Wassertein autoencoders

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf

Arthur Gretton's notes

November 1, 2018

Summary

What the paper does:

We learn:

- A generative model G that maps (decodes) from a fixed distribution P_Z on a latent space $\mathcal{Z}$ to the space of observations $\mathcal{X}$.
  - The model minimises approximate Wasserstein loss to data distribution P_X
- An approximate Wasserstein loss for any distance measure
  - the loss is specified via a learned encoder

Why interesting in theory

- A learnable estimate of the Wasserstein distance

Why interesting in practice

- Idea is similar to variational autoencoder, but with arguably more reasonable latent space behaviour
- Can define non-adversarial learning and still get good samples

The setting

- "True" data distribution P_X
- Latent variable model P_G specified by a prior P_Z on latent codes $z \in \mathcal{Z}$
  - Generative model $P_G(Y|Z)$.

The setting

- "True" data distribution P_X
- Latent variable model P_G specified by a prior P_Z on latent codes $z \in \mathcal{Z}$
  - Generative model $P_G(Y|Z)$.
- Train model P_G by minimizing optimal transport distance

$$W_c(P_X, P_G) = \inf_{\Gamma \in \mathcal{P}(X \sim P_X, Y \sim P_G)} E_{(X,Y)\sim\Gamma}[c(X, Y)]$$

  - $\mathcal{P}(X \sim P_X, Y \sim P_G)$ are distributions with marginals P_X, P_G
  - In general, caligraphic $\mathcal{P}$ is set of distributions, upper case P is a particular distribution.
  - In their experiments, authors use the square loss,

$$c(x, y) = \|x - y\|_2^2.$$

Detail: the decoder

The generation procedure: first sample from $P(Z)$ then generate $Y|Z$ using

$$p_G(y) = \int_{\mathcal{Z}} p_G(y|z)p(z)dz.$$

Assume $P_G(Y|Z)$ is deterministic, so $G =: \mathcal{Z} \to \mathcal{X}$ is a function.
A simple random decoder will be considered later.

The encoder and the O.T. loss

Given we have the deterministic generator/decoder.
Theorem: we have the equivalence:

$$W_c(P_X, P_G) = \inf_{\Gamma \in P(X \sim P_X, Y \sim P_G)} E_{(X,Y)\sim\Gamma}[c(X,Y)]$$
$$= \inf_{Q_{Z|X} : Q_Z = P_Z} E_{P_X} E_{Q(Z|X)}[c(X, G(Z))]$$

where $Q(Z|X)$ is the encoder, and we have restricted the *marginal*

$$Q(Z) := \int Q(Z|x)P(x)dx$$

to be equal to $P(Z)$.

The encoder and the O.T. loss

Given we have the deterministic generator/decoder.

Theorem: we have the equivalence:

$$W_c(P_X, P_G) = \inf_{\Gamma \in P(X \sim P_X, Y \sim P_G)} E_{(X,Y)\sim\Gamma}[c(X, Y)]$$
$$= \inf_{Q_{Z|X} : Q_Z = P_Z} E_{P_X} E_{Q(Z|X)}[c(X, G(Z)]$$

where $Q(Z|X)$ is the encoder, and we have restricted the *marginal*

$$Q(Z) := \int Q(Z|x)P(x)dx$$

to be equal to $P(Z)$.

- Note: the coupling Γ between X and Y is replaced by coupling $Q(Z|X)$ to the random Z, since from Z to Y is a deterministic mapping. This only makes sense if $Q(Z)$ matches $P(Z)$, which means it gives the right marginal over Y.

Proof of the theorem

Our joint probability families:

- $\mathcal{P}(P_X, P_G)$ are joint distributions with marginals P_X, P_G.

Proof of the theorem

Our joint probability families:

- $\mathcal{P}(P_X, P_G)$ are joint distributions with marginals P_X, P_G.
- Likweise: $\mathcal{P}(P_X, P_Z)$

Proof of the theorem

Our joint probability families:

- $\mathcal{P}(P_X, P_G)$ are joint distributions with marginals P_X, P_G.
- Likweise: $\mathcal{P}(P_X, P_Z)$
- $\mathcal{P}_{XYZ}$: joint distributions such that $X \sim P_X$, $(Y, Z) \sim P_{GZ} := P(Z)P_G(Y|Z)$, and $(Y \perp X)|Z$. I.e. all joint distributions with the correct generator, and no connection between X and Y besides the code vector.
  - This represents a subset of the family of allowable couplings Γ between X and Y: both of the marginals are correct, but coupling can only happen via Z.

Proof of the theorem

Our joint probability families:

- $\mathcal{P}(P_X, P_G)$ are joint distributions with marginals P_X, P_G.
- Likweise: $\mathcal{P}(P_X, P_Z)$
- $\mathcal{P}_{XYZ}$: joint distributions such that $X \sim P_X$, $(Y, Z) \sim P_{GZ} := P(Z)P_G(Y|Z)$, and $(Y \perp X)|Z$. I.e. all joint distributions with the correct generator, and no connection between X and Y besides the code vector.
  - This represents a subset of the family of allowable couplings Γ between X and Y: both of the marginals are correct, but coupling can only happen via Z.
- The marginals of the above distribution are $\mathcal{P}_{XY} \subseteq \mathcal{P}(P_X, P_G)$, i.e. $\mathcal{P}_{XY}$ are the members of $\mathcal{P}(P_X, P_G)$ where Z separates X, Y and $P_G(Y|Z)$ generates Y.

Proof of the theorem (continued)

We start with

$$W_c(P_X, P_G) \leq W_c^\dagger(P_X, P_G) := \inf_{P \in \mathcal{P}_{XY}} E_{XY}[c(X, Y)]$$

since we are taking an infimum over the smaller family $\mathcal{P}_{XY} \subseteq \mathcal{P}(P_X, P_G)$. But when is the upper bound tight?

Proof of the theorem (continued)

We start with

$$W_c(P_X, P_G) \leq W_c^{\dagger}(P_X, P_G) := \inf_{P \in \mathcal{P}_{XY}} E_{XY}[c(X, Y)]$$

since we are taking an infimum over the smaller family $\mathcal{P}_{XY} \subseteq \mathcal{P}(P_X, P_G)$.
But when is the upper bound tight?
Answer: it is tight when $Y = G(Z)$ (deterministic).

Proof of the theorem (continued)

We start with

$$W_c(P_X, P_G) \leq W_c^\dagger(P_X, P_G) := \inf_{P \in \mathcal{P}_{XY}} E_{XY}[c(X, Y)]$$

since we are taking an infimum over the smaller family $\mathcal{P}_{XY} \subseteq \mathcal{P}(P_X, P_G)$.
But when is the upper bound tight?
Answer: it is tight when $Y = G(Z)$ (deterministic).
Proof: If Y is a deterministic function of Z in the family $\mathcal{P}(P_X, P_G)$, then Z always separates X from Y:

$$E\left[\mathbb{I}_{Y \in A} | X, Z\right] = E\left[\mathbb{I}_{Y \in A} | Z\right],$$

so $\mathcal{P}_{XY} = \mathcal{P}(P_X, P_G)$.

Proof of the theorem (continued)

We start with

$$W_c(P_X, P_G) \le W_c^\dagger(P_X, P_G) := \inf_{P \in \mathcal{P}_{XY}} E_{XY}[c(X,Y)]$$

since we are taking an infimum over the smaller family $\mathcal{P}_{XY} \subseteq \mathcal{P}(P_X, P_G)$.
But when is the upper bound tight?
Answer: it is tight when $Y = G(Z)$ (deterministic).
Proof: If Y is a deterministic function of Z in the family $\mathcal{P}(P_X, P_G)$, then Z always separates X from Y:

$$E\left[\mathbb{I}_{Y \in A} | X, Z\right] = E\left[\mathbb{I}_{Y \in A} | Z\right],$$

so $\mathcal{P}_{XY} = \mathcal{P}(P_X, P_G)$.
Note: it is always true that $\mathcal{P}_{XZ} = \mathcal{P}(P_X, P_Z)$.

Proof of the theorem (continued)

Finally,

$$W_c^\dagger(P_X, P_G) := \inf_{P \in \mathcal{P}_{XY}} E_{XY}[c(X, Y)]$$

Proof of the theorem (continued)

Finally,

$$\begin{aligned} W_c^{\dagger}(P_X, P_G) &:= \inf_{P \in \mathcal{P}_{XY}} E_{XY}[c(X, Y)] \\ &= \inf_{P \in \mathcal{P}_{XYZ}} E_{P_Z} E_{X \sim P(X|Z)} E_{Y \sim P_G(Y|Z)}[c(X, Y)] \end{aligned}$$

Proof of the theorem (continued)

Finally,

$$
\begin{aligned}
W_c^{\dagger}(P_X, P_G) &:= \inf_{P \in \mathcal{P}_{XY}} E_{XY}[c(X, Y)] \\
&= \inf_{P \in \mathcal{P}_{XYZ}} E_{P_Z} E_{X \sim P(X|Z)} E_{Y \sim P_G(Y|Z)}[c(X, Y)] \\
&= \inf_{P \in \mathcal{P}_{XYZ}} E_{P_Z} E_{X \sim P(X|Z)}[c(X, G(Z))]
\end{aligned}
$$

Proof of the theorem (continued)

Finally,

$$
\begin{aligned}
W_c^{\dagger}(P_X, P_G) &:= \inf_{P \in \mathcal{P}_{XY}} E_{XY}[c(X, Y)] \\
&= \inf_{P \in \mathcal{P}_{XYZ}} E_{P_Z} E_{X \sim P(X|Z)} E_{Y \sim P_G(Y|Z)}[c(X, Y)] \\
&= \inf_{P \in \mathcal{P}_{XYZ}} E_{P_Z} E_{X \sim P(X|Z)}[c(X, G(Z))] \\
&=: \inf_{P \in \mathcal{P}_{XZ}} E_{XZ}[c(X, G(Z))]
\end{aligned}
$$

and we are done.

The *relaxed* OT loss

Rather than requiring the encoder to exacly match $P(Z)$, we just approximately match $P(Z)$. This gives the main result:

$$D_{WAE}(P_X, P_G) = \inf_{Q(Z|X) \in \mathcal{Q}} E_{P_X} E_{Q(Z|X)} c[X, G(Z)] + \lambda D_Z(Q_Z, P_Z),$$

where:

- $\mathcal{Q}$ is the set of encoders that we optimize over
- $D_Z(Q_Z, P_Z)$ is a divergence between the marginal Q_Z and the target $P(Z)$
- We can use non-random *encoders*: this just amounts to a particular choice in Γ.

The algorithm (MMD version)

The code distribution is $P(z) = \mathcal{N}(0, \sigma_z^2 I_d)$. The algorithm with MMD is:

Algorithm 2 Wasserstein Auto-Encoder with MMD-based penalty (WAE-MMD).

Require: Regularization coefficient $\lambda > 0$, characteristic positive-definite kernel k.

Initialize the parameters of the encoder Q_ϕ, decoder G_θ, and latent discriminator D_γ.

while (ϕ, θ) not converged **do** — generate the Z using the encoder.

Sample $\{x_1, \ldots, x_n\}$ from the training set

Sample $\{z_1, \ldots, z_n\}$ from the prior P_Z

Sample $\tilde{z}_i$ from $Q_\phi(Z|x_i)$ for $i = 1, \ldots, n$

Update Q_ϕ and G_θ by descending:

$$\frac{1}{n}\sum_{i=1}^{n} c\big(x_i, G_\theta(\tilde{z}_i)\big) + \frac{\lambda}{n(n-1)}\sum_{\ell \neq j} k(z_\ell, z_j)$$
$$+ \frac{\lambda}{n(n-1)}\sum_{\ell \neq j} k(\tilde{z}_\ell, \tilde{z}_j) - \frac{2\lambda}{n^2}\sum_{\ell, j} k(z_\ell, \tilde{z}_j)$$

end while

The algorithm (version we don't talk about)

The code is $P(z) = \mathcal{N}(0, \sigma_z^2 I_d)$. Algorithm with learned divergence on Z:

Algorithm 1 Wasserstein Auto-Encoder with GAN-based penalty (WAE-GAN).

Require: Regularization coefficient $\lambda > 0$.
Initialize the parameters of the encoder Q_ϕ, decoder G_θ, and latent discriminator D_γ.
while (ϕ, θ) not converged **do**
Sample $\{x_1, \ldots, x_n\}$ from the training set
Sample $\{z_1, \ldots, z_n\}$ from the prior P_Z
Sample $\tilde{z}_i$ from $Q_\phi(Z|x_i)$ for $i = 1, \ldots, n$
Update D_γ by ascending:

$$\frac{\lambda}{n} \sum_{i=1}^{n} \log D_\gamma(z_i) + \log(1 - D_\gamma(\tilde{z}_i))$$

Update Q_ϕ and G_θ by descending:

$$\frac{1}{n} \sum_{i=1}^{n} c(x_i, G_\theta(\tilde{z}_i)) - \lambda \cdot \log D_\gamma(\tilde{z}_i)$$

end while

How is this different to a variational autoencoder?

- Variational autoencoder: requires $Q(Z|X = x)$ to be close to $P(Z)$ for each example x: this pulls all representations towards the same "prior".
- Wasserstein autoencoder: requires only the marginal distributions on the latents to match, $Q(Z) = \int Q(Z|X)dP_X$ to match $P(Z)$

What if *decoder* is random?

For random decoders $P_G(X|Z)$, then the bound may no longer be tight.
Assuming $c(x, y) = \|x - y\|_2^2$ and
$P_G(Y|Z = z) \sim \mathcal{N}(G(z), \mathrm{diag}([\sigma_1^2, \ldots, \sigma_d^2]))$. Then

$$W_c(P_X, P_G) \leq W_c^{\dagger}(P_X, P_G) = \sum_{i=1}^{d} \sigma_i^2 + \inf_{P \in \mathcal{P}_{XZ}} E_{XZ}\|X - G(Z)\|_2^2$$

What if *decoder* is random?

For random decoders $P_G(X|Z)$, then the bound may no longer be tight.
Assuming $c(x, y) = \|x - y\|_2^2$ and
$P_G(Y|Z = z) \sim \mathcal{N}(G(z), \mathrm{diag}([\sigma_1^2, \dots, \sigma_d^2]))$. Then

$$W_c(P_X, P_G) \le W_c^\dagger(P_X, P_G) = \sum_{i=1}^{d} \sigma_i^2 + \inf_{P \in \mathcal{P}_{XZ}} E_{XZ}\|X - G(Z)\|_2^2$$

Proof: first, recall

$$W_c^\dagger(P_X, P_G) = \inf_{P \in \mathcal{P}_{XYZ}} E_{P_Z} E_{X \sim P(X|Z)} E_{Y \sim P_G(Y|Z)}[\|X - Y\|_2^2]$$

What if *decoder* is random?

For random decoders $P_G(X|Z)$, then the bound may no longer be tight.
Assuming $c(x, y) = \|x - y\|_2^2$ and
$P_G(Y|Z = z) \sim \mathcal{N}(G(z), \mathrm{diag}([\sigma_1^2, \ldots, \sigma_d^2]))$. Then

$$W_c(P_X, P_G) \le W_c^\dagger(P_X, P_G) = \sum_{i=1}^{d} \sigma_i^2 + \inf_{P \in \mathcal{P}_{XZ}} E_{XZ}\|X - G(Z)\|_2^2$$

Proof: first, recall

$$W_c^\dagger(P_X, P_G) = \inf_{P \in \mathcal{P}_{XYZ}} E_{P_Z} E_{X \sim P(X|Z)} E_{Y \sim P_G(Y|Z)} [\|X - Y\|_2^2]$$

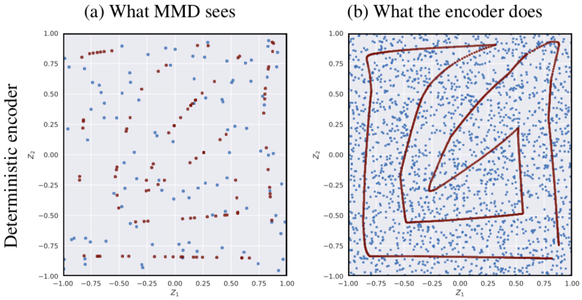
Next

$$\begin{aligned}
&E_{Y \sim P_G(Y|Z)} \|X - Y\|_2^2 \\
&= E_{Y \sim P_G(Y|Z)} \|X - G(Y) + G(Y) - Y\|_2^2 \\
&= \|X - G(Z)\|_2^2 + E_{Y \sim P(Y|Z)} \|G(Z) - Y\|_2^2.
\end{aligned}$$

Why should the *encoder* be random?

"On the latent space of Wasserstein auto-encoders", Rubenstein, Schoelkopf, Tolstikhin.

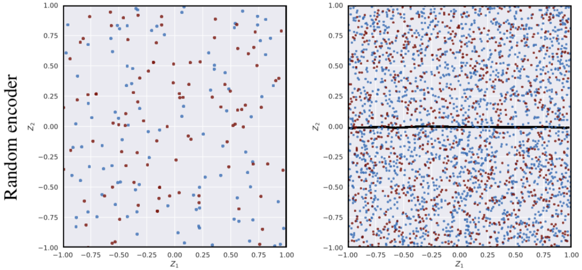
In this case, input variability is one dimensional, latent space dimension is 2.



Why should the *encoder* be random?

"On the latent space of Wasserstein auto-encoders", Rubenstein, Schoelkopf, Tolstikhin.

In this case, input variability is one dimensional, latent space dimension is 2.



You should volunteer for MLSS!

- MLSS will take place at Gatsby next July 15-26
- If you help (eg selecting students, registering when they arrive) you get to attend for free!