE）就将变分推断与神经网络结合，实现了生成模型的高效训练。

然而，变分推断也有其局限性。由于 KL 散度的非对称性，变分推断倾向于低估后验分布的尾部，从而可能导致重要后验信息的遗漏。此外，选择适当的变分分布族和优化算法也具有挑战性。在实际应用中，为了克服这些问题，研究者提出了许多扩展和改进方法，例如黑箱变分推断（BBVI）、随机梯度变分推断（SGVI）以及基于熵正则化的变分方法等。这些发展极大地扩展了变分推断的适用范围，使其成为现代统计机器学习的重要组成部分。

本文提出了一种高斯分数匹配的变分推断（GSM-VI）。这项工作的一个新颖之处在于 GSM-VI 如何拟合变分参数。它不是最小化损失函数，而是求解一组非线性方程，因此不依赖随机梯度下降（SGD）进行核心优化。比起传统的黑箱变分推断，GSM-VI 不需要调节超参数，因此比 BBVI 更稳定。本文通过实验发现，在多元高斯分布及非高斯分布的后验近似中，GSM-VI 在收敛速度和计算效率上显著优于 BBVI，且在收敛更快的同时不会牺牲准确性，同时对目标分布的维度和条件数表现出更好的鲁棒性。 [1]

2 相关工作

2.1 BBVI

黑箱变分推断（Black-box Variational Inference, BBVI）是变分推断中的一项重要技术，其核心思想是通过优化证据下界 ELBO 来逼近后验分布。BBVI 有如下假设：目标密度 p 无法进行精确的点评估或采样；但可以获得非规范化的目标密度；对数目标密度是可微的，其导数可以有效计算。BBVI 的关键特性是其“黑箱”属性，意味着研究者只需要目标分布的对数联合概率及其梯度信息，而不需要显式计算后验分布。这种灵活性使 BBVI 成为一个通用的变分推断工具，广泛应用于各种复杂的概率模型中。BBVI 的实现依赖于随机梯度下降的优化方法，使用采样技术估计梯度以减少计算开销。其通用性和相对简单的实现使其成为现代概率编程系统中的核心组件。

然而，BBVI 也存在明显的局限性。首先，它对超参数（例如学习率）的设置非常敏感，不恰当的选择可能导致优化过程收敛缓慢甚至失败。此外，BBVI 在高维复杂模型中的表现往往受到梯度估计高方差的影响，尤其当后验分布的尾部较重或与变分分布族的形状不匹配时，可能导致近似质量较低。尽管其黑箱特性为许多应用提供了便利，但其效率问题在大规模数据集和高维参数空间中仍然是一个主要瓶颈。

2.2 ADVI

自微分变分推断（Automatic Differentiation Variational Inference, ADVI）是在 BBVI 基础上的进一步发展，致力于通过自动微分技术简化变分推断的实现过程。ADVI 的核心创新在于，它结合了自动微分和变分优化，使研究者能够更方便地计算复杂模型的梯度。这种方法通常以多元高斯分布为变分分布族，并优化 ELBO 以逼近后验分布。ADVI 的自动化特性使其能够无缝集成到概率编程框架中，例如 Stan 和 PyMC3，大幅降低了模型开发和实验的复杂性。对于初学者和需要快速开发的应用场景，ADVI 提供了非常有效的工具支持。

尽管 ADVI 改善了 BBVI 的易用性，但其本质仍然依赖于随机梯度优化，因此继承了 BBVI 的一些主要缺点。例如，它同样对学习率敏感，且在高维复杂模型中仍可能遇到梯度估计不稳定的问题。此外，ADVI 倾向于假设后验分布与多元高斯分布的匹配性较好，这在某些

非高斯后验场景中可能导致次优的近似结果。尽管如此，ADVI 作为 BBVI 的增强版本，其自动化能力在实际应用中极大地提升了 VI 方法的可用性和效率。

2.3 BaM

Batch and Match（BaM）则提供了一种新的思路，它是一种基于分数的替代方法，旨在改进 BBVI，通过将优化问题分解为多个独立的小批次子问题，极大提升了优化的效率。在高维变分推断问题中，传统优化方法可能面临的一个主要瓶颈是全局搜索的计算复杂性。BaM 通过在局部数据子集上进行优化，并将这些局部近似解匹配到一个全局框架中，从而以更少的计算代价实现快速收敛。这种方法尤其适合处理数据规模大、模型维度高的场景，在分布式计算环境下表现出显著的优势。此外，通过分解后的子问题优化，BaM 也能够部分缓解梯度估计中的高方差问题。然而，这种方法的性能依赖于如何设计数据子集的分割方式以及匹配全局分布的机制。

BaM 能够容纳表现力更强的变分族，同时摆脱了对随机梯度下降的依赖，利用闭式近似更新进行优化，尤其适用于具有全协方差矩阵的高斯变分族。在收敛性方面，当目标分布为高斯分布且批量无限大时，BaM 的变分参数能够以指数级速度迅速收敛到目标的均值和协方差。性能评估表明，无论目标分布是高斯还是非高斯，BaM 通常能以更少（有时显著更少）的梯度评估次数达到收敛，优于基于 ELBO 最大化的 BBVI 方法。

3 本文方法

3.1 本文方法概述

本文方法通过最小化分数函数（即对数梯度）之间的二次差异来逼近目标后验分布，有效避免了传统变分推断方法中对数似然最大化可能导致的数值不稳定问题。GSM-VI 引入了广义分数匹配目标，通过调整权重矩阵增强了方法的灵活性，并采用小批量随机优化对高斯分布的均值和协方差进行迭代更新，从而能够高效处理高维和大规模数据。

3.2 SM-VI

分数匹配变分推断（Score Matching Variational Inference, SM-VI）的算法推导基于分数匹配的核心原则：当两个分布在其定义域上相等时，其分数函数（即对数密度的梯度）也相等。通过这一原则，SM-VI 旨在构建一种通过优化分数函数的匹配度来逼近后验分布的新方法。SM-VI 将分布拟合问题转化为一组分数匹配约束的迭代优化问题，避免了直接最小化 KL 散度的需求，并通过闭式解实现了高效计算。这一方法在高斯分布及其扩展应用中表现出显著的收敛速度和稳定性。以下是算法推导过程的关键步骤：

假设后验分布 $p(\theta|x)$ 难以直接计算。我们选择一个可调的变分分布族 $q_w(\theta)$，通过最小化与后验的分数差异来调整其参数 w，使得：

$$\nabla_\theta \log q_w(\theta) \approx \nabla_\theta \log p(\theta, x),$$

其中 $p(\theta, x)$ 为联合分布，且 $\nabla_\theta \log p(\theta, x)$ 是可计算的。

分数匹配基于以下约束：

$$\nabla_\theta \log q_w(\theta) = \nabla_\theta \log p(\theta, x), \quad \forall \theta \in \Theta.$$

当变分分布 $q_w(\theta)$ 足够灵活时，上述条件可以严格满足。

在每次迭代 t 中，SM-VI 从当前变分分布 $q_{w_t}(\theta)$ 中采样 θ_t，然后通过最小化分数的偏差更新参数 w：

$$w_{t+1} = \arg\min_w \mathrm{KL}(q_{w_t}(\theta) \| q_w(\theta)) \quad \text{s.t.} \quad \nabla_\theta \log q_w(\theta_t) = \nabla_\theta \log p(\theta_t, x).$$

当变分分布为多元高斯 $q_w(\theta) = \mathcal{N}(\mu, \Sigma)$ 时，分数匹配的约束条件可以通过以下闭式公式解决：

$$\mu_{t+1} = \mu_t + A_t\big(\nabla_\theta \log p(\theta_t, x) - \nabla_\theta \log q_{w_t}(\theta_t)\big),$$

$$\Sigma_{t+1} = \Sigma_t + (\mu_t - \theta_t)(\mu_t - \theta_t)^\top - (\mu_{t+1} - \theta_t)(\mu_{t+1} - \theta_t)^\top,$$

其中 A_t 为根据分数约束和当前分布参数计算的矩阵，具体形式见附录。

3.3 推广到 GSM-VI

GSM-VI 的核心思想是引入一个对称的正定权重矩阵 $\mathbf{A}(x)$，并重新定义分数匹配的目标函数：

$$L_{\text{GSM-VI}} = \mathbb{E}_{q_\phi(x)}\left[\frac{1}{2}(\nabla_x \log q_\phi(x) - \nabla_x \log p(x))^\top \mathbf{A}(x)(\nabla_x \log q_\phi(x) - \nabla_x \log p(x))\right].$$

通过展开目标函数并忽略与 ϕ 无关的常数项，有：

$$L_{\text{GSM-VI}} = \mathbb{E}_{q_\phi(x)}\left[\frac{1}{2}\nabla_x \log q_\phi(x)^\top \mathbf{A}(x) \nabla_x \log q_\phi(x)\right] - \mathbb{E}_{q_\phi(x)}\left[\nabla_x \log q_\phi(x)^\top \mathbf{A}(x) \nabla_x \log p(x)\right].$$

其中：第一项是 $q_\phi(x)$ 的分数函数的加权二次项；第二项是 $q_\phi(x)$ 和 $p(x)$ 的分数函数之间的加权内积。

SM-VI 是 GSM-VI 的特例，当权重矩阵 $\mathbf{A}(x)$ 是单位矩阵 $\mathbf{I}$ 时，GSM-VI 的目标函数退化为 SM-VI 的目标函数：

$$L_{\text{GSM-VI}}\big|_{\mathbf{A}(x)=\mathbf{I}} = L_{\text{SM-VI}}.$$

因此，通过选择适当的权重矩阵 $\mathbf{A}(x)$，GSM-VI 能够实现对分数匹配过程的灵活控制。例如，$\mathbf{A}(x)$ 可以设计为与数据分布相关的矩阵，以提高算法的稳定性。

对于 GSM-VI 的优化，通常采用随机梯度下降法，并利用重参数化技巧计算梯度。假设 $x \sim q_\phi(x)$，目标函数的梯度为：

$$\nabla_\phi L_{\text{GSM-VI}} = \mathbb{E}_{q_\phi(x)}\left[\mathbf{A}(x)(\nabla_x \log q_\phi(x) - \nabla_x \log p(x)) \cdot \nabla_\phi \log q_\phi(x)\right].$$

通过对分布 $q_\phi(x)$ 进行采样，可以近似计算上述梯度，并对参数 ϕ 进行迭代更新。

论文给出了如下图的伪代码：

Algorithm 1: Gaussian Score Matching VI

Input : Initial mean estimate $\boldsymbol{\mu}_0$, initial covariance estimate $\boldsymbol{\Sigma}_0$, target distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$, number of iterations $N \in \mathbb{N}$, batch size $B \in \mathbb{N}$.

Output : Multivariate normal variational distribution $q_{\boldsymbol{w}}(\boldsymbol{\theta}) := \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

for $i = 0, \ldots, N-1$ ▷ `iteration loop`

do

  for $j = 0, \ldots, B-1$ ▷ `batch loop`

  do

    Sample $\boldsymbol{\theta}^{(j)} \sim \mathcal{N}(\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$

    $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}^{(j)}|\boldsymbol{x})$

    $\boldsymbol{\varepsilon} \leftarrow \boldsymbol{\Sigma}_i \boldsymbol{g} - \boldsymbol{\mu}_i + \boldsymbol{\theta}$

    Solve $\rho(1+\rho) = \boldsymbol{g}^\top \boldsymbol{\Sigma}_i \boldsymbol{g} + \left[(\boldsymbol{\mu}_i - \boldsymbol{\theta})^\top \boldsymbol{g}\right]^2$ for $\rho > 0$

    $\boldsymbol{\delta\mu}^{(j)} \leftarrow \frac{1}{1+\rho}\left[\mathbf{I} - \frac{(\boldsymbol{\mu}_i - \boldsymbol{\theta})\boldsymbol{g}^\top}{1+\rho+(\boldsymbol{\mu}_i - \boldsymbol{\theta})^\top \boldsymbol{g}}\right]\boldsymbol{\varepsilon}$

    $\boldsymbol{\mu}_{\boldsymbol{i}}^{(j)} \leftarrow \boldsymbol{\mu_i} + \boldsymbol{\delta\mu}^{(j)}$

    $\boldsymbol{\delta\Sigma}^{(j)} \leftarrow (\boldsymbol{\mu}_i - \boldsymbol{\theta})(\boldsymbol{\mu}_i - \boldsymbol{\theta})^\top - (\boldsymbol{\mu}_{\boldsymbol{i}}^{(j)} - \boldsymbol{\theta})(\boldsymbol{\mu}_{\boldsymbol{i}}^{(j)} - \boldsymbol{\theta})^\top$

  end

  Update $\boldsymbol{\mu_{i+1}} \leftarrow \boldsymbol{\mu_i} + \sum_j \boldsymbol{\delta\mu}^{(j)}/B$

  Update $\boldsymbol{\Sigma_{i+1}} \leftarrow \boldsymbol{\Sigma_i} + \sum_j \boldsymbol{\delta\Sigma}^{(j)}/B$

end

$q_{\boldsymbol{w}}(\boldsymbol{\theta}) \leftarrow \mathcal{N}(\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$

图 1. GSM-VI 伪代码

其中 μ_0 和 Σ_0 是随机的初始均值和协方差，N 是 epoch，B 是 batch size，ϵ 是一个中间变量，用于计算偏移量，ρ 是二次方程的正根。对于每个 epoch，从当前分布中采样 batch size 个样本，对每个样本计算分数 g，也就是 log_p 的梯度，根据公式计算 μ 和 Σ 的增量。采样完毕后，更新 μ 和 Σ 为下一时刻，也就是上一时刻加上 batch size 个增量的均值。另外，因为 Σ 的更新依赖 μ，所以必须按顺序依次更新。

4 复现细节

4.1 与已有开源代码对比

作者提供了代码 GSM 与 ADVI 在相同随机多元高斯分布下的拟合情况比较。复现时，在作者已给出的开源代码的基础上，另外加上了同条件下与 BaM 的拟合进行的比较。

4.2 实验环境搭建

实验环境：Python 3.10.12，依赖：numpy1.26.4，opencv-python4.10.0.84，opencv-contrib-python4.10.0.84，jax0.4.33。

4.3 创新点

在作者提供的代码中，仅对 GSM 和 ADVI 方法在拟合随机多元高斯分布上的表现进行了对比分析。复现时，额外引入 BaM 算法，对比了三种方法在不同维度数据上的拟合效果，并通过可视化手段展示了它们在高维多元高斯分布拟合中的具体表现。

5 实验结果分析

在复现过程中，设置 ADVI 使用学习率为 $1e^{-5}$ 的 Adam 优化器进行优化；BaM 方法引入了 λ 函数正则化以增强模型的鲁棒性。评估指标采用反向 KL 散度（即 $KL(q(\theta)||p(\theta|x))$），用于量化变分分布与目标分布之间的差异，直观地反映模型的拟合性能。实验在维度 $D = 4$ 和批量大小 $B = 2$ 的设置下进行，训练 50000 个 epoch 统一了目标分布和初始条件，从而公平比较 GSM、ADVI 和 BaM 三种方法在随机多元高斯分布拟合中的表现。实验结果如下图，可以看出 GSM-VI 算法效果非常好，反向 KL 散度值非常低，说明拟合效果好，但和 BaM 算法相比收敛速度稍慢。

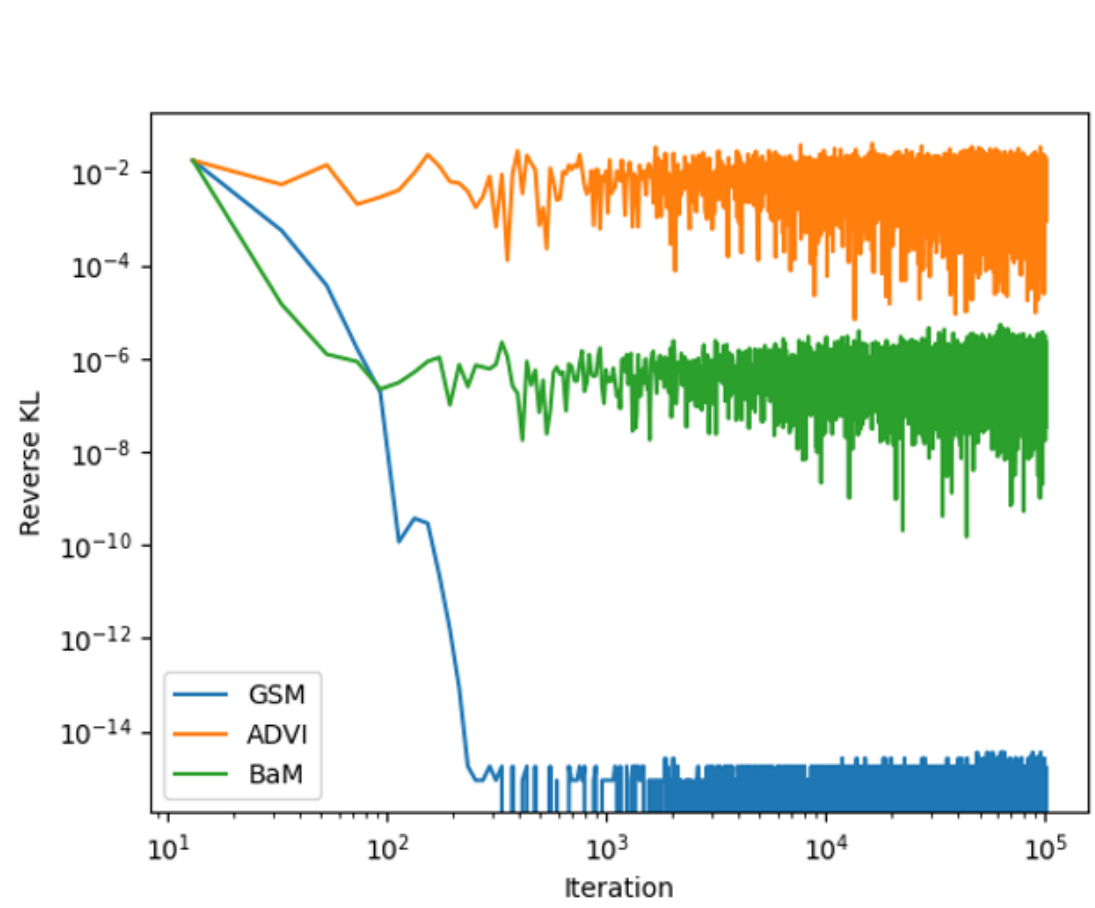

图 2. 实验结果

6 总结与展望

本次复现了论文提出的高斯分数匹配变分推断（GSM-VI）方法，并基于作者提供的开源代码，通过对比 ADVI 和 BaM 方法，探讨了 GSM-VI 在拟合随机多元高斯分布中的性能优势。实验统一了目标分布和初始条件，采用反向 KL 散度作为评估指标，分析了三种方法在不同条件下的拟合效果和收敛特性。结果表明，GSM-VI 在数值稳定性、分布拟合精度以及收敛速度方面具有显著优势。

然而，当前的实现仍然存在一些不足之处。首先，复现工作仅限于低维空间（如 $D = 4$），未能充分验证方法在高维复杂分布上的适用性。其次，BaM 方法的正则化超参数 λ 以及 ADVI

的学习率等超参数的选择相对简单，只初步进行了调整，未进行系统的超参数调优，这可能对实验结果产生一定影响。此外，在评估指标方面，虽然反向 KL 散度能够反映模型拟合的优劣，但未结合其他指标来全面评价方法性能。

未来的研究可从以下几个方向展开：首先，可以扩展实验到更高维的数据分布场景，验证 GSM-VI 方法在高维复杂任务中的表现；其次，可对超参数和评估指标的选择进行更系统的搜索与优化，以确保对比方法的最佳性能表现；再次，可以探索将 GSM-VI 应用于更广泛的实际任务，如真实数据集，图像生成或序列建模等。

参考文献

[1] Chirag Modi, Robert Gower, Charles Margossian, Yuling Yao, David Blei, and Lawrence Saul. Variational inference with gaussian score matching. *Proc. Conf. on Neural Information Processing Systems*, 36:29935–29950, 2023.