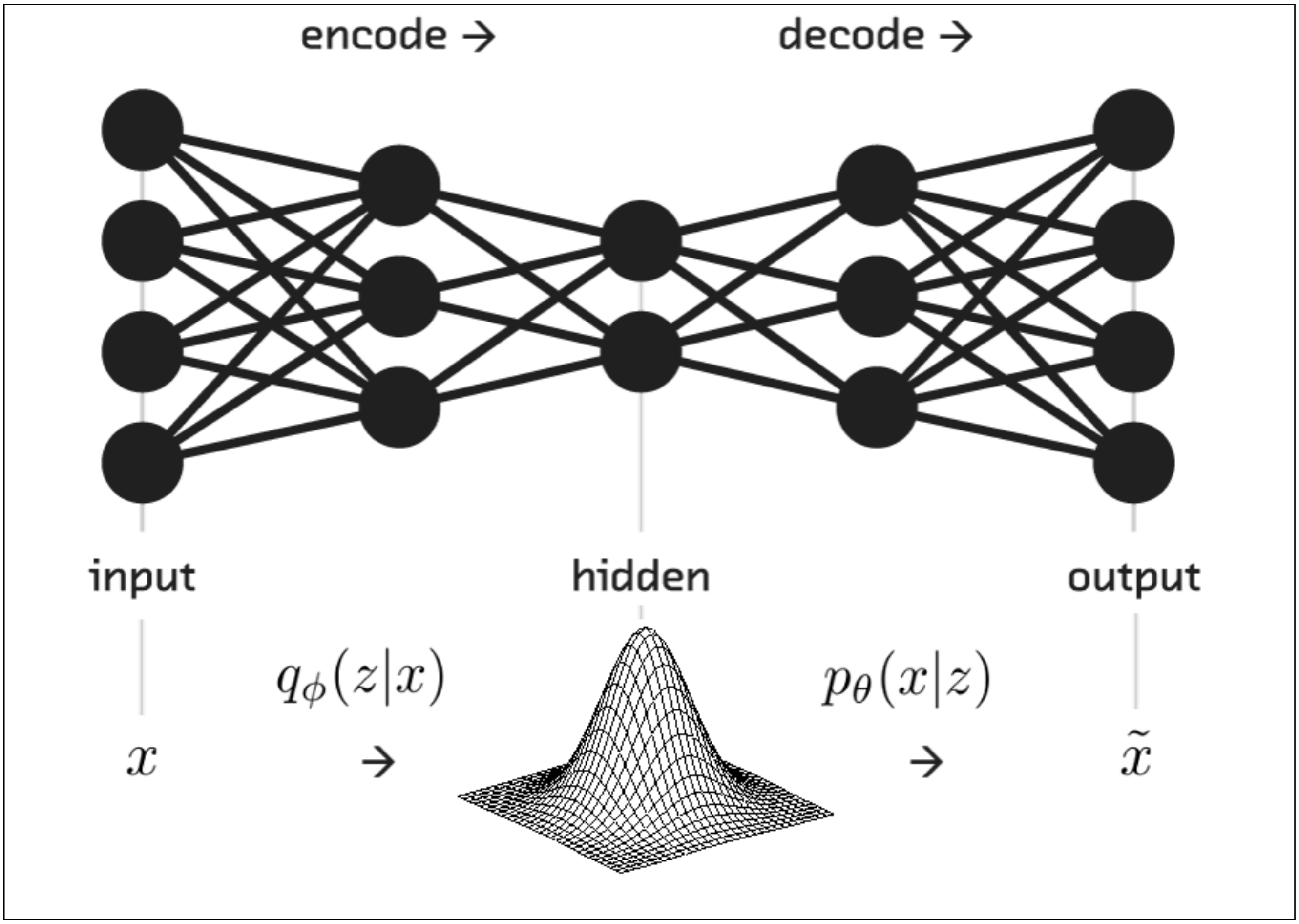
Machine Learning

Homework 2

Nathaniel Xu 徐星成 (2019280359)



1.1 Multinomial MLE

1.1 The Maximum Likelihood Estimator is calculated as...

$$\theta^{MLE} = \arg\max_{\theta} \underbrace{\log p(D|\theta)}_{\text{log likelihood}}$$

For this question $D = \{\underline{x}\}$

$\theta = \{\underline{\mu}\}$

$$\log p(\underline{x}|\underline{\mu}) = \log n! + \sum_i x_i \log \mu_i - \sum_i \log x_i!$$

We then solve for the optimal log-likelihood while considering the constraints $\sum_i \mu_i = 1$

The lagrangian function is

$$L = \log n! + \sum_i x_i \log \mu_i - \sum_i \log x_i! - \lambda\left(\sum_i \mu_i - 1\right)$$

$$\frac{\partial L}{\partial \mu_k} = x_k \cdot \frac{1}{\mu_k} - \lambda = 0 \Rightarrow \lambda = \frac{x_k}{\mu_k} \quad ①$$

Manipulating ① gives

$$\lambda \underbrace{\sum_{k=1}^{d} \mu_k}_{1} = \underbrace{\sum_{k=1}^{d} x_k}_{n} \Rightarrow \lambda = n \quad ②$$

Combining ① and ② gives

$$\boxed{\mu_k = \frac{x_k}{n}} \quad (\text{for } k = 1 \ldots d)$$

1.2 EM for Mixture of Multinomials

1.2 The question simply asks to derive the EM-Algorithm for a Multinomial Mixture Model.

↳ I will use my own notation for this derivation, then transfer the notation into the bag-of-words model.

CLUSTER PROBABILITIES

$$\left.\begin{aligned} p(\underline{x} \mid j) &\doteq M(\underline{x}; \underline{\mu}_j) = \binom{m}{x_1 \cdots x_d} \mu_1^{x_1} \cdots \mu_d^{x_d} \\ p(s=j) &\doteq \pi_j \\ \Rightarrow p(\underline{x}) &= \sum_j \pi_j M(\underline{x}; \underline{\mu}_j) \end{aligned}\right\}$$

$D = \{X\}$

$\theta = \{\pi, \mathcal{M}\}$

where

$$X = \begin{bmatrix} - & \underline{x}_1^T & - \\ & \vdots & \\ - & \underline{x}_N^T & - \end{bmatrix}$$

$$\mathcal{M} = \begin{bmatrix} - & \underline{\mu}_1^T & - \\ & \vdots & \\ - & \underline{\mu}_K^T & - \end{bmatrix}$$

CONSTRAINTS

$$\begin{cases} \sum_i x_i = m \\ \sum_i \mu_i = 1 \\ \sum_j \pi_j = 1 \end{cases}$$

NOTATION

n : index of data point 1..N

j : index of class/cluster 1..K

i : index of element in data point 1..d

LOG LIKELIHOOD

* We omitt the notation for conditional on θ for ease of readibility.

$$\begin{aligned} \log p(D \mid \theta) &= \log p(X) \\ &= \log \prod_n p(\underline{x}_n) = \sum_n \log p(\underline{x}_n) \\ &= \sum_n \log \left[\sum_j \pi_j M(\underline{x}_n; \underline{\mu}_j) \right] \end{aligned}$$

E-STEP

The E-Step suffices to compute the cluster responsibilities:

$$\gamma_{nk} \doteq p(s=k \mid \underline{x}_n) = \frac{p(\underline{x}_n \mid k) \cdot p(k)}{\sum_j p(\underline{x}_n \mid j) \cdot p(j)} \quad \text{[Bayes Rule]}$$

$$\Rightarrow \boxed{\gamma_{nk} = \frac{\pi_k M(\underline{x}_n; \underline{\mu}_k)}{\sum_j \pi_j M(\underline{x}_n; \underline{\mu}_j)}}$$

M-STEP

The M-Step involves maximizing the log likelihood w.r.t. the parameters θ.

- Maximize w.r.t. $\underline{\mu}_k$ s.t. $\sum_i \mu_{ki} = 1$

  ↳ The lagrangian optimization is...

$$L = \sum_n \log \sum_j \pi_j M(\underline{x}_n; \underline{\mu}_j) - \sum_j \lambda_j (\underline{\mu}_j^T \underline{1} - 1)$$

$$\frac{\partial L}{\partial \underline{\mu}_k} = \sum_n \left[\frac{1}{\sum_j \pi_j M(\underline{x}_n; \underline{\mu}_j)} \underbrace{\frac{\partial}{\partial \underline{\mu}_k} \left(\sum_j \pi_j M(\underline{x}_n; \underline{\mu}_j) \right)}_{\pi_k \frac{\underline{x}_n}{\underline{\mu}_k} M(\underline{x}_n; \underline{\mu}_k)} \right] - \frac{\partial}{\partial \underline{\mu}_k} \left(\lambda_k (\underline{\mu}_k^T \cdot \underline{1} - 1) \right)$$

A.1 (* the bar denotes element-wise division)

$$= \sum_n \left[\frac{\pi_k M(\underline{x}_n; \underline{\mu}_k)}{\sum_j \pi_j M(\underline{x}_n; \underline{\mu}_j)} \cdot \frac{\underline{x}_n}{\underline{\mu}_k} \right] - \lambda_k \cdot \underline{1}$$

$$= \sum_n \gamma_{nk} \cdot \frac{\underline{x}_n}{\underline{\mu}_k} - \lambda_k \cdot \underline{1} = 0$$

This system of linear equations can then be written as...

$$\begin{cases} \sum_n \gamma_{nk} \dfrac{x_{ni}}{\mu_{ki}} = \lambda_k & \text{(for all } i = 1 \ldots d) \\ \left.\begin{array}{l} \sum_i x_{ni} = m_n \\ \sum_i \mu_{ki} = 1 \end{array}\right\} & \text{constraints.} \end{cases}$$

Manipulating these equations produce an update for μ.

$$\Rightarrow \boxed{\mu_{ki} = \frac{\sum_n \gamma_{nk} \cdot x_{ni}}{\sum_n \gamma_{nk} \cdot m_n}} \quad \text{A.2}$$

- Maximize w.r.t. π_k s.t. $\sum_j \pi_j = 1$

↳ The lagrangian optimization is...

$$L = \sum_n \log \sum_j \pi_j M(\underline{x}_n; \underline{\mu}_j) - \lambda\left(\sum_j \pi_j - 1\right)$$

$$\frac{\partial L}{\partial \pi_k} = \sum_n \left[\frac{1}{\sum_j \pi_j M(\underline{x}_n; \underline{\mu}_j)} \frac{\partial}{\partial \pi_k}\left(\sum_j \pi_j M(\underline{x}_n; \underline{\mu}_j)\right)\right] - \lambda$$

$$= \sum_n \frac{M(\underline{x}_n; \underline{\mu}_k)}{\sum_j \pi_j M(\underline{x}_n; \underline{\mu}_j)} - \lambda$$

$$= \frac{1}{\pi_k} \sum_n \underbrace{\frac{\pi_k M(\underline{x}_n; \underline{\mu}_k)}{\sum_j \pi_j M(\underline{x}_n; \underline{\mu}_j)}}_{\gamma_{nk}} - \lambda \quad = \quad \frac{1}{\pi_k} \sum_n \gamma_{nk} - \lambda = 0$$

The system of linear equations can then be written as...

$$\begin{cases} \frac{1}{\pi_k} \sum_n \gamma_{nk} = \lambda \\ \sum_j \pi_j = 1 \end{cases}$$

Manipulating these equations produce an update for π

$$\Rightarrow \boxed{\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}} \qquad (A.3)$$

EM-ALGORITHM SUMMARY

E-STEP:

$$\gamma_{nk} = \frac{\pi_k M(\underline{x}_n; \underline{\mu}_k)}{\sum_j \pi_j M(\underline{x}_n; \underline{\mu}_j)}$$

M-STEP:

$$\mu_{ki} = \frac{\sum_n \gamma_{nk} \cdot x_{ni}}{\sum_n \gamma_{nk} \cdot m_n}$$

$$\pi_k = \frac{1}{N} \sum_n \gamma_{nk}$$

n: index of data point
1..N

j: index of class/cluster
1..K

i: index of element in data point
1..d

APPENDIX : INTERMEDIATE STEPS

(A.1) $$\frac{\partial}{\partial \underline{\mu}_k}\left(\sum_j \pi_j M(\underline{x}_n; \underline{\mu}_j)\right) = \pi_k \frac{\partial}{\partial \underline{\mu}_k} M(\underline{x}_n; \underline{\mu}_k)$$

$$\Rightarrow \left[\frac{\partial}{\partial \mu_{ki}} M(\underline{x}_n; \underline{\mu}_k)\right]_i = \left[\frac{\partial}{\partial \mu_{ki}} \binom{m_n}{x_{n1} \cdots x_{nd}} \mu_{k1}^{x_{n1}} \cdots \mu_{kd}^{x_{nd}}\right]$$

$$= \left[\binom{m_n}{x_{n1} \cdots x_{nd}} x_{ni} \mu_{ki}^{(x_{ni}-1)} \cdot \mu_{k1}^{x_{n1}} \cdots \mu_{kd}^{x_{nd}}\right] = \left[\frac{x_{ni}}{\mu_{ki}} \cdot \underbrace{\binom{m_n}{x_{n1} \cdots x_{nd}} \mu_{k1}^{x_{n1}} \cdots \mu_{kd}^{x_{nd}}}_{M(\underline{x}_n; \underline{\mu}_k)}\right]$$

$$= \left[\frac{x_{ni}}{\mu_{ki}}\right]_i \cdot M(\underline{x}_n; \underline{\mu}_k)$$

$$\Rightarrow \frac{\partial}{\partial \underline{\mu}_k}\left(\sum_j \pi_j M(\underline{x}_n; \underline{\mu}_j)\right) = \boxed{\pi_k \frac{\underline{x}_n}{\underline{\mu}_k} \cdot M(\underline{x}_n; \underline{\mu}_k)}$$

(A.2) $$\sum_n \gamma_{nk} \frac{x_{ni}}{\mu_{ki}} = \lambda_k \quad (\text{for all } i = 1..d)$$

$$\Rightarrow \sum_n \gamma_{nk} x_{ni} = \lambda_k \cdot \mu_{ki}$$ Taking the sum of all i on both sides give...

$$\sum_n \sum_i \gamma_{nk} x_{ni} = \lambda_k \sum \mu_{ki}$$

$$\sum_n \gamma_{nk} \underbrace{\sum_i x_{ni}}_{m_n} = \lambda_k \underbrace{\sum_i \mu_{ki}}_{1} \Rightarrow \lambda_k = \sum_n \gamma_{nk} m_n$$

Substituting back into the original equation gives us...

$$\Rightarrow \boxed{\mu_{ki} = \frac{\sum_n \gamma_{nk} \cdot x_{ni}}{\sum_n \gamma_{nk} \cdot m_n}}$$

(A.3) $$\frac{1}{\pi_k} \sum_n \gamma_{nk} = \lambda$$

$$\Rightarrow \sum_n \underbrace{\sum_k \gamma_{nk}}_{1} = \lambda \underbrace{\sum_k \pi_k}_{1} \Rightarrow \lambda = \sum_{n=1}^{N} 1 = N$$

substituting this back into the original equation gives us...

$$\Rightarrow \boxed{\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}}$$

EM-ALGORITHM for the BAG-OF-WORDS MODEL

We translate the notation to the Bag-of-Words model:

$$\begin{cases} \underline{T}_d \leftarrow \underline{x}_n & \text{where } \underline{T}_d \text{ corresponds to the row in table } T \text{ of the } d^{th} \text{ document} \\ n_d \leftarrow m_n & \\ \mu \leftarrow \mu^T & [\mu \text{ matricies are transposed}] \end{cases}$$

For indicies

$$\begin{cases} d \leftarrow n \\ w \leftarrow i \end{cases}$$

$$p(d \mid k) = p(\underline{T}_d \mid k) = M(\underline{T}_d ; \underline{\mu}_k)$$

$$\Rightarrow M(\underline{T}_d, \underline{\mu}_k) = \binom{n_d}{T_{d1} \cdots T_{dW}} \mu_{1k}^{T_{d1}} \cdots \mu_{Wk}^{T_{dW}}$$

E-STEP

$$\gamma_{dk} = \frac{\pi_k M(\underline{T}_d ; \underline{\mu}_k)}{\sum_{j=1}^{K} \pi_j M(\underline{T}_d ; \underline{\mu}_j)}$$

M-STEP

$$\mu_{wk} = \frac{\sum_{d=1}^{D} \gamma_{dk} \cdot T_{dw}}{\sum_{d=1}^{D} \gamma_{dk} \cdot n_d}$$

$$\pi_k = \frac{1}{D} \sum_{d=1}^{D} \gamma_{dk}$$

2.1 PCA: Minimum Error Formulation

2.1 PCA: MINIMUM ERROR FORMULATION

Another way we can think of PCA is as a way to minimize the error between a low dimension projection and the original data

- Let us introduce a complete orthonormal set of of D basis vectors from which we will project onto only M dimensions

$$\underline{u}_i^T \underline{u}_j = \delta_{ij} \quad (\text{for all } i,j = 1..D) \quad [\text{Orthogonal Basis}]$$

- Each data point can be expressed as a linear combination of the basis vectors

$$\underline{x}_n = \sum_{i=1}^{D} \alpha_{ni} \underline{u}_i = \sum_{i=1}^{D} \underbrace{(\underline{x}_n^T \underline{u}_i)}_{\alpha_{ni}} \underline{u}_i$$

[where $\alpha_{nj} = \underline{x}_n^T \underline{u}_j$ due to orthonormality of basis vectors]

- The projection of $\underline{x}_n$ onto a lower dimensional subspace (M-dim) can be approximated as

$$\check{\underline{x}}_n = \sum_{i=1}^{M} z_{ni} \underline{u}_i + \sum_{i=M+1}^{D} b_i \underline{u}_i$$

[z_{ni} is the value of that basis component $\underline{u}_i$ for $\underline{x}_n$

b_i is the constant for all $\underline{x}_n$]

The Minimum Error Formulation problem can now be described as solving for the variables $\underline{u}_i$, z_{ni}, b_i that minimize the MSE.

$$\underset{\underline{u}_i, z_{ni}, b_i}{\text{minimize}} \left\{ J = \frac{1}{N} \sum_{n=1}^{N} \| \underline{x}_n - \check{\underline{x}}_n \|^2 \right\}$$

The MSE can be expressed as...

$$J = \frac{1}{N} \sum_n \| \underline{x}_n - \check{\underline{x}}_n \|^2 = \frac{1}{N} \sum_n (\underline{x}_n - \check{\underline{x}}_n)^T (\underline{x}_n - \check{\underline{x}}_n)$$

$$= \frac{1}{N} \sum_n \left[\underline{x}_n^T \underline{x}_n - 2 \underline{x}_n^T \check{\underline{x}}_n + \check{\underline{x}}_n^T \check{\underline{x}}_n \right]$$

$$= \frac{1}{N} \sum_n \left[\underline{x}_n^T \underline{x}_n - 2 \underline{x}_n^T \left(\sum_1^M z_{ni} \underline{u}_i + \sum_{M+1}^D b_i \underline{u}_i \right) + \left(\sum_1^M z_{ni} \underline{u}_i^T + \sum_{M+1}^D b_i \underline{u}_i^T \right) \left(\sum_1^M z_{ni} \underline{u}_i + \sum_{M+1}^D b_i \underline{u}_i \right) \right]$$

MINIMIZE J w.r.t. z_{nj}

$$\frac{\partial J}{\partial z_{nj}} = \frac{1}{N}\left[-2\underline{x}_n^T \underline{u}_j + \underline{u}_j^T\left(\sum z_{ni}\underline{u}_i + \sum b_i\underline{u}_i\right) + \left(\sum z_{ni}\underline{u}_i^T + \sum b_i\underline{u}_i^T\right)\underline{u}_j\right]$$

$$= \frac{1}{N}\left[-2\underline{x}_n^T \underline{u}_j + 2\underline{u}_j^T\left(\sum z_{ni}\underline{u}_i + \sum b_i\underline{u}_i\right)\right]$$

$$\frac{\partial J}{\partial z_{nj}} = 0 \Rightarrow \quad \underline{x}_n^T \underline{u}_j = \underline{u}_j^T\left(\sum z_{ni}\underline{u}_i + \sum b_i\underline{u}_i\right)$$

$$= \sum_1^M z_{ni}\underline{u}_j^T\underline{u}_i + \sum_{M+1}^D b_i\underline{u}_j^T\underline{u}_i = \sum_1^M z_{ni}\delta_{ij} + \underbrace{\sum_{M+1}^D b_i\delta_{ij}}_{0}$$

$$= z_{nj} \quad (\text{Since } j \in [1..M])$$

$$\Rightarrow \boxed{z_{nj} = \underline{x}_n^T \underline{u}_j} \quad \text{for all } j = 1..M$$

MINIMIZE J w.r.t. b_j

$$\frac{\partial J}{\partial b_j} = \frac{1}{N}\sum_n\left[-2\underline{x}_n^T \underline{u}_j + 2\underline{u}_j^T\left(\sum z_{ni}\underline{u}_i + \sum b_i\underline{u}_i\right)\right]$$

$$= \frac{1}{N}\sum_n\left[-2\underline{x}_n^T \underline{u}_j + 2\left(\underbrace{\sum_1^M z_{ni}\delta_{ij}}_{0} + \sum_{M+1}^D b_i\delta_{ij}\right)\right]$$

$$= \frac{1}{N}\sum_n\left[-2\underline{x}_n^T \underline{u}_j + 2b_j\right] \quad (\text{Since } j \in [M+1..D])$$

$$\frac{\partial J}{\partial b_j} = 0 \Rightarrow \quad \sum_n b_j = \sum_n \underline{x}_n^T \underline{u}_j$$

$$N b_j = \left(\sum_n \underline{x}_n^T\right)\underline{u}_j \Rightarrow b_j = \left(\frac{1}{N}\sum_n \underline{x}_n\right)^T \underline{u}_j$$

$$\Rightarrow \boxed{b_j = \underline{\bar{x}}^T \underline{u}_j} \quad \text{for all } j = M+1..D$$

where $\underline{\bar{x}} = \frac{1}{N}\sum_{n=1}^N x_n$ (Sample Mean)

Using the expressions for the optimal values of z_{nj} and b_j we can rewrite the error formulation in terms of only the basis vectors ($\underline{u}_i$)

$$\underline{x}_n - \hat{\underline{x}}_n = \left(\sum_{1}^{D} \alpha_{ni}\underline{u}_i\right) - \left(\sum_{1}^{M} z_{ni}\underline{u}_i + \sum_{M+1}^{D} b_i\underline{u}_i\right)$$

$$= \sum_{1}^{D} (\underline{x}_n^T\underline{u}_i)\underline{u}_i - \sum_{1}^{M} (\underline{x}_n^T\underline{u}_i)\underline{u}_i - \sum_{M+1}^{D} (\bar{\underline{x}}^T\underline{u}_i)\underline{u}_i$$

$$= \sum_{M+1}^{D} (\underline{x}_n^T\underline{u}_i - \bar{\underline{x}}^T\underline{u}_i)\underline{u}_i$$

$$\|\underline{x}_n - \hat{\underline{x}}_n\|^2 = (\underline{x}_n - \hat{\underline{x}}_n)^T(\underline{x}_n - \hat{\underline{x}}_n)$$

$$= \left(\sum_i \underline{u}_i^T(\underline{u}_i^T\underline{x}_n - \underline{u}_i^T\bar{\underline{x}})\right)\left(\sum_j (\underline{x}_n^T\underline{u}_j - \bar{\underline{x}}^T\underline{u}_j)\underline{u}_j\right)$$

$$= \sum_i\sum_j \underbrace{\underline{u}_i^T\underline{u}_j}_{\delta_{ij}}(\underline{u}_i^T\underline{x}_n - \underline{u}_i^T\bar{\underline{x}})(\underline{x}_n^T\underline{u}_j - \bar{\underline{x}}^T\underline{u}_j)$$

$$= \sum_{i=M+1}^{D} (\underline{x}_n^T\underline{u}_i - \bar{\underline{x}}^T\underline{u}_i)^2$$

$$J = \frac{1}{N}\sum_n \|\underline{x}_n - \hat{\underline{x}}_n\|^2$$

$$= \frac{1}{N}\sum_n\sum_i (\underline{x}_n^T\underline{u}_i - \bar{\underline{x}}^T\underline{u}_i)^2 = \frac{1}{N}\sum_n\sum_i \underline{u}_i^T(\underline{x}_n - \bar{\underline{x}})(\underline{x}_n^T - \bar{\underline{x}})\underline{u}_i$$

$$= \sum_i\left[\underline{u}_i^T\left(\underbrace{\frac{1}{N}\sum_n(\underline{x}_n - \bar{\underline{x}})(\underline{x}_n - \bar{\underline{x}})^T}_{\text{Sample Covariance}}\right)\underline{u}_i\right]$$

$$\Rightarrow \boxed{J = \sum_{i=M+1}^{D} \underline{u}_i^T\Sigma\underline{u}_i}$$

Since we have an expression for the MSE in terms of the basis we can now solve the original optimization problem

$$\underset{\underline{u}_i}{\text{minimize}} \left\{ J = \sum_{i=M+1}^{D} \underline{u}_i^T \Sigma \underline{u}_i \right\}$$

Subject to $\underline{u}_i^T \underline{u}_i = 1$ (for all $i = M+1 \dots D$)

The lagrangian function is...

$$L = \sum_{i=M+1}^{D} \underline{u}_i^T \Sigma \underline{u}_i - \sum_{i=M+1}^{D} \lambda_i (\underline{u}_i^T \underline{u}_i - 1)$$

$$\frac{\partial L}{\partial \underline{u}_j} = 2\Sigma \underline{u}_j - 2\lambda_j \underline{u}_j = 0$$

[Recall that $\Sigma^T = \Sigma$ because the sample covariance is symmetric]

$$\Rightarrow \boxed{\Sigma \underline{u}_j = \lambda_j \underline{u}_j} \quad \text{for all } j = M+1 \dots D$$

This means that the optimal $\underline{u}_j$ vectors are eigenvectors of the sample covariance, and have eigenvalues λ_j

Now we substitute the optimal basis vectors into the MSE equation to get...

$$\boxed{J = \sum_{i=M+1}^{D} \lambda_i}$$

using the fact that $\underline{u}_j^T \Sigma \underline{u}_j = \lambda_j \underbrace{\underline{u}_j^T \underline{u}_j}_{1 \text{ by orthogonality}} = \lambda_j$

This means that to minimize the error formulation we select the non-principal subspace vectors to be the eigenvectors of Σ with the $(D-M)$ smallest eigenvectors.

$\Rightarrow$ Since the eigenvectors of Σ are all orthogonal (because Σ is a symmetric matrix), the eigenvectors defining the principal subspace are the remaining eigenvectors of Σ corresponding to the M largest eigenvalues. //

3.1 Reinforcment Learning

3.1 This question is a simple update on one iteration of the action-state value function $Q(s,a)$.

$\boxed{R(s,a)}$

a \ s	Non	Light	Heavy
No send	0	20	100
Send	-1	15	80

$\boxed{P(s'|s, a = \text{no send})}$

s \ s'	Non	Light	Heavy
Non	0.95	0.05	0.0
Light	0.2	0.75	0.05
Heavy	0.1	0.2	0.7

$\boxed{Q(s,a)}$

a \ s	Non	Light	Heavy
No Send	-1	10	50
Send	-2	35	40

$\boxed{P(s'|s, a = \text{send})}$

s \ s'	Non	Light	Heavy
Non	0.1	0.85	0.05
Light	0.1	0.2	0.7
Heavy	0.05	0.15	0.8

Notice that for this iteration

$$\begin{cases} \max_{a'} Q(s' = \text{Non}, a') = -1 \\ \max_{a'} Q(s' = \text{Light}, a') = 35 \\ \max_{a'} Q(s' = \text{Heavy}, a') = 50 \end{cases}$$

The update formula is:

$$Q(s,a) \leftarrow R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a')$$

S = Non, a = No-Send

$$Q(s,a) = 0 + 0.9(0.95(-1) + 0.05(35) + 0(50)) = 0.72$$

S = Light, a = No-Send

$$Q(s,a) = 20 + 0.9(0.2(-1) + 0.75(35) + 0.05(50)) = 45.695$$

S = Heavy, a = No-send

$$Q(s,a) = 100 + 0.9(0.1(-1) + 0.2(35) + 0.7(50)) = 137.71$$

S = Light, a = Send

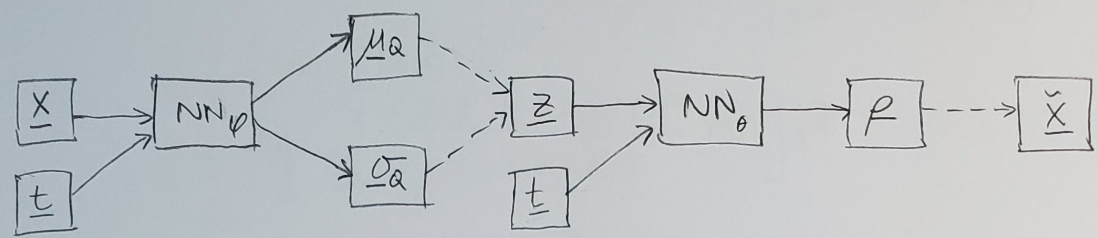
$$Q(s,a) = 15 + 0.9(0.1(-1) + 0.2(35) + 0.7(50)) = 52.71$$

S = Heavy, a = send

$$Q(s,a) = 80 + 0.9(0.05(-1) + 0.15(35) + 0.8(50)) = 120.68$$

4.1 CVAE: Model & Lower Bound

4.1 The conditional VAE model is described below



Encoder
$Q(\underline{z}|\underline{x},\underline{t})$

Decoder
$P(\underline{x}|\underline{z},\underline{t})$

The distributions of the model are..

$P(\underline{z}) = \mathcal{N}(\underline{z}; \mu_P, \Sigma_P)$ [Prior on $\underline{z}$]

$Q(\underline{z}|\underline{x},\underline{t}) = \mathcal{N}(\underline{z}; \mu_Q, \Sigma_Q)$ [$\underline{z} \sim Q(\underline{z}|\underline{x},\underline{t})$]

$P(x_i|\underline{z},\underline{t}) = B(x_i; \rho_i) = \rho_i^{x_i}(1-\rho_i)^{(1-x_i)}$ [$\underline{x} \sim P(\underline{x}|\underline{z},\underline{t})$]

$P(\underline{t}) = \frac{1}{10}$

<u>Log Likelihood</u>

$$\log P(\underline{x}|\underline{t}) = \mathbb{E}_{\underline{z}\sim Q(\underline{z}|\underline{x},\underline{t})}\left[\log P(\underline{x}|\underline{t})\right]$$ ($P(\underline{x}|\underline{t})$ independant of $\underline{z}$)

$$= \mathbb{E}_{\underline{z}}\left[\log \frac{P(\underline{z},\underline{x}|\underline{t})}{P(\underline{z}|\underline{x},\underline{t})}\right]$$ (Baye's Rule)

$$= \mathbb{E}_{\underline{z}}\left[\log \frac{P(\underline{z},\underline{x}|\underline{t})}{Q(\underline{z}|\underline{x},\underline{t})} \cdot \frac{Q(\underline{z}|\underline{x},\underline{t})}{P(\underline{z}|\underline{x},\underline{t})}\right]$$

$$= \underbrace{\mathbb{E}_{\underline{z}}\left[\log \frac{P(\underline{z},\underline{x}|\underline{t})}{Q(\underline{z}|\underline{x},\underline{t})}\right]}_{L(\varphi,\theta,\underline{x})} + \underbrace{\mathbb{E}_{\underline{z}}\left[\log \frac{Q(\underline{z}|\underline{x},\underline{t})}{P(\underline{z}|\underline{x},\underline{t})}\right]}_{D_{KL}\left[Q(\underline{z}|\underline{x},\underline{t}) \,\|\, P(\underline{z}|\underline{x},\underline{t})\right]}$$

$$\Rightarrow \log P(\underline{x}|\underline{t}) - \underbrace{D_{KL}\left[Q(\underline{z}|\underline{x},\underline{t}) \| P(\underline{z},\underline{x},\underline{t})\right]}_{\geq 0} = \underbrace{L(\varphi,\theta,\underline{x})}_{\text{variational lower bound}}$$

Variational Lower Bound

$$L(\varphi, \theta, \underline{x}) = \mathbb{E}_{\underline{z}}\left[\log \frac{P(\underline{z}, \underline{x} \mid \underline{t})}{Q(\underline{z} \mid \underline{x}, \underline{t})}\right]$$

$$= \mathbb{E}_{\underline{z}}\left[\log \frac{P(\underline{x} \mid \underline{z}, \underline{t}) \cdot P(\underline{z} \mid \underline{t})}{Q(\underline{z} \mid \underline{x}, \underline{t})}\right]$$

$$= \underbrace{\mathbb{E}_{\underline{z}}\left[\log \frac{P(\underline{z} \mid \underline{t})}{Q(\underline{z} \mid \underline{x}, \underline{t})}\right]}_{-D_{KL}[Q(\underline{z} \mid \underline{x}, \underline{t}) \,\|\, P(\underline{z} \mid \underline{t})]} + \underbrace{\mathbb{E}_{\underline{z}}\left[\log P(\underline{x} \mid \underline{z}, \underline{t})\right]}_{\text{Reconstruction Error}}$$

$$\mathbb{E}_{\underline{z}}\left[\log \frac{P(\underline{z} \mid \underline{t})}{Q(\underline{z} \mid \underline{x}, \underline{t})}\right] = \mathbb{E}_{\underline{z}}\left[\log P(\underline{z} \mid \underline{t}) - \log Q(\underline{z} \mid \underline{x}, \underline{t})\right]$$

$$= \mathbb{E}_{\underline{z}}\left[\log P(\underline{z}) - \log Q(\underline{z} \mid \underline{x}, \underline{t})\right] \qquad (\underline{z}, \underline{t} \text{ are } \underline{\text{independant}} \text{ RVs})$$

$$= \mathbb{E}_{\underline{z}}\Big[-\frac{1}{2}(\underline{z} - \mu_P)^T \Sigma_P^{-1} (\underline{z} - \mu_P) - \frac{d}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_P|$$
$$+ \frac{1}{2}(\underline{z} - \mu_Q)^T \Sigma_Q^{-1} (\underline{z} - \mu_Q) + \frac{d}{2}\log 2\pi + \frac{1}{2}\log|\Sigma_Q|\Big]$$

$$= \mathbb{E}_{\underline{z}}\left[\frac{1}{2}\log\frac{|\Sigma_Q|}{|\Sigma_P|} + \frac{1}{2}(\underline{z} - \mu_Q)^T \Sigma_Q^{-1} (\underline{z} - \mu_Q) - \frac{1}{2}(\underline{z} - \mu_P)^T \Sigma_P^{-1} (\underline{z} - \mu_P)\right]$$

$$= \frac{1}{2}\log\frac{|\Sigma_Q|}{|\Sigma_P|} + \frac{1}{2}\mathbb{E}_{\underline{z}}\left[(\underline{z} - \mu_Q)^T \Sigma_Q^{-1} (\underline{z} - \mu_Q)\right] - \frac{1}{2}\mathrm{Tr}\left\{\mathbb{E}_{\underline{z}}\left[(\underline{z} - \mu_P)(\underline{z} - \mu_P)^T\right]\Sigma_P^{-1}\right\}$$

$$= \frac{1}{2}\log\frac{|\Sigma_Q|}{|\Sigma_P|} - \frac{1}{2}\mathrm{Tr}\{I_J\} + \frac{1}{2}\mathrm{Tr}\{\Sigma_Q^{-1}\Sigma_P\} + \frac{1}{2}(\mu_Q - \mu_P)^T \Sigma_Q^{-1} (\mu_Q - \mu_P)$$

$$= \frac{1}{2}\left[\log\frac{|\Sigma_Q|}{|\Sigma_P|} - \underbrace{J}_{\text{dim of } \underline{z}} + \mathrm{Tr}\left[\Sigma_Q^{-1}\Sigma_P\right] + (\mu_Q - \mu_P)^T \Sigma_Q^{-1} (\mu_Q - \mu_P)\right]$$

$$\mathbb{E}_{\underline{z}}\left[\log P(\underline{x} \mid \underline{z}, \underline{t})\right] \simeq \frac{1}{L}\sum_{K=1}^{L}\left[\log P(\underline{x} \mid \underline{z}^{(K)}, \underline{t})\right]$$

K^{th} sample of $\underline{z}$

(Monte Carlo Method)

$$= \frac{1}{L}\sum_{K=1}^{L}\left[\log \prod_i P(x_i \mid \underline{z}^{(K)}, \underline{t})\right]$$

$$= \frac{1}{L}\sum_{K=1}^{L}\left[\sum_{i=1}^{d} \log P(x_i \mid \underline{z}^{(K)}, \underline{t})\right]$$

$$= \frac{1}{L}\sum_{K=1}^{L}\sum_{i=1}^{d} \log\left(\rho_i^{x_i}(1-\rho_i)^{(1-x_i)}\right)$$

$$= \frac{1}{L}\sum_{K=1}^{L}\sum_{i=1}^{d}\left[x_i \log \rho_i + (1-x_i)\log(1-\rho_i)\right]$$

$$\begin{cases} L: \#\text{samples of } \underline{z} \\ d: \text{dim of } \underline{x} \\ \rho_i = f_{NN}^{(i)}(\underline{z}^{(K)}) \end{cases}$$

where $\underline{z}^{(K)}$ is defined using the reparameterization trick

Loss Function

$$L(\varphi, \theta, \underline{x}) = \mathbb{E}_{\underline{z}}\left[\log \frac{P(\underline{z} \mid \underline{t})}{Q(\underline{z} \mid \underline{x}, \underline{t})}\right] + \mathbb{E}_{\underline{z}}\left[\log P(\underline{x} \mid \underline{z}, \underline{t})\right]$$

$$= \frac{1}{2}\left[\log \frac{|\Sigma_Q|}{|\Sigma_P|} - J + Tr\left[\Sigma_Q^{-1}\Sigma_P\right] + (\mu_Q - \mu_P)^T \Sigma_Q^{-1}(\mu_Q - \mu_P)\right] +$$

$$\frac{1}{L}\sum_{K=1}^{L}\sum_{i=1}^{d}\left[x_i \log \rho_i + (1-x_i)\log(1-\rho_i)\right]$$

For $\begin{pmatrix} \mu_P = \underline{0} \\ \Sigma_P = I \end{pmatrix}$ and $\left(\mu_Q = [\mu_1 \cdots \mu_J]^T,\ \Sigma_Q = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_J^2 \end{bmatrix}\right)$

$$\boxed{L(\varphi, \theta, \underline{x}) = \frac{1}{2}\sum_{i=1}^{J}\left[1 + \log \sigma_i^2 - \sigma_i^2 - \mu_i^2\right] + \frac{1}{L}\sum_{K=1}^{L}\sum_{i=1}^{d}\left[x_i \log \rho_i + (1-x_i)\log(1-\rho_i)\right]}$$

where $\begin{cases} J: \text{dim of } \underline{z} \\ d: \text{dim of } \underline{x} \\ L: \#\text{ of samples of } \underline{z} \\ \rho_i = f_{NN_\theta}^{(i)}(\underline{z}^{(K)}) \end{cases}$

and the reparameterization trick is used

$$\begin{cases} \underline{z}^{(K)} = \underline{\mu} + \underline{\sigma} \otimes \underline{\varepsilon}^{(K)} \\ \underline{\varepsilon}^{(K)} \sim \mathcal{N}(0, I) \end{cases}$$

element-wise multiplication

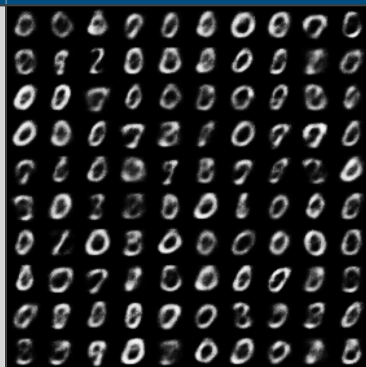
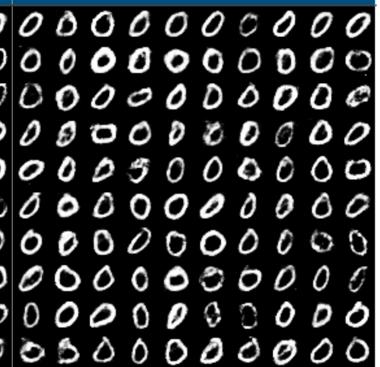
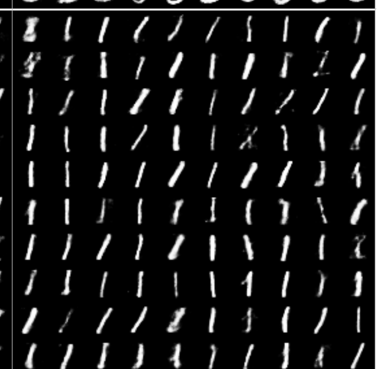
Training the CVAE

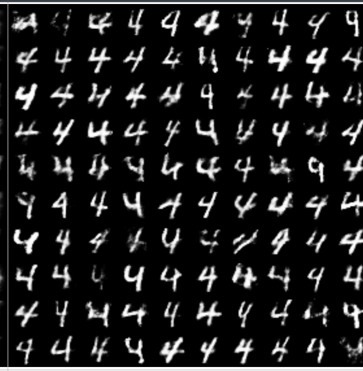
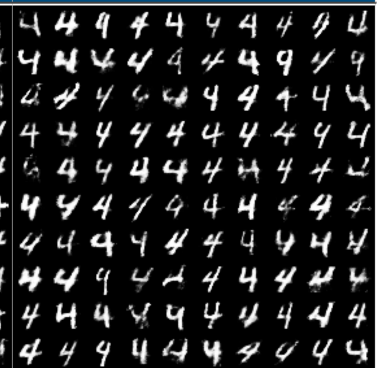
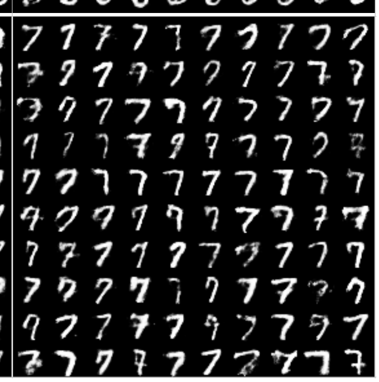
The CVAE model was created using Tensorflow and ZhuSuan. It was then trained on the MNIST training set which composes of digits (0-9) and their labels in one-hot-encoded form. The loss function being optimized is the variational lower bound while using the reparameterization trick to allow for backpropagation.

The model was trained for 1000 epochs. At each epoch the lower bound was calculated and recorded. At each 10th epoch 100 generated images of each label (digits 0-9) were generated using the CVAE model and saved. The below table shows the lower bound obtained for every 100th epoch.

Epoch	Lower Bound
0	−257.682586669922
100	−91.8138122558594
200	−85.1723480224609
300	−82.5363616943359
400	−81.0244827270508
500	−80.1718063354492
600	−79.1386871337891
700	−78.5410461425781
800	−78.1240768432617
900	−77.8921432495117
1000	−77.6972656250000

Generating Data

Generated Digit	Epoch 10	Epoch 500	Epoch 1000
0			
1			
2			
3			

Generated Digit	Epoch 10	Epoch 500	Epoch 1000
4			
5			
6			
7			

Generated Digit	Epoch 10	Epoch 500	Epoch 1000
8			
9