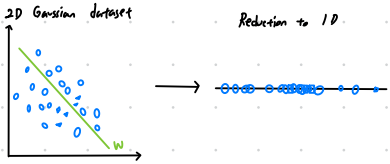
Recap - Linear Dimensionality Reduction

Ex)



What would be a good reduction?

① Find w s.t. it maximizes the variance of the projected data

② Find w s.t. it minimizes the reconstruction error

① = ② : same

① Find w s.t. it maximizes the variance of the projected data

$$\text{var} = \frac{1}{N}\sum_{i=1}^{N}\left(w^T(x^{(i)}-\bar{x})\right)^2 = \frac{1}{N}\sum_{i=1}^{N} w^T(x^{(i)}-\bar{x})(x^{(i)}-\bar{x})w$$

$$= w^T \underbrace{\frac{1}{N}\sum_{i=1}^{N}(x^{(i)}-\bar{x})(x^{(i)}-\bar{x})}_{S} w = w^T S w$$

Hence, maximizing the variance can be written as

$$\max_w \; w^T S w \quad \text{s.t. } w^T w = 1$$

by applying Lagrangian function,

we get

$$L(w,a) = w^T S w + a(1-w^T w) = w^T S w + a - a w^T w$$

We need to find $\frac{\partial L}{\partial w} = 0$ to get optimal w

$$\frac{\partial L}{\partial w} = 2Sw - 2aw$$

$$2Sw - 2aw = 0$$

$$Sw = aw$$

We derived to the equation for getting eigenvector & eigenvalue.

★ Variance is a measure of the variability of the data. Therefore, if you were to select a single principal component, you would want it to account for the most variability possible.

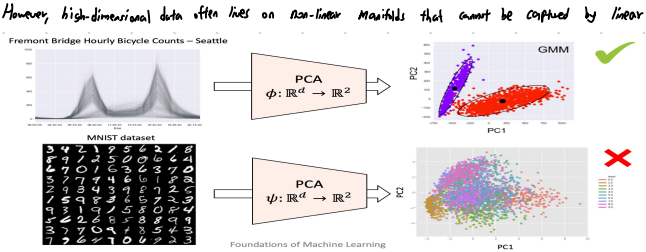
How to project a vector x onto the subspace spanned by the unit vector w?

$$\therefore \hat{x} = ww^T x$$

Therefore, the reconstruction error :

$$\frac{1}{N}\sum_{i=1}^{N}\|\hat{x}^{(i)} - x^{(i)}\|^2 = \frac{1}{N}\sum_{i=1}^{N}\|ww^T x^{(i)} - x^{(i)}\|^2$$

However, high-dimensional data often lies on non-linear manifolds that cannot be captured by linear models such as PCA ($ww^T x^{(i)}$ is linear)



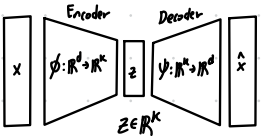
PCA vs Auto-Encoder (AE)

PCA

PCA: Linear dimensionality Reduction

- Forward transform: $z = W^T x$
- Inverse transform: $\hat{x} = Wz$
- Training loss: $\min_W \mathbb{E}_x \left[\|x - \hat{x}\|^2 \right] = \mathbb{E}_x \left[\|x - WW^T x\|^2 \right]$

  s.t $W^T W = I_{k \times k}$

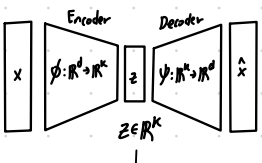


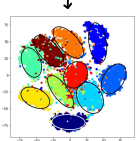
AE: Nonlinear dimensionality Reduction

- Forward transform: $z = \phi(x)$
- Inverse transform: $\hat{x} = \psi(z) = \psi(\phi(x))$
- Training loss: $\min_{\phi,\psi} \mathbb{E}_x \left[\|x - \hat{x}\|^2 \right] = \mathbb{E}_x \left[\|x - \psi(\phi(x))\|^2 \right]$

Auto Encoder

- Autoencoding is a form of compression. Smaller latent space will force a larger training bottleneck
- Hidden layer forces network to learn a compressed latent representation
- Reconstruction loss forces latent representation to capture as much information about the data as possible.





Need to estimate the latent distribution post-hoc.

How can we avoid Post-hoc Density Estimation?

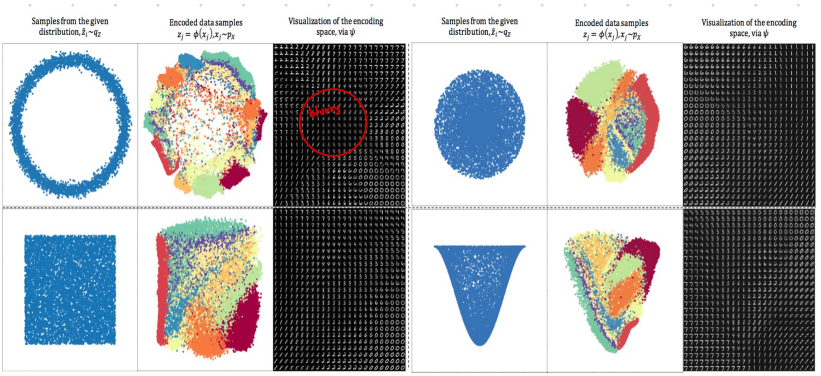
: ① VAE

② Sliced Wasserstein Auto-Encoders

⋮

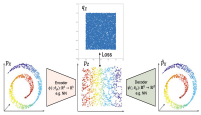
etc

What does it mean by avoiding post-hoc density estimation?



: By comparing p_z to q_z, we can model our model to produce distribution as we want. e.g. Gaussian.

Auto-Encoders with Priors on Latent Space



Loss: $\min_{\phi_\theta, \psi_\theta} \mathbb{E}_{x \sim p_x}\left[\|x - \hat{x}\|^2\right] + \lambda D(p_z, q_z)$, where D stands for dissimilarity measure.

A very common dissimilarity measure

① KL-Divergence:

$$D_{KL}(p\|q) = \mathbb{E}_{x\sim p}\left[\log \frac{p(x)}{q(x)}\right] = \int p(x) \log \frac{p(x)}{q(x)}\, dx$$

Problem w/ KL

- Not symmetric
- Not defined when we have non-overlapping support, i.e., $q(x)=0$

② Jensen-Shannon Divergence (JSD)

$$JS(p\|q) = \frac{1}{2}\left(D_{KL}\left(p \,\|\, \frac{p+q}{2}\right) + D_{KL}\left(q \,\|\, \frac{p+q}{2}\right)\right)$$

③ L_p-metrics ($p \geq 1$)

: $\left(\int_x |p(x) - q(x)|^p dx\right)^{\frac{1}{p}}$

ex) L_1 metric : $\int_x |p(x) - q(x)|\, dx$

L_2 metric: $\left(\int_x |p(x) - q(x)|^2 dx\right)^{\frac{1}{2}}$

④ Symmetric chi-squared distances:

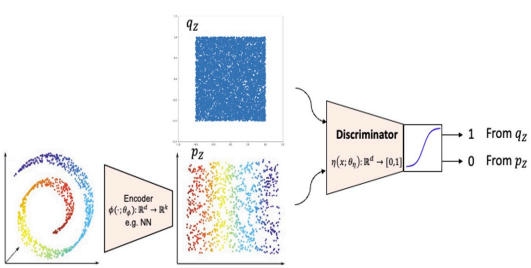
$$\int_x \frac{2(p(x) - q(x))^2}{q(x) + p(x)}\, dx$$

⑤ Hellinger distances:

$$\left(\frac{1}{2}\int_x \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 dx\right)^{\frac{1}{2}}$$

⑥ Wasserstein distance

Adversarial Network Allow us to measure Jensen-Shannon Divergence



☆ We don't have access to p_z, but we only have samples from it.

Dissimilarity between p_z and q_z.

$$D(p_z, q_z) = \max_\eta \mathbb{E}_{z\sim q_z}\left[\log(\eta(z))\right] + \mathbb{E}_{z'\sim p_z}\left[\log(1-\eta(z'))\right]$$

We want $\eta(z)=1$, $\eta(z')=0$, where $z\sim q_z$, $z'\sim p_z$, hence $\log(\eta(z)) = \log(1)$
$\log(1-\eta(z')) = \log(1)$

Loss: $\min_{\phi_\theta, \psi_\theta} \max_\eta \mathbb{E}_{x\sim p_x}\left[\|x - \psi(\phi(x))\|^2 + \log(1-\eta(\phi(x)))\right] + \mathbb{E}_{z\sim q_z}\left[\log(\eta(z))\right]$

Why Adversarial Network allow us to measure JSD?

: $\max_\eta \mathbb{E}_{z\sim q_z}\left[\log(\eta(z))\right] + \mathbb{E}_{x\sim p_x}\left[\log(1-\eta(\phi(x)))\right] = \mathbb{E}_{z\sim q_z}\left[\log(\eta(z))\right] + \mathbb{E}_{z'\sim p_z}\left[\log(1-\eta(z'))\right]$

$$= \underbrace{\int_z q_z(z)\log(\eta(z))\, dz + \int_z p_z(z)\log(1-\eta(z))\, dz}_{L}$$

Then, we can find η^* when $\frac{\partial L}{\partial \eta} = 0$

$$\frac{\partial L}{\partial \eta} = \int_z \frac{q_z(z)}{\eta(z)} dz - \int_z \frac{p_z(z)}{1-\eta(z)} dz$$

$$\int_z \frac{q_z(z)}{\eta(z)} dz - \int_z \frac{p_z(z)}{1-\eta(z)} dz = 0$$

$$\int_z \frac{q_z(z)}{\eta(z)} - \frac{p_z(z)}{1-\eta(z)} dz = 0$$

It must holds for all z, so

$$\frac{q}{\eta} - \frac{p}{1-\eta} = 0$$

$$\Rightarrow \frac{q}{\eta} = \frac{p}{1-\eta} \Rightarrow q(1-\eta) = \eta p$$

$$\Rightarrow q - q\eta = \eta p$$

$$\Rightarrow q = \eta p + \eta q$$

$$\Rightarrow q = \eta(p+q)$$

$$\Rightarrow \eta = \frac{q}{p+q}$$

$$\eta^* = \frac{q_z}{q_z + p_z}$$

Therefore,

$$\max_\eta \mathbb{E}_{z\sim q_z}[\log(\eta(z))] + \mathbb{E}_{x\sim p_x}[\log(1-\eta(\phi(x)))] = \int_z q_z(z)\log\left(\frac{q_z}{q_z+p_z}\right)dz + \int_z p_z(z)\log\left(1-\frac{q_z}{q_z+p_z}\right)dz$$

$$= \int_z q_z(z)\log\left(\frac{q_z}{q_z+p_z}\right)dz + \int_z p_z(z)\log\left(\frac{p_z}{q_z+p_z}\right)dz, \quad \text{since } \log(1-a) = \log(1) + \log(a) \text{ and } \log(1) = 0$$

$$= \int_z q_z(z)\log\left(\frac{q_z}{\frac{q_z+p_z}{2}\cdot 2}\right)dz + \int_z p_z(z)\log\left(\frac{p_z}{\frac{q_z+p_z}{2}\cdot 2}\right)dz$$

$$= \int_z q_z(z)\left(\log\left(\tfrac{1}{2}\right) \frac{q_z}{\frac{q_z+p_z}{2}}\right)dz + \int_z p_z(z)\log\left(\tfrac{1}{2}\cdot\frac{p_z}{\frac{q_z+p_z}{2}}\right)dz$$

$$= \int_z q_z(z)\left(\log\left(\tfrac{1}{2}\right) + \log\left(\frac{q_z}{\frac{q_z+p_z}{2}}\right)\right)dz + \int_z p_z(z)\left(\log\left(\tfrac{1}{2}\right) + \log\left(\frac{p_z}{\frac{q_z+p_z}{2}}\right)\right)dz$$

$$= \int_z q_z(z)\log\left(\tfrac{1}{2}\right)dz + \int_z q_z(z)\log\left(\frac{q_z}{\frac{q_z+p_z}{2}}\right)dz + \int_z p_z(z)\log\left(\tfrac{1}{2}\right)dz + \int_z p_z(z)\log\left(\frac{p_z}{\frac{q_z+p_z}{2}}\right)dz$$

Since $q_z(z)$ is pdf, $\int_z q_z(z)\,dz$ and $\int_z p_z(z)dz = 1$.

Therefore, $\int_z q_z(z)\log\left(\frac{1}{2}\right)dz = -\log(2)$

$\int_z p_z(z)\log\left(\frac{1}{2}\right)dz = -\log(2)$

$$= \int_z q_z(z)\log\left(\frac{q_z}{\frac{q_z+p_z}{2}}\right)dz + \int_z p_z(z)\log\left(\frac{p_z}{\frac{q_z+p_z}{2}}\right)dz - 2\log(2) = 2JS(p_z, q_z) - 2\log(2)$$

For the optimal discriminator, η^*: $\underset{\theta,\psi}{\operatorname{argmin}}\ \mathbb{E}_{x\sim p_x}\left[\|x-\hat{x}\|^2\right] + \lambda \cdot JS(p_z, q_z)$

Then, what are the benefits of using prior distribution?

① Regularization: By imposing a prior distribution on the latent space, such as Gaussian distribution, it helps to prevent overfitting and encourages the model to learn a more meaningful representation of the input data.

② Improved Latent space: The prior distribution guides the learning process, encouraging AE to map input data into a latent space that follows the specified distribution. It can lead to more structured and interpretable latent space representation.

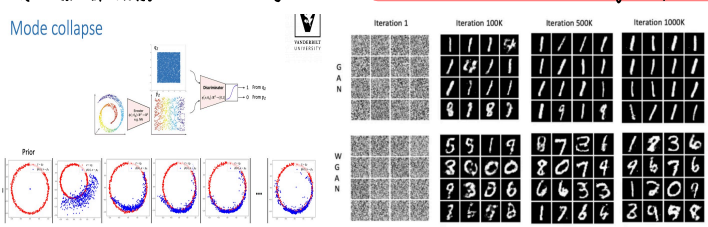
③ Enhance Generative Modelling

④ Interpretability and Controllability: w/ a prior distribution, it becomes easier to interpret and control the latent space. For example, in a Gaussian prior, each dimension of the latent space corresponds to a specific features, allowing for more meaningful manipulation of generated samples.

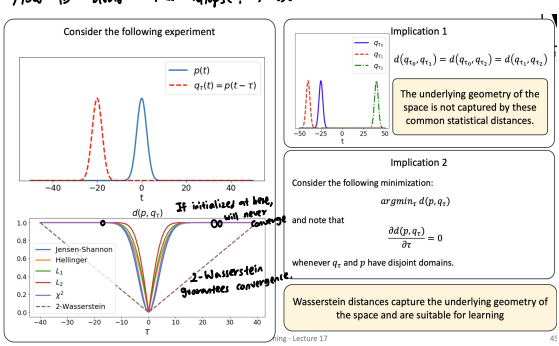
What's the problem of using JS Divergence?

: Mode Collapse.

Mode Collapse happens when the generator focuses on producing a limited set of data patterns that deceive the discriminator. It becomes fixated on a few dominant modes in the training data and fails to capture the full diversity of the data distribution.



How to avoid Mode Collapse? : use Wasserstein distance



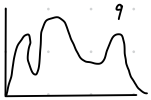
Wasserstein distance: Also known as Earth Mover's distance, is a measure of dissimilarity between two probability distributions. Given two distributions U and V on metric space (X,d) and (Y,d), the Wasserstein distance $W_p(U,V)$ is defined as the minimum cost of transporting mass from U to V, where the cost is measured as the p^{th} power of the distance function d.

$$W_p(U,V) \overset{p\geq 1}{\overset{\text{def}}{=}} \left(\min_{\gamma} \int_{X\times Y} d^p(x,y)\, d\gamma(x,y) \right)^{\frac{1}{p}}$$

s.t. $\gamma(A,Y) = U(A), \gamma(X,B) = V(B), \forall A \subseteq Y, \forall B \subseteq X$

Before dive deep into Wasserstein distance, how to sample data from an unknown probability density function?

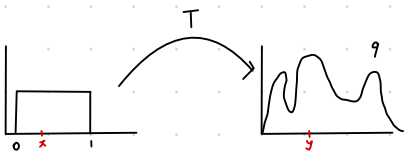
let's say there's q, unknown pdf



how to sample data from q?

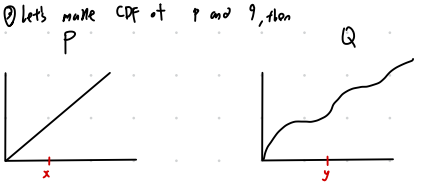
① First, generate a uniform distribution, p

② let's say there's T that map x to y.



s.t. $T(x) = y$

③ Let's make CDF of p and q, then



We need mapping x to y s.t. $P(x) = Q(y)$

Since $T(x) = y$, $P(x) = Q(T(x))$

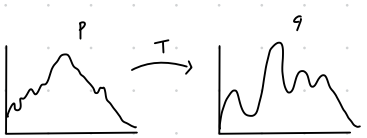
$$Q^{-1}(P(x)) = T(x)$$

Since p is uniform distribution, $P(x) = x$, hence

$$Q^{-1}(x) = T(x)$$

And T is optimal transport map.

Now, let there two unknown p, q



Expected transport cost : $\int \|T(x) - x\|^2 p(x)\, dx$

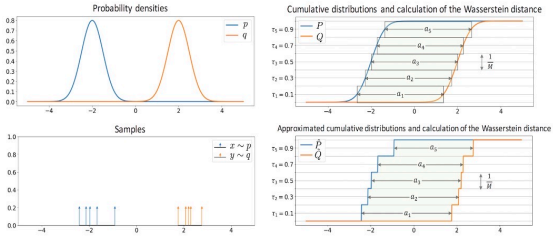
$= \int \|Q^{-1}(P(x)) - x\|^2 p(x)\, dx$ (P: cdf, p: pdf)

Let $P(x) = v$, then $x = P^{-1}(v)$, and $p(x)\,dx = dv$

$= \int \|Q^{-1}(v) - P^{-1}(v)\|^2 dv.$ It's the general idea of Wasserstein distance.

w/ 2-Wasserstein,

Loss: $\min_{\phi_e, \psi_\theta} \mathbb{E}_{x \sim p_X}\left[\|x - \hat{x}\|^2\right] + \lambda W_2^2(p_z, q_z)$



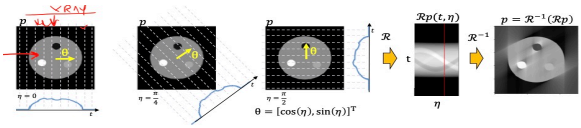
where $W_2^2(P, Q) = \int_0^1 |F^{-1}(u) - Q^{-1}(u)|^2 du$

and $P(t) = \int_{-\infty}^{t} p(x)dx = M((-\infty, t])$ where $p(x)$ is pdf, and $P(t)$ is cdf

$Q(t) = \int_{-\infty}^{t} q(x)dx = Y_t((-\infty, t])$

Slicing high-dimensional distribution

: $R_p(t, \theta) = \int_X \delta(t - x \cdot \theta) p(x) dx, \quad t \in \mathbb{R}, \ \theta \in S^{d-1}$



And, sliced Wasserstein distance:

$SW_p(u, v) = \left(\int_{S^{d-1}} W_p^p(R_p(\cdot, \theta), R_q(\cdot, \theta))\, d\theta\right)^{\frac{1}{p}}$

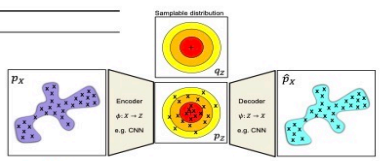
Monte-Carlo Integration:

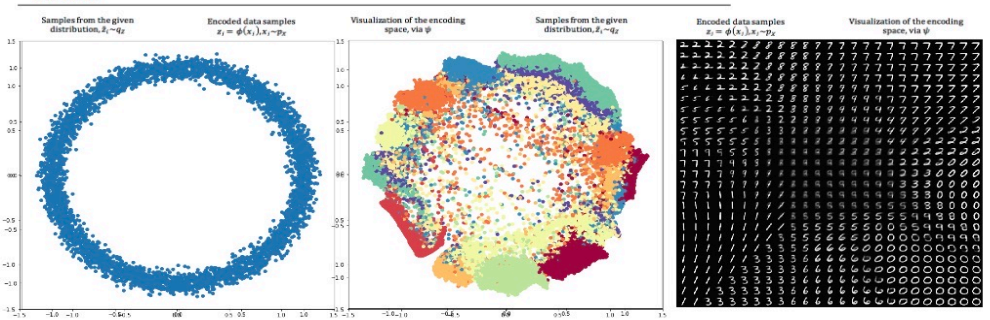
$SW_{p,L}(u, v) = \left(\frac{1}{L}\sum_{\ell=1}^{L} W_p^p(R_p(\cdot, \theta_\ell), R_q(\cdot, \theta_\ell))\right)^{\frac{1}{p}}$

Sliced Wasserstein Auto-Encoders

Algorithm 1 Sliced-Wasserstein Auto-Encoder (SWAE)

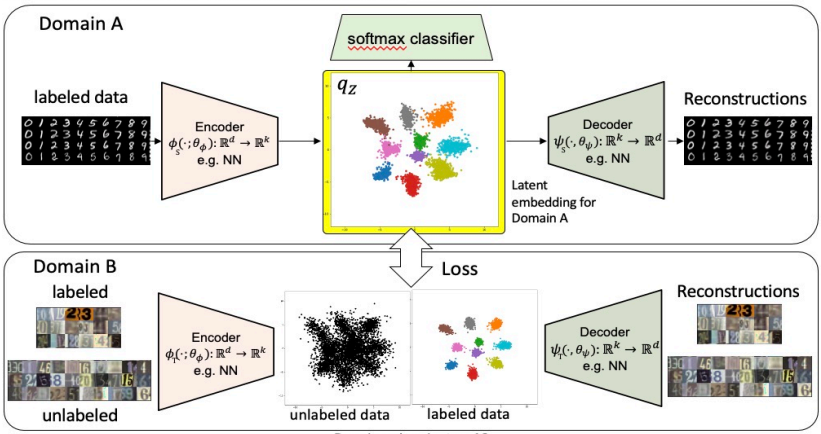
Require: Regularization coefficient λ, and number of random projections, L.
Initialize the parameters of the encoder, ϕ, and decoder, ψ
while ϕ and ψ have not converged **do**
Sample $\{x_1, ..., x_M\}$ from training set (i.e. p_X)
Sample $\{\tilde{z}_1, ..., \tilde{z}_M\}$ from q_Z
Sample $\{\theta_1, ..., \theta_L\}$ from $\mathbb{S}^{K-1}$
Sort $\theta_l \cdot \tilde{z}_M$ such that $\theta_l \cdot \tilde{z}_{i[m]} \leq \theta_l \cdot \tilde{z}_{i[m+1]}$
Sort $\theta_l \cdot \phi(x_m)$ such that $\theta_l \cdot \phi(x_{j[m]}) \leq \theta_l \cdot \phi(x_{j[m+1]})$
Update ϕ and ψ by descending: $\sum_{m=1}^{M} c(x_m, \psi(\phi(x_m))) + \lambda \sum_{l=1}^{L} \sum_{m=1}^{M} c(\theta_l \cdot \tilde{z}_{i[m]}, \theta_l \cdot \phi(x_{j[m]}))$
end while





SWAE for Domain Adaptation

Main idea: Set the prior distribution, q_Z, for Domain B to be the latent distribution of Domain A.



SWAE for domain adaptation