Sliced-Wasserstein Flows: Learning Generative Models via Single Data Pass with Guarantees

Anonymous Author(s)
Affiliation
Address
email

Abstract

1

2 1 Constructing an entropy-regularized $\mathcal{SW}_2$ gradient flow

3 Bonnotte [1] considers the IDT algorithm [2] and develops a continuity equation, given as follows:

$$\partial_t \rho_t - \nabla \cdot (\rho_t v_t) = 0, \tag{1}$$

4 where ρ_t is the density of μ_t and

$$v_t(x) \triangleq \int_{\mathbb{S}^{d-1}} \psi'_{t,\theta}(\langle \theta, x \rangle)\theta \, d\theta. \tag{2}$$

5 Here, $\psi_{t,\theta}$ denotes the Kantorovich potential between $\theta^*_\# \mu_t$ and $\theta^*_\# \nu$. In fact, one can show that, this
6 is nothing but a gradient flow in the Wasserstein spaces, given as follows:

$$\partial_t \rho = -\nabla_{\mathcal{W}_2} \mathcal{F}(\rho) \tag{3}$$

7 where $\nabla_{\mathcal{W}_2}$ denotes a notion of gradient in the $\mathcal{W}_2$ metric and the function $\mathcal{F}$ is chosen as the
8 sliced-Wasserstein distance between μ and ν:

$$\mathcal{F}(\rho) \triangleq \frac{1}{2} \mathcal{SW}_2^2(\rho, \pi) \tag{4}$$

9 where π denotes the density of ν. Here we abused the notation by defining $\mathcal{SW}_2$ on the densities
10 instead of the measures; we implicitly assume that both measures μ and ν are dominated by the
11 Lebesgue measure. This gradient flow basically constructs a path $(\rho_t)_{t \geq 0}$ that minimizes $\mathcal{F}$ as t
12 increases. In other words, the goal in construction such a flow is to solve the following problem:

$$\rho^\star = \arg\min_{\rho} \mathcal{F}(\rho). \tag{5}$$

13 Since we obviously have $\rho^\star = \pi$, this gradient flow will start from μ_0 and bring it closer to π as t
14 evolves.

15 By following [3], the gradient given in (3) can be written as follows:

$$\nabla_{\mathcal{W}_2} \mathcal{F}(\rho) = -\nabla \cdot \Big(\rho \nabla \big(\frac{\delta \mathcal{F}}{\delta \rho}(\rho) \big) \Big), \tag{6}$$

16 where $\frac{\delta \mathcal{F}}{\delta \rho}(\rho)$ denotes the first variation of $\mathcal{F}$.

17 By this definition, μ_t directly goes to ν. However, in practical settings, we will have finitely
18 many samples from μ_0 and ν, therefore this scheme will somehow 'overfit' to the data distribution.

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

Therefore what we propose is to somehow 'regularize' the gradient flow by introducing an entropy term to the minimization process. In particular, we modify the gradient flow given in (3) as follows:

$$\partial_t \rho = -\nabla_{\mathcal{W}_2} \mathcal{F}_\lambda(\rho), \tag{7}$$

where

$$\mathcal{F}_\lambda(\rho) \triangleq \mathcal{F}(\rho) + \lambda \mathcal{H}(\rho). \tag{8}$$

Here, $\mathcal{H}(\rho) \triangleq \int (h \circ \rho)(x) dx$ denotes the negative entropy of ρ with $h(t) = t \log t$. This regularization somewhat corresponds to assuming a Gaussian prior on the density ρ: when λ goes to infinity, the optimal ρ that minimizes $\mathcal{F}_\lambda$ will be a Gaussian density since the Gaussian densities have the maximum entropy.

This time the optimization problem is modified:

$$\min_\rho \mathcal{F}_\lambda(\rho), \tag{9}$$

in which π is no longer an optimizer. The idea in this new gradient flow formulation is to take ρ_t as close as possible to π, while trying to keep its entropy at a certain level, so that it would be expressive for generative modeling purposes.

We need to compute the gradient of $\mathcal{F}_\lambda$ by using the definition given in (6). We first start by computing the first variation of the functional:

$$\frac{\delta \mathcal{F}_\lambda}{\delta \rho}(\rho) = \frac{\delta \mathcal{F}}{\delta \rho}(\rho) + \frac{\delta \mathcal{H}}{\delta \rho}(\rho) \tag{10}$$

$$= \frac{\delta \mathcal{F}}{\delta \rho} + \lambda(\log \rho + 1). \tag{11}$$

When we use this identity in (6):

$$\nabla_{\mathcal{W}_2} \mathcal{F}_\lambda(\rho) = -\nabla \cdot \Big(\rho \nabla \big(\frac{\delta \mathcal{F}_\lambda}{\delta \rho}(\rho) \big) \Big) \tag{12}$$

$$= -\nabla \cdot \Big(\rho \nabla \big(\frac{\delta \mathcal{F}}{\delta \rho} + \lambda(\log \rho + 1) \big) \Big) \tag{13}$$

$$= -\nabla \cdot \Big(\rho \nabla \big(\frac{\delta \mathcal{F}}{\delta \rho} \big) \Big) - \lambda \nabla \cdot \Big(\rho \nabla \big(\log \rho + 1 \big) \Big) \tag{14}$$

By using $\nabla(\log \rho + 1) = \nabla \log \rho = \frac{\nabla \rho}{\rho}$, we have:

$$\nabla_{\mathcal{W}_2} \mathcal{F}_\lambda(\rho) = \nabla \cdot \Big(\rho \nabla \big(\frac{\delta \mathcal{F}}{\delta \rho} \big) \Big) - \lambda \nabla \cdot \Big(\nabla \rho \Big) \tag{15}$$

$$= \nabla \cdot \Big(\rho \nabla \big(\frac{\delta \mathcal{F}}{\delta \rho} \big) \Big) - \lambda \Delta \rho, \tag{16}$$

where Δ denotes the Laplacian operator.

Since we already know the definition of $\nabla \big(\frac{\delta \mathcal{F}}{\delta \rho} \big)$ from (2), we can now construct the modified gradient flow as follows:

$$\partial_t \rho = -\nabla_{\mathcal{W}_2} \mathcal{F}_\lambda(\rho) \tag{17}$$

$$= \nabla \cdot (\rho \, v_t) + \lambda \Delta \rho. \tag{18}$$

2 Connecting with stochastic differential equations

We now consider the modified flow given in (18). We can observe that, this equation is the Fokker-Planck equation associated with the following stochastic differential equation (SDE):

$$dX_t = -v_t(X_t)dt + \sqrt{2\lambda} dW_t, \tag{19}$$

where W_t denotes the standard Brownian motion. In practice we can simulate this SDE by using the Euler-Maruyama scheme:

$$X_{n+1} = X_n - v_n(X_n) + \sqrt{2\lambda} Z_{n+1}, \tag{20}$$

where $\{Z_n\}_n$ denotes a series of standard Gaussian random variables. In practical applications, it will not be possible to exactly simulate v_n, therefore we will need to develop an unbiased estimator of v_n, such that $\mathbb{E}[\hat{v}_n(x)] = v_n(x)$ for all n and x. After that we might hope to have some error bounds on our estimation.

3 Open questions

1. We assume that the flow given in Bonnotte converges to ν. Can we say/prove something about this?
   Do you think there is some restrictive assumption to put on ν that would make it clear? Something less limiting than the Gaussian assumption, that's kind of useless. What about your intuition of overfitting? Let's say that actually, we do not observe ν, but rather some $\hat{\nu}$ that is atomic, and just a sum of diracs over the observations, maybe convergence to $\hat{\nu}$ is obtained through Bonnotte's scheme? The problem is that actually, this solution is probably not our true objective, and these stuff would kind of justify your idea of justifiying our regularisation as avoiding overfitting.
2. We need to develop an unbiased (or maybe biased) estimator for v_n. We also need to think about the 'single-data-pass' aspect? Can we still do a single data pass in this scheme?
   I understand this means we approximate the sum over θ as a finite sum, right? It's still a bit unclear to me, but in the 1D case, $\psi'_{t,\theta}(\langle\theta, x\rangle)$ is precisely given by the transport map (for which we have the analytical expression), right? In this case, I think I will implement this approximation by picking a different set of θ each time, just like they do in IDT.
   Concerning the single data-pass aspect, maybe I'm wrong but I think it's not changed: all we need for building the transport map, whatever the current ρ_t, is the distribution $\theta^*_\#\nu$ (that does not depend on ρ_t). Correct me if I'm wrong, but in practice, it looks to me we are simply adding a Gaussian noise term to the solution of the IDT, right?
3. When we use the entropy regularization, the flow will no longer converge to ν (assuming the first flow converges to ν). Let's say it converges to ν_λ. Can we show a bound between ν and ν_λ?
   I think some answers will be found in [4]
4. Let's say we developed an estimator for v_n (or more generally v_t). Can we show error bounds? I am sure there are related studies to this. I will check the relevant literature. I am guessing that we might need to assume some sort of regularity in v_t in terms of t.
5. This scheme reminds me the normalizing flows [5], continuous-time flows [6] and Stein Variational Descent [7, 8]. We need to understand the differences/similarities.

References

[1] Nicolas Bonnotte. *Unidimensional and evolution methods for optimal transportation*. PhD thesis, Paris 11, 2013.

[2] Francois Pitie, Anil C Kokaram, and Rozenn Dahyot. N-dimensional probability density function transfer and its application to color transfer. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1434–1439. IEEE, 2005.

[3] Filippo Santambrogio. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 7(1):87–154, 2017.

[4] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.

[5] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

[6] Changyou Chen, Chunyuan Li, Liqun Chen, Wenlin Wang, Yunchen Pu, and Lawrence Carin. Continuous-time flows for efficient inference and density estimation. *arXiv preprint arXiv:1709.01179*, 2018.

[7] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2378–2386, 2016.

[8] Qiang Liu. Stein variational gradient descent as gradient flow. In *Advances in neural information processing systems*, pages 3118–3126, 2017.