Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm

Chen Gong

22 November 2019

这篇文章要结合 Qiang Liu，写的另一片文章 "A Kernelized Stein Discrepancy for Goodness-of-fit Tests" 来阅读，这篇文章发表在 The International Conference on Machine Learning，2016。这篇文章中有对于 Stein Operator 的详细证明和理解，并且有对 Kernelized Stein Discrepancy 的详细证明，有助于深一步的理解。有关于可再生核希尔伯特空间的知识，请参考作者的另一篇文章：https://zhuanlan.zhihu.com/p/92651215。小编在理解 Stein 变分梯度下降的过程中，非常感谢裴神的指导。

1 Conference Name

Conference on Neural Information Processing Systems 2017

2 Thesis statement

2.1 大致思想描述

Stein 变分梯度下降法，是变分梯度的一种。在这类算法中，引入了 Stein 算子，Stein 特征和 Stein 类等概念。我们假设 X 是观测变量，Z 是参数和隐变量。通俗的话说就是，现有一个未知分布为 $p(Z|X)$，我们想用一个已知分布 $q(Z)$ 来近似这个分布。Stein Disparency 中提出了一种度量两个分布之间距离的方法：

$$D_{stein}(p||q) = \sup_{f \in F} |\mathbb{E}_{q(z)}[f(z)] - \mathbb{E}_{p(z|x)}[f(z)]|^2 \tag{1}$$

而 $\mathbb{E}_{p(z|x)}[f(z)]$ 中包含未知分布，计算非常的 intractable。燃煤我们像构建一个函数集 $f(z)$，使得 $\mathbb{E}_{p(z|x)}[f(z)] = 0$。然后，在 Stein 变分法中给出了一类适合的 $f(z)$，也就是

$$f(z) = \phi(z)\nabla \log p(z, x) + \nabla_z \phi(z) \tag{2}$$

将此公式代入到 $\mathbb{E}_{p(z|x)}[f(z)]$，我们可以得到如下的推导：

$$\begin{aligned}\mathbb{E}_{p(z|x)}[f(z)] =& \mathbb{E}_{p(z|x)}[\phi(z)\nabla_z \log p(z, x) + \nabla_z \phi(z)] \\ =& \int p(z|x)\phi(z)\nabla_z \log p(z, x) + p(z|x)\nabla_z \phi(z) dz\end{aligned}$$

$$
\begin{aligned}
&= \int p(z|x)\phi(z)\frac{\nabla_z p(z,x)}{p(z,x)} + p(z|x)\nabla_z\phi(z)dz \\
&= \int p(z|x)\phi(z)\frac{\nabla_z p(z|x)p(x)}{p(z|x)p(x)} + p(z|x)\nabla_z\phi(z)dz \\
&= \int \phi(z)\nabla_z p(z|x) + p(z|x)\nabla_z\phi(z)dz \\
&= \int \nabla_z(\phi(z)p(z|x))dz \\
&= \phi(z)p(z|x)|_{-\infty}^{+\infty}
\end{aligned} \tag{3}
$$

所以，需要满足一个边界条件：

$$
\lim_{|z|\longrightarrow\infty} p(z|x)\phi(z) = 0 \tag{4}
$$

当找到合适的 $f(z)$ 是公式 (2) 的 $\mathbb{E}_{p(z|x)}[f(z)] = 0$ 后，我们就只要需要计算 $\mathbb{E}_{q(z)}[f(z)]$ 就可以了。而在 Stein Variational Gradient Descent 法中，采用就是 Kernel 的方法来计算，具体的将也就是可再生核希尔伯特空间 (RKHS: Reproducing Kernel Hilbert Space)。那么现在，我们将大体的思路讲清楚了，下面我们将进行详细的分析给个部分。

2.2 可再生核希尔伯特空间

相关知识的讲解请看 https://zhuanlan.zhihu.com/p/92651215。

2.3 Stein 性质和 Stein 算子

假设 $\mathcal{X}$ 是一个 $\mathbb{R}^d$ 的子集，并且 $p(x)$ 是一个连续可微的概率密度分布，并且在定义域在 $\mathcal{X}$ 中。我们将 $p(x)$ 的 score function 定义为：

$$
s_p = \nabla_x \log p(x) = \frac{\nabla_x p(x)}{p(x)} \tag{5}
$$

对于一个函数满足：$f : \mathcal{X} \longrightarrow \mathbb{R}$，并且 $\Phi(x) = [\phi_1(x), \phi_2(x), \cdots, \phi_d(x)]^T$ 是一个光滑的函数向量，Stein 特征可以被描述为：

$$
\mathbb{E}_{x\sim p}[\mathcal{A}_p\phi(x)] = 0 \quad where \quad \mathcal{A}_p\phi(x) = \phi(x)\nabla_x \log p(x)^T + \nabla_x\phi(x) \tag{6}
$$

所以，$\mathcal{A}_p[\cdot]$ 被我们称为 Stein 算子，这个 Stein 算子有什么用呢？我们前面已经讲过了，它的主要作用是消掉计算表达式中含有未知分布的那一部分。而且这个 Stein 算子还需要满足一个零边界条件，也就是 $\lim_{||x||\longrightarrow\infty} \phi(x)p(x) = 0$，至于为什么？式 (4) 中也得到了证明。所以我们称，如果 Stein 特征满足的情况下，$\Phi(x)$ 是分布 p 的 Stein 类。

有了 Stein 类和 Stein 算子，那么我们就可以很好的引出衡量两个分布之间差异的方法也就是 Kernelized Stein Discrepancy。在这之前做了一个小铺垫，也就是两个分布之间的 score function 的差的形式。因为 $\mathbb{E}_{x\sim p}[\mathcal{A}_p\phi(x)] = 0$，所以我们可以得到：

$$
\begin{aligned}
\mathbb{E}_{x\sim p}[\mathcal{A}_q\phi(x)] =& \mathbb{E}_{x\sim p}[\mathcal{A}_q\phi(x) - \mathcal{A}_p\phi(x)] \\
=& \mathbb{E}_{x\sim p}[\phi(x)\nabla_x \log q(x)^T + \nabla_x\phi(x) - \phi(x)\nabla_x \log p(x)^T - \nabla_x\phi(x)] \\
=& \mathbb{E}_{x\sim p}[\phi(x)\nabla_x \log q(x)^T - \phi(x)\nabla_x \log p(x)^T]
\end{aligned} \tag{7}
$$

$$
\begin{aligned}
&=\mathbb{E}_{x\sim p}[\phi(x)(\nabla_x \log q(x)^T - \nabla_x \log p(x)^T)] \\
&=\mathbb{E}_{x\sim p}[\phi(x)(s_q(x) - s_p(x))^T]
\end{aligned}
$$

又因为 $\phi(x)$ 是 $d \times 1$ 维向量，$s(x)$ 是 $d \times 1$ 维向量。所以，我们可以得到：

$$
\mathbb{E}_p[trace(\mathcal{A}_q \phi(x))] = \mathbb{E}_{x\sim p}[\phi(x)(s_q(x) - s_p(x))^T] \tag{8}
$$

2.4 Kernelized Stein Discrepancy

对于两个光滑分布 p 和 q 之间的 Kernelized Stein Discrepancy 的定义，我们可以表达为：

$$
\mathbb{S}(p, q) = \mathbb{E}_{x,x'\sim p}\left[\delta_{q,p}(x)^T k(x, x') \delta_{q,p}(x')\right] \tag{9}
$$

其中，$\delta_{q,p}(x) = s_q(x) - s_p(x)$，是 score function 之间的差异，而 x, x' 是从 $p(x)$ 中采样出的，独立分布的数据点。

那么，有的同学就会想，你这样定义有用吗？你为什么要这样定义？由于 $k(x, x')$ 是一个严格的正定核函数，$\delta_{q,p}(x)^T k(x, x') \delta_{q,p}(x') \geq 0$ 恒成立。只用当 $s_q(x) = s_p(x)$ 时，才会等于 0。而且，它的计算可以非常的方便的使用核函数来完成。所以 Kernelized Stein Discrepancy 是一个衡量两个分布之间的距离的非常好的测度。

在这个问题中我们采用的是 RBF 核函数：$k(x, x') = exp\{-\frac{1}{2h^2}||x - x'||_2^2\}$。这个 RBF 核函数，同时 $\mathcal{X} = \mathbb{R}^p$ 是属于一个 Stein 类。如果觉得有问题，可以自行进行证明。

现在知道了 KSD 是什么？那么我们马上提出下一个问题，怎么计算这个 KSD 呢？

假设 p 和 q 是光滑的密度分布，并且 $k(x, x')$ 是 p 的 Stein 类。我们定义：

$$
u_q(x, x') = s_q(x)^T k(x, x') s_q(x') + s_q(x)^T \nabla_{x'} k(x, x') + \nabla_x k(x, x')^T + trace(\nabla_{x,x'} k(x, x')) \tag{10}
$$

那么，

$$
\mathbb{S}(p, q) = \mathbb{E}_{x,x'\sim p}[u_q(x, x')] \tag{11}
$$

下面我将给出详细的推导证明：我们定义 $v(x, x') = k(x, x') s_q(x') + \nabla_{x'} k(x, x') = \mathcal{A}_q k_x(x')$。而，$\mathcal{A}_q k_x(x') = k(x, x')(s_q(x) - s_p(x'))$。那么，

$$
\begin{aligned}
\mathbb{S}(p, q) =& \mathbb{E}_{x,x'\sim p}\left[(s_q(x) - s_p(x))^T k(x, x')(s_q(x') - s_p(x'))\right] \\
=& \mathbb{E}_{x,x'\sim p}\left[(s_q(x) - s_p(x))^T v(x, x')\right]
\end{aligned} \tag{12}
$$

因为，对于 p 中的任意 x' 的 $k(\cdot, x)$ 都在 Stein 类中，那么同样对于 $\nabla_{x'} k(\cdot, x')$ 也是这样。因为：

$$
\begin{aligned}
\int_x \nabla_x (p(x) \nabla_{x'} k(x, x'))dx =& \int_x \nabla_x p(x) \nabla_{x'} k(x, x') + p(x) \nabla_x \nabla_{x'} k(x, x')dx \\
=& \nabla_{x'} \int_x \nabla_x p(x) k(x, x') + p(x) \nabla_x k(x, x')dx \\
=& \nabla_{x'} \int_x \nabla_x (p(x) k(x, x'))dx \\
=& 0
\end{aligned} \tag{13}
$$

所以，$v(x,x')$ 任然在一个 Stein 类中。所以，

$$\begin{aligned}
\mathbb{S}(p,q) =&\mathbb{E}_{x,x'\sim p}\left[(s_q(x)-s_p(x))^T v(x,x')\right]\\
=&\mathbb{E}_{x,x'\sim p}\left[s_q(x)^T v(x,x')-s_p(x)^T v(x,x')\right]\\
=&\mathbb{E}_{x,x'\sim p}\left[\mathcal{A}_q v(x,x')\right]\\
=&\mathbb{E}_{x,x'\sim p}\left[s_q(x)^T v(x,x')+\nabla_x v(x,x')\right]\\
=&\mathbb{E}_{x,x'\sim p}\left[s_q(x)^T(k(x,x')s_q(x')+\nabla_{x'}k(x,x'))+\nabla_x(k(x,x')s_q(x')+\nabla_{x'}k(x,x'))\right]\\
=&\mathbb{E}_{x,x'\sim p}\left[s_q(x)^T k(x,x')s_q(x')+s_q(x)^T\nabla_{x'}k(x,x')+s_q(x)^T\nabla_x k(x,x')+trace(\nabla_{x,x'}k(x,x'))\right]
\end{aligned} \tag{14}$$

做到这里，我们已经成功的得到了 $\mathbb{S}(p,q)$ 的计算形式，那么怎么具体的算出来呢？这就要用到关于 $k(x,x')$ 的特征分解了，我们将会展示 $\mathbb{S}(p,q)$ 如何在 $k(x,x')$ 的矩阵的特征值上，同时高效的使用 Stein 算子。

假设 $k(x,x')$ 是一个正定核函数，特征向量和特征值为 $e_j(x)$ 和 λ_j。并且，$u_q(x,x')$ 也是一个正定核函数，可以被记做：

$$u_q(x,x')=\sum_j \lambda_j[\mathcal{A}_q e_j(x)]^T[\mathcal{A}_q e_j(x')] \tag{15}$$

并且，$\mathcal{A}_q e_j(x)=s_q(x)e_j(x)+\nabla_x e_j(x')$ 是作用在 e_j 上的 Stein 算子。那么，

$$\begin{aligned}
u_q(x,x') =&s_q(x)^T k(x,x')s_q(x')+s_q(x)^T\nabla_{x'}k(x,x')+s_q(x)^T\nabla_x k(x,x')+trace(\nabla_{x,x'}k(x,x'))\\
=&\sum_j \lambda_j\left[s_q(x)^T e_j(x)e_j(x')s_q(x')+s_q(x)^T e_j(x)\nabla_{x'}e_j(x')+s_q(x')^T\nabla_x e_j(x)e_j(x')+trace(\nabla_x e_j(x)\nabla_{x'}e_j(x'))\right.\\
=&\sum_j \lambda_j\left[s_q(x)e_j(x)+\nabla_x e_j(x)^T\right]\left[s_q(x')e_j(x')+\nabla_x e_j(x')\right]\\
=&\sum_j \lambda_j\left[\mathcal{A}_q e_j(x)\right]^T\left[\mathcal{A}_q e_j(x')\right]
\end{aligned} \tag{16}$$

所以，

$$\begin{aligned}
\mathbb{S}(p,q) =&\mathbb{E}_{x,x'}[u_q(x,x')]\\
=&\sum_i \lambda_i \mathbb{E}_x[\mathcal{A}_q e_j(x)]^T\mathbb{E}_{x'}[\mathcal{A}_q e_j(x')]\\
=&\sum_i \lambda_i ||\mathbb{E}_x[\mathcal{A}_q e_j(x)]||_2^2
\end{aligned} \tag{17}$$

推导到了这里，我们知道了 $\mathbb{S}(p,q)$ 的值，到达是个什么玩意了。这实际上就是 RKHS 空间中的一个内积。那么这个优化问题，是在 $k(x,x')$ 的 RKHS 中，找到一个 f。使 $\mathbb{E}_{x\sim p}[\mathcal{A}_p f(x)]$ 最大化，也就是在 RKHS 中的 β 和 f 的内积。我们令 $\beta(x')=\mathbb{E}_{x\sim p}[\mathcal{A}_p k_{x'}(x)]$，所以有：

$$\mathbb{S}(p,q) =\mathbb{E}_{x,x'\sim p}\left[(s_q(x)-s_p(x))^T k(x,x')(s_q(x')-s_p(x'))\right] \tag{18}$$

$$=\mathbb{E}_{x,x'\sim p}\left[(s_q(x)-s_p(x))^T <k(x,\cdot),k(x',\cdot)>_{\mathcal{H}}(s_q(x')-s_p(x'))\right] \tag{19}$$

$$=\sum_{l=1}^d \left\langle \mathbb{E}_x[(s_q^l(x)-s_p^l(x))k(x,\cdot)],\mathbb{E}_{x'}[(s_q^l(x)-s_p^l(x))k(x,\cdot)]\right\rangle \tag{20}$$

$$= \sum_{l=1}^{d} < \beta_l, \beta_l >_{\mathcal{H}} \tag{21}$$

$$= ||\beta||^2_{\mathcal{H}^d} \tag{22}$$

并且，$\beta(x') = \mathbb{E}_{x\sim p}[\mathcal{A}_q k_{x'}(x)] = \mathbb{E}_{x\sim p}[(s_q(x)k(x,x') + \nabla_x k(x,x'))] = \mathbb{E}_x[(s_q(x) - s_p(x))k(x,x')]$。所以，

$$\begin{aligned} < f, \beta >_{\mathcal{H}^d} &= \sum_{l=1}^{d} < f_l, \mathbb{E}_{x\sim p}[(s_q^l(x)k(x,\cdot) + \nabla_x k(x,\cdot)] >_{\mathcal{H}} \\ &= \sum_{l=1}^{d} \mathbb{E}_{x\sim p}[(s_q^l(x) < f_l, k(x,\cdot) >_{\mathcal{H}} + < f_l, \nabla_{x_l} k(x,\cdot) >_{\mathcal{H}})] \\ &= \sum_{l=1}^{d} \mathbb{E}_{x\sim p}[(s_q^l(x) f_l(x) + \nabla_{x_l} f_l(x))] \\ &= \mathbb{E}_{x\sim p}[trace(\mathcal{A}_q f(x))] \end{aligned} \tag{23}$$

其中，我们使用到了 $\nabla_x f(x) =< f(\cdot), \nabla_x k(x,\cdot) >_{\mathcal{H}}$。所以，$\mathbb{S}(p,q)$ 的最大化问题就是：

$$||\beta||_{\mathcal{H}^d} = \max_{f\in\mathcal{H}^d} \{< f, \beta >_{\mathcal{H}^d}, \qquad s.t.\ ||f||_{\mathcal{H}^d} \le 1\} \tag{24}$$

事实上也就是：

$$\mathbb{S}(p,q) = \max_{f\in\mathcal{H}^d} \left\{\mathbb{E}_{x\sim q}(trace(\mathcal{A}_p f(x)))^2, \qquad s.t.\ ||f||_{\mathcal{H}^d} \le 1\right\} \tag{25}$$

而最优化的结果，也就是我们得到的方向就是 $f = \frac{\beta}{||\beta||_{\mathcal{H}_d}}$，至于为什么？可以去看看文献. A kernel test of goodness of fit. ICML, 2016。了解机器学习的同学都知道，对于梯度来说，大小根本就不重要，重要的是方向。很多时候为了简化运算，我们都只得到方向就行了。

我觉得作者这里写得并不是很好，因为 $||\beta||^2_{\mathcal{H}^d}$ 和 $\sum_i \lambda_i ||\mathbb{E}_x[\mathcal{A}_q e_j(x)]||_2^2$ 是等价的，这里完全没有必要重新推导，可以直接等价过来。

下面我们来看 Stein 方法，第一个强大的地方，对于数据集 $\{D_k\}$ 是独立同分布的，先验为 $\bar{p}(z) = p_0(x)\prod_{k=1}^{N} p(D_k|x)$；而后验为 $p(x) = \bar{p}(z)/Z$，其中 $Z = \int \bar{p}(x)dx$。实际上这个归一化因子 Z 非常的难算。而在 Stein 方法中，我们只关注 score function，$s_p(x) = \nabla_x \log p(x)$，我们就会发现一求导，那个恶心的 Z 就不见了，所以Stein 方法可以有效的跳过求解归一化参数的过程，大大的简化了计算，也可以求解分布不明确归一化参数未知的问题。

3 Methods

在第二部分，我们算是比较完整的介绍了 Stein 方法的背景，实现的方法和思路。下一步，则是要看看如何在变分梯度下降中使用这个方法了。

我想先对这个概念捋一捋，我们要求一个 q^* 来近似等于 p。q 分布需要进行不断的迭代更新，每一次迭代更新就相当于一次平滑变换 T。而平滑变换的方法有很多，我们需要找到使目标函数下降最快的变换方向 (也就是梯度)。而 q 本身是一个函数，T 也是一个函数，这就是函数的函数，也就是泛函的问题。如何找到最优的变换 T，也就是泛函中的极值问题，我们可以使用变分法来求得。而 Stein 变分梯度下降中是在 RKHS 中计算的，并且使用了 particle 层次的方法。至于为什么这样做，我们下文会给出详细的解释。

3.1 利用平滑变换进行变分推断

首先，我们需要明确问题，也就是用一个简单的分布 $q^*(x)$ 来近似另一个目标分布 $p(x)$。而 $q^*(x)$ 的分布形式实在是太多了，如果我们漫无目的去找，根本就算不完。所以，我们需要定义一个函数族，他们都有着同样的形式，或者变换方法，我们称之为预定义分布集 $\mathcal{Q} = \{q(x)\}$，并采用最小化 q 和 p 之间的 KL 散度来进行求解。

$$q^* = argmin_{q\in\mathcal{Q}} \{KL(q||p) \equiv \mathbb{E}_q[\log q(x)] - \mathbb{E}_q[\log \bar{p}(x)] + \log Z\} \tag{26}$$

这里的 Z 并不用管，前面已经讲过了，就当成一个常数好了。那么，下一个问题，我们如何来选择这个函数集 $\mathcal{Q}$ 呢？这里的 $\mathcal{Q}$ 需要满足三个性质：

1. 准确度：广度足够，以至于可以近似的表示非常多种类的目标分布。

2. 计算可行性：也就是组成的分布比较简单，易于进行推断。

3. 可解决性：KL 散度最小化问题可以被有效的解决。

在此工作中，我们将 $\mathcal{Q}$ 看成一个由随机变量 1 对 1 变换得到的函数集，变换可以被我们定义为 $z = \mathbf{T}(x)$，$\mathbf{T} : \mathcal{X} \longrightarrow \mathcal{X}$，其中 x 是从 $q_0(x)$ 中采样获得的。实际上就相当于将 $q(x) \longrightarrow q(\mathbf{T}(x)) = q_{[\mathbf{T}]}(z)$。然后，我们需要参考一下坐标变换公式：

$$q_{[\mathbf{T}]}(z) = q(x) \cdot \left| det(\frac{\partial x}{\partial z}) \right| = q(\mathbf{T}^{-1}(z)) \cdot \left| det(\nabla_z \mathbf{T}^{-1}(z)) \right| \tag{27}$$

其中 $\mathbf{T}^{-1}$ 是 $\mathbf{T}$ 的逆变换，$\nabla_z(T)^{-1}$ 是 $\mathbf{T}^{-1}$ 的雅克比 (Jacobian) 变换。那么这个函数集 $\mathcal{Q}$ 理论上可以近似表示任何分布。在实际应用中，我们考虑 $\mathbf{T}$ 是一个参数化形式，并且我们需要优化其中的参数。所以我们需要寻找一个合适的函数族来平衡准确度和可行性这些东西，并且还需要一个有效的方法来计算 Jacobian 矩阵。好了，这就是作者说的，实现的困难。说了困难，那么紧接着就是自己的解决方法了。

作者提出了一种迭代式构建增量变换的方法可以有效的表现出 $\mathbf{T}$ 最快速的下降方向，不需要参数形式，也不要计算 Jacobian 矩阵，可以有效的表示梯度下降算法。(前面说了有问题，那么这里就要讲我们的方法可以解决嘛，逻辑关系很明确)

3.2 用 Stein 算子作为 KL 散度的梯度

既然，在上面的表述中，我们将目标函数都已经明确了，下一步就是想想如何使得 KL 散度最小化了。文章中考虑到了一个增量转换的形式，$\mathbf{T}(x) = x + \epsilon\phi(x)$，其中 $\phi(x)$ 就可以被我们看成函数的扰动，并且 ϵ 代表扰动的幅度。当 $|\epsilon|$ 的值非常的小时，$\mathbf{T}$ 的雅克比矩阵：$\nabla_x T(x) \approx \nabla_x x = I$，所以就保证了一个一对一的映射关系，这是由反函数理论得到的。在论文中给出了一个结论，这个结论有什么用呢？这个结论中给出了 $KL(q_{[\mathbf{T}]}||p)$ 的方向导数。

详细的描述即为：令 $\mathbf{T}(x) = x + \epsilon\phi(x)$ 并且 $q_{[\mathbf{T}]}(z)$ 表示当 $x \sim q(x)$ 时 $z = \mathbf{X}$，那么我们可以得到：

$$\nabla_\epsilon KL(q_{[\mathbf{T}]}||p)|_{\epsilon=0} = -\mathbb{E}_{x\sim q}[trace(\mathcal{A}_p\phi(x))] \tag{28}$$

并且，$\mathcal{A}_p\phi(x)) = \nabla_x \log p(x)\phi(x)^T + \nabla_x\phi(x)$ 是 Stein 算子。这个地方我刚开始看的非常的懵逼，这里省略了一个定义的说明也就是泛函导数，详细的描述请参考，https://en.wikipedia.org/wiki/Functional_derivative，同学相信我看完你就懂了。这里我们做一个简要的解释吧。

3.2.1 泛函导数 (Functional Derivative)

泛函的表达可以看成是 $F(f+\epsilon\eta)$，通过一系列的变化可以得到这么一个玩意：

$$F(f+\epsilon\eta)=\int_{x_0}^{x_1}F(x,f+\epsilon\eta,f'+\epsilon\eta')dx \tag{29}$$

积分完了以后，肯定只剩下了一个关于 ϵ 的函数，我们记为 $\phi(\epsilon)$，这个函数的性质就是当 $\epsilon=0$ 时取得函数的极值。也就是：

$$\frac{dF(f+\epsilon\eta)}{d\epsilon}|_{\epsilon=0} \tag{30}$$

3.2.2 方向导数的值

所以我们说，$\nabla_x KL(q_{[\mathbf{T}]}||p)|_{\epsilon=0}$ 得到的就是方向导数。

在理解了方向导数的由来之后，而 $\nabla_x KL(q_{[\mathbf{T}]}||p)|_{\epsilon=0}=-\mathbb{E}_{x\sim q}[trace(\mathcal{A}_p\phi(x))]$ 是怎么来的呢？下面将给出详细的推导。首先，就是使用分布变换公式：

$$q_{[\mathbf{T}^{-1}]}(x)=q(\mathbf{T}(x))\cdot|det(\nabla_x\mathbf{T}(x))| \tag{31}$$

而且，下面这个公式的导出也非常的自然：

$$KL(q_{[\mathbf{T}]}(x)||p)=KL(q||p_{[\mathbf{T}]}(x)) \tag{32}$$

所以，我们可以得到：

$$\begin{aligned}
\nabla_\epsilon KL(q_{[\mathbf{T}]}(x)||p)=&\nabla_\epsilon KL(q||p_{[\mathbf{T}^{-1}]}(x))\\
=&\nabla_\epsilon\left[\int q(x)\log\frac{q(x)}{p_{[\mathbf{T}^{-1}]}x}dx\right]\\
=&-\nabla_\epsilon\left[\int q(x)\log p_{[\mathbf{T}^{-1}]}xdx\right]\\
=&-\mathbb{E}_{x\sim q}[\nabla_\epsilon\log p_{[\mathbf{T}^{-1}]}(x)]\\
=&-\mathbb{E}_{x\sim q}[\nabla_\epsilon\log p(T(x))|\nabla_x T(x)|]\\
=&-\mathbb{E}_{x\sim q}[\nabla_\epsilon\log p(T(x))+\log|\nabla_x T(x)|]\\
=&-\mathbb{E}_{x\sim q}[\nabla_x\log p(T(x))^T\nabla_\epsilon T(x)+(\nabla_x T(x))^{-1}\nabla_\epsilon\nabla_x T(x)]
\end{aligned} \tag{33}$$

又因为，$T(x)=x+\epsilon\phi(x)$，$\epsilon=0$。所以可以得到：

$$T(x)=x,\quad \nabla_\epsilon T(x)=\phi(x),\quad \nabla_x T(x)=I,\quad \nabla_\epsilon\nabla_x T(x)=\nabla_x\phi(x) \tag{34}$$

所以，

$$\nabla_\epsilon KL(q_{[\mathbf{T}]}(x)||p)=-\mathbb{E}_{x\sim q}[\nabla_x\log p(x)^T\phi(x)+(I)^{-1}\nabla_x\phi(x)] \tag{35}$$

$$\nabla_\epsilon KL(q_{[\mathbf{T}]}(x)||p)=-\mathbb{E}_{x\sim q}[trace(\mathcal{A}_p\phi(x))] \tag{36}$$

实际上这里的 ϕ 就是变化的方法，我们需要寻找的是一个函数，所以变化的不像传统的一维变换那样是一个标量，我们变换的是一个方向。而变换函数的集合，我们可以看成是一个“球”，也就

是所有方向的集合：$\mathcal{B} = \{\phi \in \mathcal{H}^d : ||\phi||^2_{\mathcal{H}^d} \leq \mathbb{S}(p,q)\}$。而我们需要找到令 $\nabla_\epsilon KL(q_{[\mathbf{T}]}(x)||p) = -\mathbb{E}_{x\sim q}[trace(\mathcal{A}_p\phi(x))]$ 最大的这个值对应的 $\phi(x)$，记为：$\phi^*_{q,p}$。这个值我们前面说过了，也就是：

$$\phi^*_{q,p}(\cdot) = \mathbb{E}_{x\sim q}\left[k(x,\cdot)\nabla_x \log p(x) + \nabla_x k(x,\cdot)\right] \tag{37}$$

实际上也就等价于，惊奇的发现了这个 KL 散度的梯度方向就是 Stein 核差异：

$$\nabla_\epsilon KL(q_{[\mathbf{T}]}(x)||p) = -\mathbb{S}(p,q) \tag{38}$$

这个结果，给了我们一个启发。我们如何从一个初始的分布 q_0 通过有限次变换来逼近目标分布 p，最开始的时候，最优的变换分向为 $T^*_0 = x + \epsilon_0 \cdot \phi^*_{q_0,p}(x)$。因为，$\phi^*_{q,p}(\cdot) = -\mathbb{S}(q,p)$。所以，实际上，KL 散度减少的是 $\epsilon_0 \cdot \mathbb{S}(q_0,p)$。经过这一步变换，我们可以得到一个新的分布 $q_1(x) = q_{0[T_0]}(x)$。第二步的转换和第一步其实也是一样的。第二步的最优变换方向为 $T^*_1 = x + \epsilon_1 \cdot \phi^*_{q_1,p}(x)$。KL 散度减少的是 $\epsilon_1 \cdot \mathbb{S}(q_1,p)$。这一步变换之后，我们得到的分布为 $q_2(x) = q_{1[T_1]}(x)$。重复这个过程，从初始的分布 q_0 到目标分布 p 之间，将会构建一系列的分布 $\{q_l\}_{l=1}^n$，通过：

$$q_{l+1} = q_{l[T^*_l]} \qquad\qquad T^*_l(x) = x + \epsilon_l \cdot \phi^*_{q_l,p}(x) \tag{39}$$

实际上，分布最终会收敛到目标分布 p，只要 ϵ_l 足够的小。到了最后会有 $\phi^*_{p,q_\infty} \equiv 0$。所以，有且仅有当 $\phi^*_{p,q_\infty} \equiv 0$ 时，有 $p = q_\infty$。

3.2.3 泛函梯度 (Functional Gradient)

在这一小节中，我们将要推导一下，为什么式 (37) 是可再生核希尔伯特空间的泛函导数。这里将和 2.4 小节完全串起来，也就是解释了，为什么 $\phi^*_{p,q}$ 是最大核差异方向了。对于任意的泛函 $F[f]$，$f \in \mathcal{H}^d$。它的泛函梯度 $\nabla_f F[f]$ 是 $\mathcal{H}^d$ 中的一个函数，对于任何 $g \in \mathcal{H}^d$ 并且 $\epsilon \in \mathbb{R}$：

$$F[f + \epsilon g(x)] = F[f] + \epsilon < \nabla_f F[f], g >_{\mathcal{H}^d} + O(\epsilon^2) \tag{40}$$

令 $T(x) = x + f(x)$，$f \in \mathcal{H}^d$，并且，$q_{[T]}$ 代表 $z = T(x)$ 变换后的分布，并且 $x \sim q$，

$$\nabla_f KL(q_{[T]}||p)|_{f=0} = -\phi^*_{q,p}(x) \tag{41}$$

RKHS 的范数为 $||\phi^*_{p,q}||^2_{\mathcal{H}^d} = \mathbb{S}(q,p)$。下面我们来证明一下：
令 $\mathcal{H}^d = \mathcal{H} \times \cdots \times \mathcal{H}$ 是一个向量值。$F[f]$ 是 f 的函数，$F[\cdot]$ 的梯度 $\nabla_f F[f]$，在 RKHS 中满足，

$$F[f + \epsilon g(x)] = F[f] + \epsilon < \nabla_f F[f], g >_{\mathcal{H}^d} + O(\epsilon^2) \tag{42}$$

我们很容易可以知道 $F[f] = KL(q_{[x+f(x)]}||p) = KL(q||p_{(x+f(x))^{-1}})$。并且，这里有一个公式我们需要经常使用，这里我们再次强调一下，$p_{[T^{-1}]}(x) = p(T(x)) \cdot |\nabla_x T(x)|$。

$$\begin{aligned}
F(f+\epsilon g) =& KL(q||p_{[(x+f(x)+\epsilon g(x))^{-1}]}) \\
=& \int q(x) \log \frac{q(x)}{(p(x)+f(x)+\epsilon g(x))^{-1}} dx \\
=& \mathbb{E}_q[\log q(x) - \log(p(x)+f(x)+\epsilon g(x))^{-1}] \\
=& \mathbb{E}_q[\log q(x) - \log(p(x)+f(x)+\epsilon g(x)) - \log det(I + \nabla_x f(x) + \epsilon \nabla_x g(x))]
\end{aligned} \tag{43}$$

所以，我们可以得到：

$$F(f+\epsilon g)-F(f)=-\triangle_1-\triangle_2 \tag{44}$$

$$\triangle_1=\mathbb{E}_q[\log p(x+f(x)+\epsilon g(x))]-\mathbb{E}_q[\log p(x+f(x))] \tag{45}$$

$$\triangle_2=\mathbb{E}_q[\log det(I+\nabla_x f(x)+\epsilon\nabla_x g(x))]-\mathbb{E}_q[\log det(I+\nabla_x f(x))] \tag{46}$$

对于第一项：

$$\begin{aligned}\triangle_1=&\mathbb{E}_q[\log p(x+f(x)+\epsilon g(x))]-\mathbb{E}_q[\log p(x+f(x))]\\=&\epsilon\mathbb{E}_q[\nabla_x\log p(x+f(x))\cdot g(x)]+O(\epsilon^2)\\=&\epsilon\mathbb{E}_q[\nabla_x\log p(x+f(x))\cdot\langle k(x,\cdot),g\rangle_{\mathcal{H}^d}]+O(\epsilon^2)\\=&\epsilon\langle\mathbb{E}_q[\nabla_x\log p(x+f(x))\cdot k(x,\cdot)],g\rangle_{\mathcal{H}^d}+O(\epsilon^2)\end{aligned} \tag{47}$$

并且，

$$\begin{aligned}\triangle_2=&\mathbb{E}_q[\log\det(I+\nabla_x f(x)+\epsilon\nabla_x g(x))]-\mathbb{E}_q[\log\det(I+\nabla_x f(x))]\\=&\mathbb{E}_q[\log\det(I+\nabla_x f(x))+\epsilon\frac{|I+\nabla_x f(x)|}{|I+\nabla_x f(x)|}\cdot(I+\nabla_x f(x))^{-1}\cdot\nabla_x g(x)-\log\det(I+\nabla_x f(x))]\\=&\epsilon\mathbb{E}[\mathrm{trace}(I+\nabla_x f(x))^{-1}\nabla_x g(x))]\\=&\epsilon\mathbb{E}[\mathrm{trace}(I+\nabla_x f(x))^{-1}\cdot\langle\nabla_x k(x,\cdot),g)\rangle_{\mathcal{H}^d}]+O(\epsilon^2)\\=&\langle\epsilon\mathbb{E}[\mathrm{trace}(I+\nabla_x f(x))^{-1}\cdot\nabla_x k(x,\cdot)],g)\rangle_{\mathcal{H}^d}+O(\epsilon^2)\end{aligned} \tag{48}$$

所以，

$$F(f+\epsilon g)-F(f)=\epsilon<\nabla_f F[f],g>_{\mathcal{H}^d}+O(\epsilon^2) \tag{49}$$

其中，

$$\nabla_f F[f]=-\mathbb{E}_q[\nabla_x\log p(x+f(x))+trace((I+\nabla_x f(x)))^{-1}\cdot k(x,\cdot)\cdot\nabla_x k(x,\cdot)] \tag{50}$$

由于 $f=0$，所以就可以得到最后的结果：

$$\nabla_f F[f]=-\mathbb{E}_q[\nabla_x\log p(x)\cdot k(x,\cdot)+\nabla_x k(x,\cdot)] \tag{51}$$

在这里的 $T(x)=x$ 中为什么令 $f=0$ 呢？因为有非常多的好处，如果 $f\neq 0$，$|\nabla_x T(x)^{-1}|$ 的计算将变得非常的复杂。

3.3 Stein 变分梯度下降

首先，我先总结一下，Bayesian Inference via Variational Gradient Descent 为：
输入：一个目标分布函数 $p(x)$，一系列初始粒子 $\{x_i^0\}_{i=1}^n$。
输出：一系列近似目标分布的粒子 $\{x_i\}_{i=1}^n$。
对于每次迭代 l：

$$x_i^{l+1}\longleftarrow x_i^l+\epsilon_l\hat{\phi}^*(x_i^l)\qquad\hat{\phi}^*(x)=\frac{1}{n}\sum_{j=1}^{n}[k(x_j^l,x)\nabla_{x_j^l}\log p(x_j^l)+\nabla_{x_j^l}k(x_j^l,x)] \tag{52}$$

在这里使用了粒子的迭代程序，可以用来近似计算式 (37) 中的 $\phi^*_{q,p}(x)$。我们首先从原始的分布 q_0 中提取一系列粒子 $\{x_i^0\}_{i=1}^n$。然后，进行不断的迭代更新，经过 l 次迭代，最终可以得到最后的粒子集合 $\{x_i^l\}_{i=1}^n$。所以大家有没有注意到，我们的算法不依赖与初始的粒子集合 $\{x_i\}_{i=1}^n$，也就是说任意的初始化粒子，我们都可以得到最终的任意的复杂的分布。

在这个平均场理论的采样中，有坚实的理论基础做支撑，也就是霍夫丁不等式。

$$\sum_{i=1}^{n} \frac{h(x_i^l)}{n} - \mathbb{E}_{q_l}[h(x)] = O(\frac{1}{\sqrt{n}}) \tag{53}$$

这是机器学习中的基础理论，揭示了采样计算的均值和真实均值之间的联系，也就是说，采样的粒子数量越多，真实的均值和采样获得的均值之间的差就越小。

下面我们 $\hat{\phi}^*(x)$ 进一步进行分析,它有着两个完全不同的作用。第一个部分是 $k(x_j^l, x)\nabla_{x_j^l} \log p(x_j^l)$，这个函数使得粒子向高概率的方向移动，而核函数就是相当于一个权值的作用。而第二项 $\nabla_{x_j^l} k(x_j^l, x)$ 就刚好相反，就以 RBF 核函数为例，第二项等于 $\sum_j \frac{2}{h}(x - x_j)k(x_j, x)$，也就让 x 远离 x_j 临近的点，这一项是防止陷入局部最优的，也就是相当于一个正则项。两者互补，类似强化学习中的开发和探索。还有一个有趣的地方，当 $n = 1$ 时，就相当于随机梯度下降了，我不知道大家可不可以 get 到这个点，其实就是，和粒子的数量无关。粒子数量减少，当然优化的时间不步数就变得更长了。而且，算法中给出了确定性的下降方法，远远的优于传统的 MCMC 算法，MCMC 算法中使用随机性的方法来增加粒子的多样性，在 Stein 变分中根本就不需要。

3.4 提高计算效率的方法

马上就有了一个新的问题，这个算法中最主要的计算瓶颈在于，对于所有的粒子 $\{x_i\}_{i=1}^n$ 计算 $\nabla_x \log p(x)$ 的梯度，特别是 $p(x) \propto p_0(x)\prod_{k=1}^N p(D_k|x)$，随着数据量的增加，这会变得非常的难算。所以，文中提出了一种下采样的方法，先采一个子集 $\Omega \subset \{1, \cdots, N\}$。那么我们可以来估计计算梯度，用局部做整体的计算：

$$\nabla_x \log p(x) \approx \nabla_x \log p_0(x) + \nabla_x \frac{N}{|\Omega|} \sum_{k\in\Omega} \log p(D_k|x) \tag{54}$$

总体来说下采样还是个挺香的方法，这种计算 $\sum_{i=1}^n$ 的东西，都可以使用下采样的方法。

3.5 实验细节

这里简单的提几个实验细节吧，核函数我们使用的是 RBF 核函数：$k(x, x') = exp\left(-\frac{1}{h}||x - x'||_2^2\right)$，并且我们令 $h = \frac{med^2}{\log n}$，其中 med 是中位数。其他的实验细节，请朋友们自己去论文中阅读，或者看作者的代码吧，在论文的 5 Experiments 部分都有详细的介绍。

4 Conclusion and Discussion

说实话我理解 Stein 变分花了很长的时间，才摸索清楚。这里面需要大量的数学和机器学习的基础，对于我来说难度非常的大。同时也不得不说，这个工作非常的 soild。我读到的论文的几大优点，这里做一个简单的小结：

1. Stein 方法关注的主要是得分函数 $\nabla_x \log p(x)$。可以有效的跳过求解归一化参数的过程，大大的简化了计算，也可以求解分布不明确归一化参数未知的问题。

2.Stein 变分在泛函空间中计算，$\phi(x)$ 属于可再生核希尔伯特空间，求得了泛函的导数为梯度方向，有效的避免了求解 $\nabla_x T^{-1}(x)$ 这个恶心的东西。

3. 在计算分布之间的距离的时候，Stein 算子可以成功的是未知分布部分的值为零，从而消除未知分布的无法计算的问题。非常的漂亮！

4. 使用 Stein 变分的方法，成功的退导出了，梯度就是核差异的方向，核差异就是泛函的导数的方向。所以，就给出了梯度方向的解析解，这是一种确定性的方向，可以有效的增加优化的速度和效率。不像随机性方法那样，需要通过随机性来增加粒子的多样性。

5. 还有一个有意思的地方，也就是我们最终求得的梯度的解析式，似乎本身就带有正则化。第一项和第二项之间，相互制衡，可以同时兼顾探索也开发。

6. 算法不依赖与初始的粒子集合 $\{x_i\}_{i=1}^n$，也就是说任意的初始化粒子，我们都可以得到最终的任意的复杂的分布。

但是，同时缺点也很明显，那么就是，这个计算实在是太复杂了，以至于小编的机器会跑嗝屁。裴神提出了一种优化的方法，跳过了对每个参数来做优化。而是，直接求得了梯度的值，然后通过链式求导的方法来获得梯度方向来避免每一个粒子进行复制运算。也就是将：

$$\frac{\partial \mathcal{J}}{\partial \theta} = \frac{\partial \mathcal{J}}{\partial q} \cdot \frac{\partial q}{\partial \theta} \tag{55}$$

第一项是，损失函数对分布 q 求导，就是梯度方向 $\phi(q)$，第二项为，分布对 θ 求导也就是为了获得梯度的方向。所以，我可以直接写成 $\phi(q) \cdot q$，也就是直接在后面乘上一个 q。ϕ 直接求出了，再利用 $\nabla_\theta q$ 来获得方向，这样就会简单很多。这里感觉没懂的，欢迎和作者进行详细讨论，同时调参也有好方法，这里不做过多的描述了!!!

5 My Reflections

变分法博大精深，什么地方都可以用，这个方法太好了。也就是解决函数的函数的问题。因为，我们经常需要做一个转换，这个转换就是一个函数。而变分法则可以帮助我们求得这个最优的函数，也就是泛函导数。进一步深入理解变分法的精髓，还需要努力的学习。我非常喜欢一句话 “一切皆可变分！”。