Contents

An Idiot's Guide to Stein Points

These are my (ie the idiot) introduction to Stein Points, and the two associated computational problems I will investigate. These will be fleshed out over time as I come to understand more theory, and will include a number of examples that I found helpful on the way. Essentially everything in here is taken from Chen Et Al (2018).

Stein Point Concepts

Motivation

A common statistical problem is to approximate a probability distribution $P(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^d$ with a sequence of points $\{\mathbf{x}_i\}_{i=1}^n$. Such a sequence of points is an *empirical distribution* and is useful because it permits efficient calculation of various quantities (moments, quantiles etc). In addition they have well known and usefull theoretical properties. The critical goal is to produce a sequence which achieves *convergence* in the sense that

$$\frac{1}{n}\sum_{i=1}^{n} h(\mathbf{x}_i) \;\to \int h dP \tag{1}$$

when $n \to \infty$, where we are subject to various technical conditions in the above.

This projects concerns a particular sequence $\{\mathbf{x}_i\}_{i=1}^n$ of points called **Stein Points**, which are chosen carefully to provide a *best approximation* to $P(\mathbf{x})$, where we will define exactly what we mean by *best* in a later section. There are many techniques for generating convergent sequences of points from a given distribution, but most require all normalisation constants to be known. This requirement is commonly violated in practice, where we know the *functional form* of the distribution but may not be able to compute multiplicative constants. MCMC techniques can circumvent this problem by defining transition probabilities as ratios, so that these unknown constants cancel.
Stein Points also do not require multiplicative constants to be known, so exist in a smaller set of plausible methods for this very common problem.

Definitions

A *discrepency* quantifies how well the points $\{\mathbf{x}_i\}$ cover the domain of the random variable $\mathbf{x}$ with respect to the distribution $p(\mathbf{x})$. The set of points which minimise a particular type of discrepency (a *Kernel Stein Discrepency*) with respect to the target $p(\mathbf{x})$ are referred to as *Stein Points*. The Stein Point methodology exists in the more general framework of *Reproducing Kernel Hilbert Spaces* (RKHS), which is a popular framework because analytic formulas for discrepencies are available. However, these general results suffer from the usual challenge that the target distribution's normalisation constant is required. It can be shows that particular choices of *Reproducing Kernels* (those chosen from a *Stein Set*) both simplify the maths in the more general RHKS and do not require knowledge of the normalisation constants. Such discrepencies are called *Kernel Stein Discrepencies* (KSD), and involve particular kernels referred to as *Stein Reproducing Kernels*. In particular the *Kernel Stein Discrepency* (KSD) is defined

$$\mathcal{D} = \sqrt{\frac{1}{n}\sum_i \sum_j k_0(\mathbf{x}_i, \mathbf{y}_j)} \tag{2}$$

where $k_0(\mathbf{x}, \mathbf{y})$ is the *Stein Reproducing Kernel* and is defined in a subsequent section.

Process

The process of approximating a target $p(\mathbf{x})$ using Stein Points proceeds by

1. Choosing a particular kernel $k(\mathbf{x}, \mathbf{y})$, which measures distances between points. This choice is subject to various technical conditions.
2. Choose a *Stein Operator* τ, which should be chosen cleverly in conjunction with the kernel above.
3. Compute the *Stein Reproducing Kernel* (SRK) $k_0(\mathbf{x}, \mathbf{y})$ by solving the integral equation $\int \tau[k] dP = 0$
4. Use the *Stein Reproducing Kernel* (SRK) to compute the *Kernel Stein Discrepency*

$$\boxed{\mathcal{D} = \sqrt{\frac{1}{n}\sum_i \sum_j k_0(\mathbf{x}_i, \mathbf{y}_j)}}$$

5. Choose an optimisation strategy to minimise the KSD above

Note that *Stein Operators* τ are chosen to

- Produce discrepencies that do not require normalisation constants
- Simplify formulas for discrepencies that are required when using the more general RKHS framework
- Combine with the choice of kernal $k(\mathbf{x}, \mathbf{y})$ to guarantee that as the KSD $\rightarrow 0$ the empirical distribution produced by our point sequence is convergent to the target $p(\mathbf{x})$.

After writing $k = k(\mathbf{x}, \mathbf{y})$ to save notation, the particular choice of τ considered here is the *Langevin Stein Operator*

$$\tau[k] = \nabla \cdot (pk)/p \tag{3}$$

which generates a Stein Reproducing Kernel (SRK)

$$\boxed{k_0 = \nabla_{\mathbf{x}} \cdot \nabla_{\mathbf{y}} k + \nabla_{\mathbf{x}} k \cdot \nabla_{\mathbf{y}} \log p(\mathbf{y}) + \nabla_{\mathbf{y}} k \cdot \nabla_{\mathbf{x}} \log p(\mathbf{x}) + k \nabla_{\mathbf{x}} \log p(\mathbf{x}) \cdot \nabla_{\mathbf{y}} \log p(\mathbf{y})} \tag{4}$$

The `div` $\cdot$ `grad` structure of the SRK results in a computation where partial derivatives are evaluated at each of the d components of $\mathbf{x}$ before being summed, that is, $k_0(\mathbf{x}, \mathbf{y})$ can be written

$$\boxed{k_0(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} \left(\frac{\partial^2 k}{\partial x_i \partial y_i} + \frac{\partial k}{\partial x_i}\frac{\partial g}{\partial y_i} + \frac{\partial k}{\partial y_i}\frac{\partial g}{\partial x_i} + k(\mathbf{x}, \mathbf{y})\frac{\partial g}{\partial x_i}\frac{\partial g}{\partial y_i} \right)} \tag{5}$$

where I have substituted $g = \log p$ to again save notation.

Examples Part 1: Multivariate Normal Distribution with IMQ Kernel

In this section I will work through a few simple examples to clarify my understanding of the problem. These will focus on the IMQ kernel and the target distribution will be the multivariate normal distribution. The multivariate normal distribution with mean μ and covariance Σ is

$$p(\mathbf{z}) = \frac{1}{\sqrt{\det\left(2\pi\Sigma\right)}} \exp\left[-\frac{1}{2}\left(\mathbf{z}-\mu\right)^T \Sigma^{-1} \left(\mathbf{z}-\mu\right)\right]$$

with corresponding log target $g(\mathbf{z})$

$$g(\mathbf{z}) = -\frac{1}{2}\log\left(\det\left(2\pi\Sigma\right)\right) - \frac{1}{2}\left(\mathbf{z}-\mu\right)^T \Sigma^{-1} \left(\mathbf{z}-\mu\right)$$

where $\mathbf{z} \in \mathbb{R}^d$ and of course the additive constant can be ignored for our purposes. The IMQ kernel is

$$k(\mathbf{x}, \mathbf{y}) = \left(\alpha + \|\mathbf{x}-\mathbf{y}\|^2\right)^{\beta}$$

with $\alpha > 0$ and $-1 < \beta < 0$.

Example 1.1: Univariate Normal Distribution

In this example I take $d = 1$ and investigate:

- What do the KSD and SRK formulas actually look like?
- What kind of objective functions result from the Greedy Minimisation strategy?

With $d = 1$ the objective function for the greedy minimisation strategy are easy to visualise. This is the univariate normal target $p(x)$. This will be the only case where the bold font will not be used for the random variable. The SRK is

$$k_0(x, y) = \frac{\partial^2 k}{\partial x \partial y} + \frac{\partial k}{\partial x}\frac{\partial g}{\partial y} + \frac{\partial k}{\partial y}\frac{\partial g}{\partial x} + k(x, y)\frac{\partial g}{\partial x}\frac{\partial g}{\partial y} \tag{6}$$

The log gradient for this choice of p is

$$\frac{\partial g}{\partial z} = -(z - \mu)/\Sigma \tag{7}$$

where both $\mu, \Sigma \in \mathbb{R}$, and the kernel derivatives are

$$\begin{aligned}
\frac{\partial k}{\partial x} =& 2\beta(x - y)\left(\alpha + \|x - y\|^2\right)^{\beta-1} \\
\frac{\partial k}{\partial y} =& -2\beta(x - y)\left(\alpha + \|x - y\|^2\right)^{\beta-1} \\
\frac{\partial^2 k}{\partial x \partial y} =& -2\beta\left(\alpha + \|x - y\|^2\right)^{\beta-2}\left[2(\beta - 1)(x - y)^2 + \left(\alpha + \|x - y\|^2\right)\right]
\end{aligned}$$

Using the greedy optimisation method (Section 3.1) we set the initial Stein Point to be $x_1 = \mu$, the distribution mode. Eventually this will require a global optimisation call but in this simple example lets just assign it. For each subsequent x_n for $n > 1$ take x_n to be the value which minimises the objective

$$\operatorname{argmin}_z \left\{\frac{1}{2}k_0(z, z) + \sum_{j=1}^{n-1} k_0(x_j, z)\right\} \tag{8}$$

Since we are in one dimension we can do an exhaustive grid search, and plot the objective function at each iteration to get a feel for the minimisation surface. In the plot below each curve plots the greedy objective for point x_j where $j = 2, 3, \dots, n$. No objective is plotted for the initial point but the point itself is plotted in red. Subsequent Stein Points are plotted in black. This example would appear to be the *simplest possible example* of the computation of Stein Points and yet the scale of the optimisation problem is already apparent.

To do:

1. Use the scipy global minimizer -> does it work here?
2. Compute say n=100 points, and compute an empirical distribution. How does it compare to a sample of 100 points sampled from the true distribution?

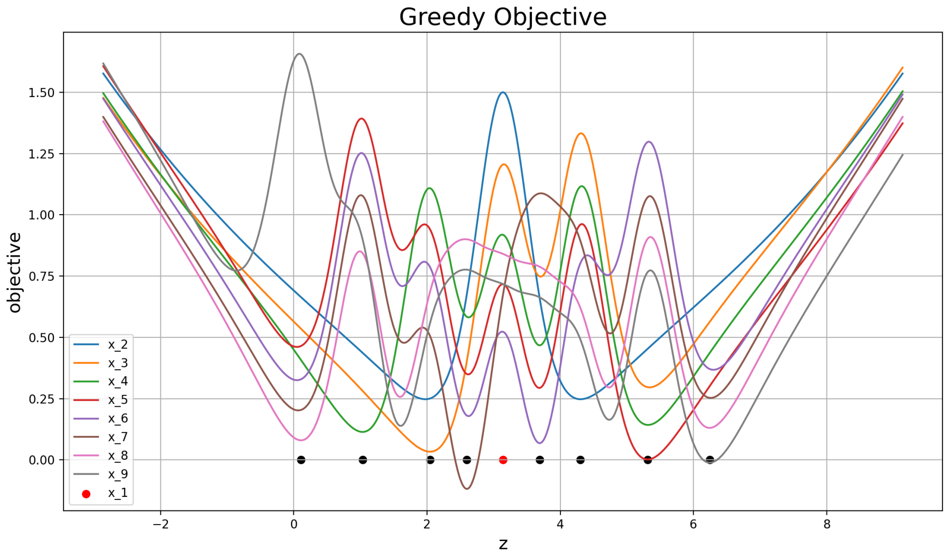

Figure 1: Greedy Objective for a Univariate Normal Distribution

Next Section

The three derivatives of $k(\mathbf{x}, \mathbf{y})$ are problem independent (ie do not depend on the target required) and are easy to calculate analytically

$$\begin{aligned}\frac{\partial k}{\partial x_i} =& 2\beta(x_i - y_i)\left(\alpha + \|\mathbf{x} - \mathbf{y}\|^2\right)^{\beta-1}\\ \frac{\partial k}{\partial y_i} =& -2\beta(x_i - y_i)\left(\alpha + \|\mathbf{x} - \mathbf{y}\|^2\right)^{\beta-1}\\ \frac{\partial^2 k}{\partial x_i \partial y_i} =& -2\beta\left(\alpha + \|\mathbf{x} - \mathbf{y}\|^2\right)^{\beta-2}\left[2(\beta-1)(x_i - y_i)^2 + \left(\alpha + \|\mathbf{x} - \mathbf{y}\|^2\right)\right]\end{aligned}$$