Wasserstein Distance in Machine Learning

Huidong (Leon) Liang

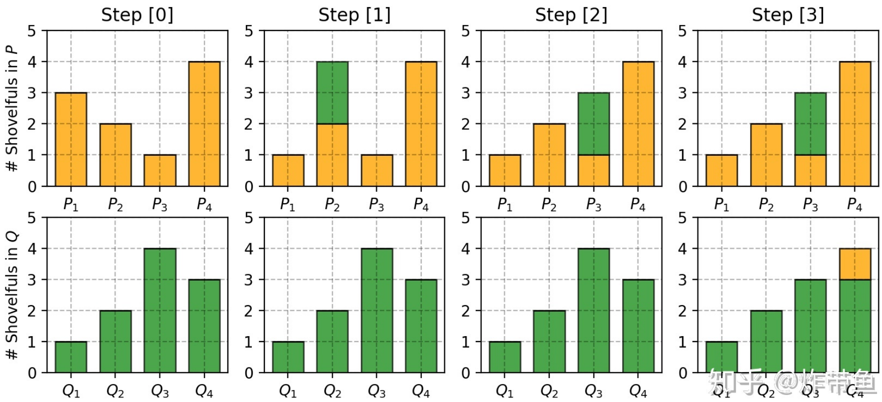
Discipline of Business Analytics
The University of Sydney Business School

November 2021

Overview

Wasserstein Distance

Optimal Transport
Earth Mover Distance



- 1-Wasserstein distance between two distributions $\mathbb{P}_r$ and $\mathbb{P}_g$:

$$W_1(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \mathcal{P}(r \sim \mathbb{P}_r, z \sim \mathbb{P}_g)} \mathbb{E}[\|r - z\|_2] \tag{1}$$

  where $\mathcal{P}(r \sim \mathbb{P}_r, z \sim \mathbb{P}_g)$ is the set of all joint distributions $\gamma(r, g)$ with marginals $\mathbb{P}_r$ and $\mathbb{P}_g$ respectively.

- Kantorovich-Rubinstein duality:

$$W_1(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \leq 1} \left\{ \mathbb{E}_{r \sim \mathbb{P}_r}[f(r)] - \mathbb{E}_{z \sim \mathbb{P}_g}[f(z)] \right\} \tag{2}$$

  where f is any continuous function that satisfies 1-Lipschitz continuity.

Wasserstein Distance, KL divergence and JS divergence

Comparison

- Kullback-Leibler Divergence

$$\mathbf{KL}(\mathbb{P}_r||\mathbb{P}_g) = \int \log(\frac{\mathbb{P}_r(x)}{\mathbb{P}_g(x)})\mathbb{P}_r(x)dx \tag{3}$$

- Jensen-Shannon Divergence

$$\mathbf{JS}(\mathbb{P}_r||\mathbb{P}_g) = \frac{1}{2}\mathbf{KL}(\mathbb{P}_r||\frac{\mathbb{P}_g + \mathbb{P}_r}{2}) + \frac{1}{2}\mathbf{KL}(\mathbb{P}_g||\frac{\mathbb{P}_g + \mathbb{P}_r}{2}) \tag{4}$$

- 1-Wasserstein Distance

$$W_1(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \mathcal{P}(r \sim \mathbb{P}_r, z \sim \mathbb{P}_g)} \mathbb{E}[\|r - z\|_2] \tag{5}$$

- Let $z \sim U[0, 1]$ be a random variable from an 1-dimensional uniform distribution. Let $\mathbb{P}_0$ be a distribution of $(0, z) \in \mathbb{R}^2$ (i.e. 0 on the x-axis and z on the y-axis), which is uniform on a vertical line that passes through the origin. We now define $\mathbb{P}_\theta(z) of (\theta, z)$ with one single parameter θ [1]. It is not hard to find the following:
  - $W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|$,
  - $JS(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0 \,, \\ 0 & \text{if } \theta = 0 \,, \end{cases}$
  - $KL(\mathbb{P}_\theta \| \mathbb{P}_0) = KL(\mathbb{P}_0 \| \mathbb{P}_\theta) = \begin{cases} +\infty & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases}$

- Although this simple example features distributions with disjoint supports, the same conclusion holds when the supports have a non empty intersection contained in a set of measure zero. This happens to be the case when two low dimensional manifolds intersect in general position [1].

Wasserstein GAN

Related Work: GAN [2]

Loss Function and Algorithm

- Objective Function

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

- Algorithm

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

- the Optimal Discriminator

$$-P_r(x)\log D(x) - P_g(x)\log[1-D(x)]$$

$$-\frac{P_r(x)}{D(x)} + \frac{P_g(x)}{1-D(x)} = 0$$

$$D^*(x) = \frac{P_r(x)}{P_r(x)+P_g(x)}$$

- Substitute back to the objective function

$$\mathbb{E}_{x\sim P_r}\log\frac{P_r(x)}{\frac{1}{2}[P_r(x)+P_g(x)]} + \mathbb{E}_{x\sim P_g}\log\frac{P_g(x)}{\frac{1}{2}[P_r(x)+P_g(x)]} - 2\log 2$$

$$2JS(P_r||P_g) - 2\log 2$$

- Kantorovich-Rubinstein duality:

$$W_1(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \leq 1} \left\{ \mathbb{E}_{r \sim \mathbb{P}_r}[f(r)] - \mathbb{E}_{z \sim \mathbb{P}_g}[f(z)] \right\} \tag{6}$$

  where f is any continuous function that satisfies 1-Lipschitz continuity.

- Loss Function of Wasserstein GAN [1]

$$\max_{w \in \mathcal{W}} \mathbb{E}_{x \sim \mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z \sim p(z)}[f_w(g_\theta(z))]$$

  w is clipped in to a compact space $\mathcal{W}$ for Lipschitz constraint.

- gradient penalty for Lipschitz constraint: add $\lambda \cdot \mathbb{E}(||\nabla f(x)|| - 1)^2$ to the objective.

Wasserstein GAN

Algorithm

- Wasserstein GAN Algorithm

while θ has not converged **do**

  for $t = 0, \dots, n_{\text{critic}}$ **do**

    Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.

    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of priors.

    $g_w \leftarrow \nabla_w \left[\frac{1}{m} \sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m} \sum_{i=1}^{m} f_w(g_\theta(z^{(i)})) \right]$

    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$

    $w \leftarrow \text{clip}(w, -c, c)$

  end for

  Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.

  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$

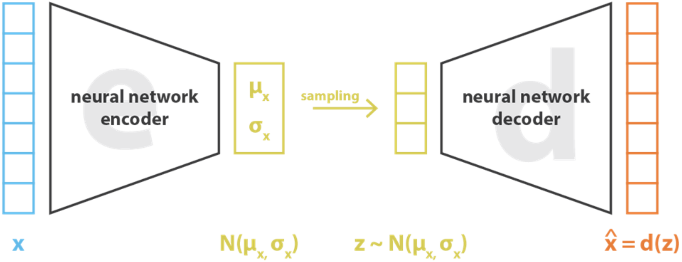
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$

end while

Wasserstein Auto-Encoders

Related Work: VAE [3]

- Model Structure



- Loss function

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\Big[\log p(\mathbf{x}|\mathbf{z})\Big] - \mathbf{KL}\Big(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\Big) \tag{7}$$

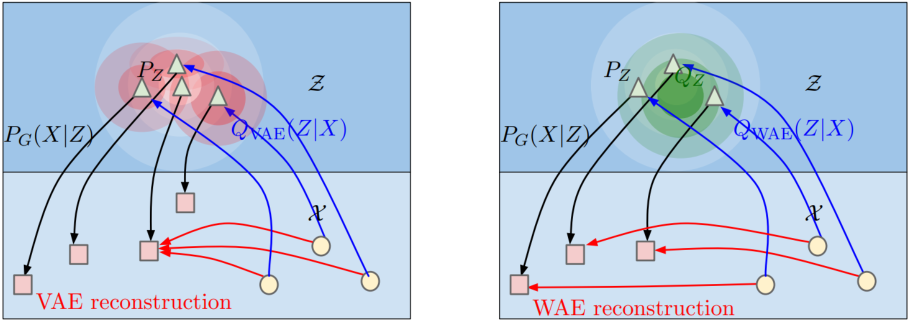


Figure from Wasserstein Auto-Encoders [4]

- Objective Function

$$\inf_{\Gamma\in\mathcal{P}(X\sim P_X, Y\sim P_G)} \mathbb{E}_{(X,Y)\sim\Gamma}\left[c(X,Y)\right] = \inf_{Q:\, Q_Z=P_Z} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}\left[c(X,G(Z))\right]$$

where Q_z is the marginal distribution of Z when $X \sim P_X$ and $Z \sim Q(Z|X)$

$$p_G(x) := \int_{\mathcal{Z}} p_G(x|z)p_z(z)dz, \quad \forall x \in \mathcal{X}$$

- Relax the constraints on Q_z by adding a penalty term

$$D_{\text{WAE}}(P_X, P_G) := \inf_{Q(Z|X)\in\mathcal{Q}} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}\left[c(X,G(Z))\right] + \lambda \cdot \mathcal{D}_Z(Q_Z, P_Z)$$

- GAN-based Penalty (WAE-GAN)

$$\mathcal{D}_Z(Q_Z, P_Z) \;=\; D_{\mathrm{JS}}(Q_Z, P_Z)$$

- Maximum Mean Discrepancy (MMD)-based Penalty (WAE-MMD)

$$\mathrm{MMD}_k(P_Z, Q_Z) = \Big\| \int_{\mathcal{Z}} k(z,\cdot)dP_Z(z) - \int_{\mathcal{Z}} k(z,\cdot)dQ_Z(z) \Big\|_{\mathcal{H}_k}$$

The End

Reference

Martin Arjovsky, Soumith Chintala, and Léon Bottou.
Wasserstein generative adversarial networks.
In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio.
Generative adversarial nets.
Advances in Neural Information Processing Systems, 27, 2014.

Diederik P Kingma and Max Welling.
Auto-encoding variational Bayes.
In *International Conference on Learning Representations*, 2014.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf.
Wasserstein auto-encoders.
In *International Conference on Learning Representations*, 2018.