A modified relative entropy loss for learning the Smoluchowski Diffusion force function

Andrew Gracyk

August 18, 2021

Abstract

We provide a method for recovering the external force vector field of the Smoluchowski Diffusion equation using neural networks. We produce numerical solution formulations. We modify relative entropy loss, producing a means of deducing the discretized external force. This write-up is meant to be a discussion of an idea.

1 Introduction

This short write-up is concerned with a method of deducing the external force vector field function $\boldsymbol{\Gamma}(\boldsymbol{r}) = (\Gamma_1(\boldsymbol{r}), \Gamma_2(\boldsymbol{r}), \Gamma_3(\boldsymbol{r}))^T$ as part of the Smoluchowski Diffusion equation, where such a force acts upon a Brownian particle in $\mathbb{R}^3$ space. We may employ techniques in machine learning to do so. If we are given data of solutions to the Smoluchowski Diffusion equation, which are probability density functions, it is our aim to use this information to extrapolate the external force vector field. We may utilize our loss function in doing so, where we may minimize some notion of a difference between probability distributions that satisfy the Smoluchowski Diffusion equation, notably one that is an approximate distribution computed with the neural network prediction and one that is the true data that conforms to the equation.

The data of solutions that we will be concerned with will be the discretized approximations to the distributions that satisfy the Smoluchowski Diffusion equation, discretized over some mesh $\Omega \subseteq \mathbb{R}^3$. These numerical solutions can be found by a variety of numerical methods, especially methods for divergence and gradient approximations. It is overwhelming challenging to deduce the true distribution solutions that the particle follows, making numerical computations a viable choice.

Kullback-Leibler divergence is a mathematical framework that is frequently used in the formulation of neural network loss functions when comparing probability distributions. This loss possesses desirable properties and is the most apt in density function relations. For such reasons, this loss will be what we turn to when comparing numerical solutions to our Smoluchowski Diffusion equation; however, it may be noted that problems can arise when using this loss function. This loss can force solutions to converge to the trivial solution, the solution may not converge to a probability distribution upon training, or undefined or infinite values may appear. To rectify possible issues, we develop a custom loss function that derives from Kullback-Leibler divergence.

The Smoluchowski Diffusion equation, the key equation we will be considering, is given by

$$\frac{\partial \rho(\boldsymbol{r}, t|\boldsymbol{r}_0, t_0)}{\partial t} = \nabla \cdot \left[D(\boldsymbol{r})(\nabla - \beta \boldsymbol{\Gamma}(\boldsymbol{r}))\rho(\boldsymbol{r}, t|\boldsymbol{r}_0, t_0) \right] \tag{1}$$

whose solution is a probability distribution $\rho : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}$ that the particle follows at time t. $D(\boldsymbol{r})$ is a spatially-dependent diffusion function, which we will hold constant throughout our investigation. β is an additional constant parameter. $\boldsymbol{r}_0$ is an initial position of the particle situated at time t_0, where we will generally impose the starting time $t_0 = 0$. $\boldsymbol{\Gamma}(\boldsymbol{r})$ is the function in which it is our overarching goal to find. This function can be set when generating data that coheres to this equation, but it can be left unknown in our attempts to deduce it with our machine learning methods. Note that equation (1) can be rewritten

$$\frac{\partial \rho(\boldsymbol{r}, t|\boldsymbol{r}_0, t_0)}{\partial t} = D\Delta\rho(\boldsymbol{r}, t|\boldsymbol{r}_0, t_0) - D\beta\nabla \cdot \boldsymbol{\Gamma}(r)\rho(\boldsymbol{r}, t|\boldsymbol{r}_0, t_0) \tag{2}$$

which is an easier form in finding numerical solutions and in dealing with a loss function.

2 Numerical solutions

We can discretize the time derivative using the method $(\Phi^{i+1} - \Phi^i)/\Delta t$ and iterate this to arrive at the equation

$$\Phi^{T/\Delta t} = \Phi^0 + \sum_{i=0}^{T/\Delta t - 1} \Delta t \Big[\nabla \cdot D(\nabla - \beta\boldsymbol{\Gamma}(\boldsymbol{x}))\Phi^i\Big], \tag{3}$$

giving us discrete approximations to solutions of the Smoluchowski Diffusion equation, where

$$\Phi^i \approx \rho(\boldsymbol{r}, t_i|\boldsymbol{r}_0, t_0)\Big|_{r\in\Omega}. \tag{4}$$

We construct a mesh $\Omega \subseteq \mathbb{R}^3$ and find approximations Φ^i along our mesh at $\boldsymbol{x} \in \Omega$, where these approximations can be computed with numerical schemes for the Laplacian and divergence. $\boldsymbol{\Gamma}(\boldsymbol{x})$ is the external force vector field evaluated along the mesh. The differential operators applied to our discrete solutions denote estimations of the same operators applied to the continuous versions.

We can similarly construct approximate distribution solutions using a neural network in place of external force vector field $\boldsymbol{\Gamma}(\boldsymbol{r})$. This gives us a new distribution solution that offers the ability for comparison with estimations Φ, both of which we may input in a loss function. Let $\tilde{\boldsymbol{\Gamma}}$ denote our discrete neural network solution to $\boldsymbol{\Gamma}$. We can designate our neural network to have three outputs, $\tilde{\boldsymbol{\Gamma}}(\boldsymbol{r}) = (\tilde{\Gamma}_1(\boldsymbol{r}), \tilde{\Gamma}_2(\boldsymbol{r}), \tilde{\Gamma}_3(\boldsymbol{r}))$, mapping spatial position within Ω to this predicted force function. Let

$$\overline{\Psi}(\boldsymbol{r}, t + \Delta t; \overline{\tilde{\boldsymbol{\Gamma}}}) = \rho(\boldsymbol{r}, t|\boldsymbol{r}_0, t_0) + \Delta t\Big[\nabla \cdot D(\nabla - \beta\overline{\tilde{\boldsymbol{\Gamma}}}(\boldsymbol{r}))\rho(\boldsymbol{r}, t|\boldsymbol{r}_0, t_0)\Big] \tag{5}$$

denote an approximate Smoluchowski Diffusion solution with neural network input instead of true force function input. The neural network approximation is extended to be continuous, which we denote $\overline{\tilde{\boldsymbol{\Gamma}}}$, where $\overline{\Psi}$ is also continuous. Ψ^i will denote the analogous discrete solution at time t_i, computed in a similar manner as Φ^i, but with neural network approximation $\tilde{\boldsymbol{\Gamma}}$.

3 Developing our loss

We develop a custom loss function that derives from Kullback-Leibler divergence, one that remedies problems that may arise. Consider

$$\int_{\mathbb{R}^3} \overline{\Psi}(\boldsymbol{r}, T; \tilde{\overline{\boldsymbol{\Gamma}}}) \cdot \log\Big(\frac{\overline{\Psi}(\boldsymbol{r}, T; \tilde{\overline{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0)}\Big) d\boldsymbol{r} = \tag{6}$$

Applying our time discretization for $\overline{\Psi}(\boldsymbol{r}, T; \tilde{\overline{\boldsymbol{\Gamma}}})$,

$$= \int_{\mathbb{R}^3} \Big(\rho(\boldsymbol{r}, T-\Delta t|\boldsymbol{r}_0, t_0) + \Delta t\Big[\nabla \cdot D(\nabla - \beta\tilde{\overline{\boldsymbol{\Gamma}}}(\boldsymbol{r}))\rho(\boldsymbol{r}, T-\Delta t|\boldsymbol{r}_0, t_0)\Big]\Big) \cdot \log\Big(\frac{\overline{\Psi}(\boldsymbol{r}, T; \tilde{\overline{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0)}\Big) d\boldsymbol{r}. \tag{7}$$

Rewriting terms within the integral,

$$= \int_{\mathbb{R}^3} \Big(\rho(\boldsymbol{r}, T-\Delta t|\boldsymbol{r}_0, t_0) + \Delta t\Big[D\Delta\rho(\boldsymbol{r}, T-\Delta t|\boldsymbol{r}_0, t_0) \tag{8}$$

$$- \nabla \cdot \tilde{\overline{\boldsymbol{\Gamma}}}(\boldsymbol{r})\rho(\boldsymbol{r}, T-\Delta t|\boldsymbol{r}_0, t_0)\Big]\Big) \cdot \log\Big(\frac{\overline{\Psi}(\boldsymbol{r}, T; \tilde{\overline{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0)}\Big) d\boldsymbol{r}. \tag{9}$$

We assume sufficiently nice properties of our functions. Separating the integral,

$$= \int_{\mathbb{R}^3} \rho(\boldsymbol{r}, T-\Delta t|\boldsymbol{r}_0, t_0) \cdot \log\Big(\frac{\overline{\Psi}(\boldsymbol{r}, T; \tilde{\overline{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0)}\Big) d\boldsymbol{r} \tag{10}$$

$$+ D\Delta t \int_{\mathbb{R}^3} \Delta\rho(\boldsymbol{r}, T-\Delta t|\boldsymbol{r}_0, t_0) \cdot \log\Big(\frac{\overline{\Psi}(\boldsymbol{r}, T; \tilde{\overline{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0)}\Big) d\boldsymbol{r} \tag{11}$$

$$- \Delta t \int_{\mathbb{R}^3} \nabla \cdot \tilde{\overline{\boldsymbol{\Gamma}}}(\boldsymbol{r})\rho(\boldsymbol{r}, T-\Delta t|\boldsymbol{r}_0, t_0) \cdot \log\Big(\frac{\overline{\Psi}(\boldsymbol{r}, T; \tilde{\overline{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0)}\Big) d\boldsymbol{r}. \tag{12}$$

We may insert absolute values and replace subtraction with addition, yielding

$$\leq \int_{\mathbb{R}^3} \Big|\rho(\boldsymbol{r}, T-\Delta t|\boldsymbol{r}_0, t_0) \cdot \log\Big(\frac{\overline{\Psi}(\boldsymbol{r}, Tt; \tilde{\overline{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0)}\Big)\Big| d\boldsymbol{r} \tag{13}$$

$$+ D\Delta t \int_{\mathbb{R}^3} \Big|\Delta\rho(\boldsymbol{r}, T-\Delta t|\boldsymbol{r}_0, t_0) \cdot \log\Big(\frac{\overline{\Psi}(\boldsymbol{r}, T; \tilde{\overline{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0)}\Big)\Big| d\boldsymbol{r} \tag{14}$$

$$+ \Delta t \int_{\mathbb{R}^3} \Big|\nabla \cdot \tilde{\overline{\boldsymbol{\Gamma}}}(\boldsymbol{r})\rho(\boldsymbol{r}, T-\Delta t|\boldsymbol{r}_0, t_0) \cdot \log\Big(\frac{\overline{\Psi}(\boldsymbol{r}, T; \tilde{\overline{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0)}\Big)\Big| d\boldsymbol{r}. \tag{15}$$

Using the definition of the L^1 norm,

$$= \Big|\Big|\rho(\boldsymbol{r}, T-\Delta t|\boldsymbol{r}_0, t_0) \cdot \log\Big(\frac{\overline{\Psi}(\boldsymbol{r}, T; \tilde{\overline{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0)}\Big)\Big|\Big|_{L^1(\mathbb{R}^3)} \tag{16}$$

$$+ D\Delta t\Big|\Big|\Delta\rho(\boldsymbol{r}, T-\Delta t|\boldsymbol{r}_0, t_0) \cdot \log\Big(\frac{\overline{\Psi}(\boldsymbol{r}, T; \tilde{\overline{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0)}\Big)\Big|\Big|_{L^1(\mathbb{R}^3)} \tag{17}$$

$$+ \Delta t\Big|\Big|\nabla \cdot \tilde{\overline{\boldsymbol{\Gamma}}}(\boldsymbol{r})\rho(\boldsymbol{r}, T-\Delta t|\boldsymbol{r}_0, t_0) \cdot \log\Big(\frac{\overline{\Psi}(\boldsymbol{r}, T; \tilde{\overline{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0)}\Big)\Big|\Big|_{L^1(\mathbb{R}^3)}. \tag{18}$$

Now, by Hölder's inequality,

$$\leq \Big|\Big|\rho(\boldsymbol{r},T{-}\Delta t|\boldsymbol{r}_0,t_0)\Big|\Big|_{L^2(\mathbb{R}^3)}\Big|\Big|\log\Big(\frac{\overline{\Psi}(\boldsymbol{r},T;\bar{\tilde{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r},T|\boldsymbol{r}_0,t_0)}\Big)\Big|\Big|_{L^2(\mathbb{R}^3)} \tag{19}$$

$$+ D\Delta t\Big|\Big|\Delta\rho(\boldsymbol{r},T-\Delta t|\boldsymbol{r}_0,t_0)\Big|\Big|_{L^2(\mathbb{R}^3)}\Big|\Big|\log\Big(\frac{\overline{\Psi}(\boldsymbol{r},T;\bar{\tilde{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r},T|\boldsymbol{r}_0,t_0)}\Big)\Big|\Big|_{L^2(\mathbb{R}^3)} \tag{20}$$

$$+ \Delta t\Big|\Big|\nabla\cdot\bar{\tilde{\boldsymbol{\Gamma}}}(\boldsymbol{r})\rho(\boldsymbol{r},T-\Delta t|\boldsymbol{r}_0,t_0)\Big|\Big|_{L^2(\mathbb{R}^3)}\Big|\Big|\log\Big(\frac{\overline{\Psi}(\boldsymbol{r},T;\bar{\tilde{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r},T|\boldsymbol{r}_0,t_0)}\Big)\Big|\Big|_{L^2(\mathbb{R}^3)}, \tag{21}$$

and by rewriting,

$$= \Big(\Big|\Big|\rho(\boldsymbol{r},T{-}\Delta t|\boldsymbol{r}_0,t_0)\Big|\Big|_{L^2(\mathbb{R}^3)} + D\Delta t\Big|\Big|\Delta\rho(\boldsymbol{r},T-\Delta t|\boldsymbol{r}_0,t_0)\Big|\Big|_{L^2(\mathbb{R}^3)} \tag{22}$$

$$+ \Delta t\Big|\Big|\nabla\cdot\bar{\tilde{\boldsymbol{\Gamma}}}(\boldsymbol{r})\rho(\boldsymbol{r},T-\Delta t|\boldsymbol{r}_0,t_0)\Big|\Big|_{L^2(\mathbb{R}^3)}\Big)\Big|\Big|\log\Big(\frac{\overline{\Psi}(\boldsymbol{r},T;\bar{\tilde{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r},T|\boldsymbol{r}_0,t_0)}\Big)\Big|\Big|_{L^2(\mathbb{R}^3)}. \tag{23}$$

Our distribution solutions approch 0 along the boundary of $\overline{\Omega} = [-d,d]\times[-d,d]\times[-d,d]$ for sufficiently large d, assuming our distribution is primarily situated within this domain. Hence, we can approximate the L^2 norms above with

$$\approx \Big(\Big|\Big|\rho(\boldsymbol{r},T{-}\Delta t|\boldsymbol{r}_0,t_0)\Big|\Big|_{L^2(\overline{\Omega})} + D\Delta t\Big|\Big|\Delta\rho(\boldsymbol{r},T-\Delta t|\boldsymbol{r}_0,t_0)\Big|\Big|_{L^2(\overline{\Omega})} \tag{24}$$

$$+ \Delta t\Big|\Big|\nabla\cdot\bar{\tilde{\boldsymbol{\Gamma}}}(\boldsymbol{r})\rho(\boldsymbol{r},T-\Delta t|\boldsymbol{r}_0,t_0)\Big|\Big|_{L^2(\overline{\Omega})}\Big)\Big|\Big|\log\Big(\frac{\overline{\Psi}(\boldsymbol{r},T;\bar{\tilde{\boldsymbol{\Gamma}}})}{\rho(\boldsymbol{r},T|\boldsymbol{r}_0,t_0)}\Big)\Big|\Big|_{L^2(\overline{\Omega})}. \tag{25}$$

The above is the final relative entropy loss considered for the continuous versions of our solutions. Of course, we will not be dealing with continuous solutions in our data to the Smoluchowski Diffusion equation, but discretized ones. We can use Riemann sum approximations for discrete estimations of our integrals. For example,

$$\Big|\Big|\rho(\boldsymbol{r},t_i|\boldsymbol{r}_0,t_0)\Big|\Big|_{L^2(\overline{\Omega})} \approx \Big(h^3\sum_{j,k,l}\Big(\Phi^i_{j,k,l}\Big)^2\Big)^{1/2} \tag{26}$$

$$\Big|\Big|\Delta\rho(\boldsymbol{r},t_i|\boldsymbol{r}_0,t_0)\Big|\Big|_{L^2(\overline{\Omega})} \approx \Big(h^3\sum_{j,k,l}\Big(\Delta\Phi^i_{j,k,l}\Big)^2\Big)^{1/2} \tag{27}$$

$$\Big|\Big|\nabla\cdot\bar{\tilde{\boldsymbol{\Gamma}}}(\boldsymbol{r})\rho(\boldsymbol{r},t_i|\boldsymbol{r}_0,t_0)\Big|\Big|_{L^2(\overline{\Omega})} \approx \Big(h^3\sum_{j,k,l}\Big(\nabla\cdot\tilde{\boldsymbol{\Gamma}}(\boldsymbol{x})\Phi^i_{j,k,l}\Big)^2\Big)^{1/2} \tag{28}$$

for mesh size h. As previously mentioned, the differential operators applied to Φ denote estimations of the same operators applied to ρ.

It is uncertain if this loss function provides real significant benefit as opposed to an alternative modified relative entropy, such as one in which the absolute value of the integrand is taken, as this also can resolve issues that arise with relative entropy loss. It is possible convergence rates for this loss are superior, but one can always modify the learning rate to get different rates of convergence anyway.

One can also consider the alternative loss function

$$\int_{\mathbb{R}^3} \rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0) \cdot \log\left(\frac{\rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0)}{\overline{\Psi}(\boldsymbol{r}, T; \bar{\tilde{\boldsymbol{\Gamma}}})}\right) d\boldsymbol{r}, \tag{29}$$

in which case one can repeat a similar process to divide the loss into L^2 norms. This does not necessarily simplify our result however, due to the computation required to approximate $\rho(\boldsymbol{r}, T|\boldsymbol{r}_0, t_0)$ that we previously outlined, and the computation required for $\overline{\Psi}(\boldsymbol{r}, T : \bar{\tilde{\boldsymbol{\Gamma}}})$.