0 Instructions

Homework is due Tuesday, April 2, 2024 at 23:59pm Central Time. Please refer to `https://courses.grainger.illinois.edu/cs446/sp2024/homework/hw/index.html` for course policy on homeworks and submission instructions.

Reminder: Answers must be typeset. LaTeXand other methods of typesetting math are accepted.

1 PCA: 6pts

1. (1pts) Recall that PCA finds a direction w in which the projected data has highest variance by solving the following program:

$$\max_{w:||w||^2=1} w^T \Sigma w. \tag{1}$$

Here, Σ is a covariance matrix. You are given a dataset of two 2-dimensional points $(1,3)$ and $(4,7)$. Draw the two data points on the 2D plane. What is the first principal component w of this dataset?

Answer: $\frac{1}{5}[3,4]^T$

2. (3pts) Now you are given a dataset of four points $(2,0)$, $(2,2)$, $(6,0)$ and $(6,2)$. Given this dataset, derive the covariance matrix Σ in Eq.1. Then plot the centralized data with the first and the second principal components in one figure. **Include the plot in your written submission**. **Answer:**

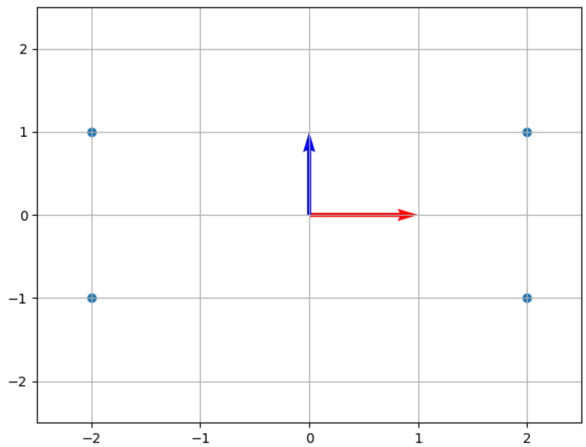
$$X = \begin{bmatrix} 2 & 2 & 6 & 6 \\ 0 & 2 & 0 & 2 \end{bmatrix}$$

First, center the data.

$$\bar{X} = \begin{bmatrix} -2 & -2 & 2 & 2 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

The covariance matrix Σ is a 2-by-2 matrix.

$$\Sigma = \frac{1}{4}\bar{X}\bar{X}^T = \frac{1}{4}\begin{bmatrix} 16 & 0 \\ 0 & 4 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$



3. (2pts) What is the optimal w and the optimal value of the program in Eq.1 given

$$\Sigma = \begin{bmatrix} 12 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}.$$

   Give your justification. **Answer:** By inspection, the eigenvector corresponding to the largest eigenvalue is:

$$[0, 0, 1, 0]^T,$$

   and the largest eigenvalue is 20. Therefore, $w^* = [0, 0, 1, 0]^T$ and the optimal value is 20.

2 Basics in Information Theory: 7pts

Let X be a discrete variable, and P, Q be two probability distributions over X. Define a new random variable X' as follows:

$$X' = \begin{cases} X \sim P & \text{if } B = 1, \\ X \sim Q & \text{if } B = 0, \end{cases}$$

where $B \in \{0, 1\}$ is an independent and Bernoulli distribution over $\{0, 1\}$ with the parameter λ, such that $\Pr(B = 1) = \lambda = 1 - \Pr(B = 0)$.

1. (2pts) Derive and represent the mixture distribution $\Pr(X' = x)$ in terms of $P(x)$ and $Q(x)$.

Answer:

$$\begin{aligned}\Pr(X'=x) &= \Pr(x\sim P, B=1)+\Pr(x\sim Q, B=0)\\ &= \Pr(x\sim P)\Pr(B=1)+\Pr(x\sim Q)\Pr(B=0)\\ &= \lambda P(x)+(1-\lambda)Q(x)\end{aligned}$$

2. (5pts) Show that $I(X';B)=D_\lambda(P\|Q)$, where $D_\lambda(P\|Q)$ is the λ-divergence between P and Q, i.e., $D_\lambda(P\|Q)=\lambda D_{\mathrm{KL}}(P\|\lambda P+(1-\lambda)Q)+(1-\lambda)D_{\mathrm{KL}}(Q\|\lambda P+(1-\lambda)Q)$. Note that by setting $\lambda=0.5$, the λ-divergence gives the Jensen-Shannon divergence.

   Answer:

$$\begin{aligned}I(X';B) =& \sum_x [\Pr(X'=x,B=1)\cdot\log\frac{\Pr(X'=x,B=1)}{\Pr(X'=x)\Pr(B=1)}\\ &+\Pr(X'=x,B=0)\cdot\log\frac{\Pr(X'=x,B=0)}{\Pr(X'=x)\Pr(B=0)}]\\ =& \sum_x[\lambda P(x)\cdot\log\frac{\lambda\cdot P(x)}{\Pr(X'=x)\cdot\lambda}+(1-\lambda)Q(x)\cdot\log\frac{(1-\lambda)\cdot Q(x)}{\Pr(X'=x)\cdot(1-\lambda)}]\\ =&\lambda\sum_x P(x)\log\frac{P(x)}{\lambda P(x)+(1-\lambda)Q(x)}\\ &+(1-\lambda)\sum_x Q(x)\log\frac{Q(x)}{\lambda P(x)+(1-\lambda)Q(x)}\\ =&\lambda D_{\mathrm{KL}}(P\|\lambda P+(1-\lambda)Q)+(1-\lambda)D_{\mathrm{KL}}(Q\|\lambda P+(1-\lambda)Q)\end{aligned}$$

3 k-Means with Soft Assignments: 10pts

Consider the following exemplar-based, hard-assignment form as the objective of k-Means for K clusters and n data points $x^{(i)}$ for $i=1,...,n$:

$$\min_{\mu_1,\dots,\mu_K}\sum_{i=1}^n\min_k\|x^{(i)}-\mu_k\|_2^2=\min_{\mu_1,\dots,\mu_K}\min_{\substack{A\in\{0,1\}^{n\times K}\\ A\cdot\mathbf{1}_K=\mathbf{1}_n}}\sum_{i=1}^n\sum_{k=1}^K A_{ik}\|x^{(i)}-\mu_k\|_2^2, \tag{2}$$

where μ_k denotes the center for the k-th cluster, the matrix $A\in\{0,1\}^{n\times K}$ indicates the hard assignment of each data point to the clusters, and $A\cdot\mathbf{1}_K=\mathbf{1}_n$, which tells us that each row of A has one 1 with all remaining elements as 0, i,e, $\sum_{k=1}^K A_{ik}=1,\forall i$.

We extend this setting to soft assignment by designing the matrix $A\in[0,1]^{n\times k}$, and the objective becomes:

$$\min_{\mu_1,\ldots,\mu_K} \min_{\substack{A\in[0,1]^{n\times K}\\ A\cdot\mathbf{1}_K=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{k=1}^{K} A_{ik}\|x^{(i)}-\mu_k\|_2^2. \tag{3}$$

1. (3pts) Show that the following holds:

$$\min_{\mu_1,\ldots,\mu_K} \min_{\substack{A\in[0,1]^{n\times K}\\ A\cdot\mathbf{1}_K=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{k=1}^{K} A_{ik}\|x^{(i)}-\mu_k\|_2^2 \le \min_{\mu_1,\ldots,\mu_K} \min_{\substack{A\in\{0,1\}^{n\times K}\\ A\cdot\mathbf{1}_K=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{k=1}^{K} A_{ik}\|x^{(i)}-\mu_k\|_2^2$$

   Hint: Note that $\{0,1\}^{n\times K}$ can be seen as a subset of $[0,1]^{n\times K}$.

   Answer: Denote $[0,1]^{n\times K}$ as U and $\{0,1\}^{n\times K}$ as V, it's obvious that $V\subseteq U$. Therefore, changing from the hard assignment to the soft assignment can be interpreted as the minimization over a larger set and $\min_{s\in U} g(s) \le \min_{s\in V} g(s)$ holds if $V\subseteq U$. So the above inequality holds.

2. (5pts) Show that the following also holds:

$$\min_{\mu_1,\ldots,\mu_K} \min_{\substack{A\in[0,1]^{n\times K}\\ A\cdot\mathbf{1}_K=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{k=1}^{K} A_{ik}\|x^{(i)}-\mu_k\|_2^2 \ge \min_{\mu_1,\ldots,\mu_K} \min_{\substack{A\in\{0,1\}^{n\times K}\\ A\cdot\mathbf{1}_K=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{k=1}^{K} A_{ik}\|x^{(i)}-\mu_k\|_2^2$$

   Hint: You may use the fact that $\|x^{(i)}-\mu_k\|_2^2 \ge \min_{l}\|x^{(i)}-\mu_l\|_2^2$, for any i and k.

Answer:

$$
\begin{aligned}
&\min_{\mu_1,\ldots,\mu_K} \min_{\substack{A\in[0,1]^{n\times K}\\ A\cdot\mathbf{1}_K=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{k=1}^{K} A_{ik}\|x^{(i)}-\mu_k\|_2^2 \\
\geq &\min_{\mu_1,\ldots,\mu_K} \min_{\substack{A\in[0,1]^{n\times K}\\ A\cdot\mathbf{1}_K=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{k=1}^{K} A_{ik}\cdot\min_l\|x^{(i)}-\mu_l\|_2^2 \\
= &\min_{\mu_1,\ldots,\mu_K} \min_{\substack{A\in[0,1]^{n\times K}\\ A\cdot\mathbf{1}_K=\mathbf{1}_n}} \sum_{i=1}^{n}\left(\sum_{k=1}^{K} A_{ik}\right)\cdot\min_l\|x^{(i)}-\mu_l\|_2^2 \\
= &\min_{\mu_1,\ldots,\mu_K} \sum_{i=1}^{n}\min_l\|x^{(i)}-\mu_l\|_2^2 \qquad (\text{by } \sum_{k=1}^{K} A_{ik}=1) \\
= &\min_{\mu_1,\ldots,\mu_K} \min_{\substack{A\in\{0,1\}^{n\times K}\\ A\cdot\mathbf{1}_K=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{k=1}^{K} A_{ik}\|x^{(i)}-\mu_k\|_2^2 \qquad (\text{by Eq. 2})
\end{aligned}
$$

3. (2pts) Show that the soft assignment problem introduced in this problem (Eq. 3) corresponds to a globally optimal hard assignment.

   Answer: Based on the two inequalities from the previous two problems, we can conclude that:

$$
\min_{\mu_1,\ldots,\mu_K} \min_{\substack{A\in[0,1]^{n\times K}\\ A\cdot\mathbf{1}_K=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{k=1}^{K} A_{ik}\|x^{(i)}-\mu_k\|_2^2 = \min_{\mu_1,\ldots,\mu_K} \min_{\substack{A\in\{0,1\}^{n\times K}\\ A\cdot\mathbf{1}_K=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{k=1}^{K} A_{ik}\|x^{(i)}-\mu_k\|_2^2
$$

   Therefore, the minimization with soft assignment leads to a globally optimal hard assignment.

4 Bernoulli Mixture Model: 18pts

Extended from the Gaussian mixture model introduced in the lecture, we explore the Bernoulli mixture model in this problem. We represent the dataset as $X \in \{0,1\}^{n\times d}$ and each data instance is a set of d independent binary random variables $x^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_d^{(i)}\}$ and the probability that $x^{(i)}$ is generated from the k-th Bernoulli distributions is calculated as:

$$
\Pr(x^{(i)}|\mu_k) = \prod_{j=1}^{d} \mu_k^{x_j^{(i)}} (1-\mu_k)^{\left(1-x_j^{(i)}\right)},
$$

where μ_k is the mean of the k-th Bernoulli distribution.

We consider K mixed Bernoulli distributions and introduce the auxiliary/latent variable $z_{ik} \in \{0,1\}$ with $\sum_{k=1}^{K} z_{ik} = 1 \; \forall i$ as the assignment for $x^{(i)}$ to the k-th Bernoulli distribution. Also, we have $\Pr(z_{ik}=1) = \pi_k$ and $\Pr(x^{(i)}|z_{ik}=1) = \Pr(x^{(i)}|\mu_k)$.

1. (5pts) Derive the log-likelihood $\log \Pr(x^{(i)}, z_i|\pi,\mu)$. **Answer:**

$$\begin{aligned}
\Pr(z_i|\pi) &= \prod_{k=1}^{K} \pi_k^{z_{ik}} \\
\Pr(x^{(i)}|z_i,\pi,\mu) &= \prod_{k=1}^{K} \Pr(x^{(i)}|\mu_k)^{z_{ik}} = \prod_{k=1}^{K}\left(\prod_{j=1}^{d} \mu_k^{x_j^{(i)}} (1-\mu_k)^{\left(1-x_j^{(i)}\right)}\right)^{z_{ik}} \\
\therefore \Pr(x^{(i)}, z_i|\pi,\mu) &= \Pr(x^{(i)}|z_i,\pi,\mu)\Pr(z_i|\pi,\mu) = \prod_{k=1}^{K}\left(\pi_k \prod_{j=1}^{d} \mu_k^{x_j^{(i)}} (1-\mu_k)^{\left(1-x_j^{(i)}\right)}\right)^{z_{ik}} \\
\therefore \log \Pr(x^{(i)}, z_i|\pi,\mu) &= \sum_{k=1}^{K} z_{ik}\left(\log \pi_k + \sum_{j=1}^{d} x_j^{(i)} \log \mu_k + \left(1-x_j^{(i)}\right)\log(1-\mu_k)\right)
\end{aligned}$$

2. (5pts) In the **expectation** step, derive the update step for the assignment (posterior) $z_{ik}^{\text{new}} = \Pr(z_{ik}=1|x^{(i)})$.

   Answer:

$$\begin{aligned}
z_{ik}^{\text{new}} = \Pr(z_{ik}=1|x^{(i)}) &= \frac{\Pr(z_{ik}=1)\Pr(x^{(i)}|z_{ik}=1)}{\sum_{\hat{k}=1}^{K}\Pr(z_{i\hat{k}}=1)\Pr(x^{(i)}|z_{i\hat{k}}=1)} \\
&= \frac{\pi_k \Pr(x^{(i)}|\mu_k)}{\sum_{\hat{k}=1}^{K}\pi_{\hat{k}}\Pr(x^{(i)}|\mu_{\hat{k}})} \\
&= \frac{\pi_k \prod_{j=1}^{d} \mu_k^{x_j^{(i)}} (1-\mu_k)^{\left(1-x_j^{(i)}\right)}}{\sum_{\hat{k}=1}^{K}\pi_{\hat{k}}\prod_{j=1}^{d} \mu_{\hat{k}}^{x_j^{(i)}} (1-\mu_{\hat{k}})^{\left(1-x_j^{(i)}\right)}}
\end{aligned}$$

3. (8pts) In the **maximization** step, derive the update step for the model parameter, i.e., μ_k^{new} and π_k^{new}.

   Answer:

$$\mathbb{E}[\log \Pr(X, Z|\pi, \mu)] = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik}^{\text{new}} \left(\log \pi_k + \sum_{j=1}^{d} x_j^{(i)} \log \mu_k + \left(1 - x_j^{(i)}\right) \log(1 - \mu_k) \right)$$

$$\begin{aligned}
\frac{\partial \mathbb{E}[\log \Pr(X, Z|\pi, \mu)]}{\partial \mu_k} &= \sum_{i=1}^{n} z_{ik}^{\text{new}} \left(\sum_{j=1}^{d} \frac{x_j^{(i)}}{\mu_k} - \frac{1 - x_j^{(i)}}{1 - \mu_k} \right) \\
&= \sum_{i=1}^{n} z_{ik}^{\text{new}} \left(\sum_{j=1}^{d} \frac{x_j^{(i)} - \mu_k}{\mu_k(1 - \mu_k)} \right) = 0 \\
\therefore \mu_k^{\text{new}} &= \frac{\sum_{i=1}^{n} \sum_{j=1}^{d} z_{ik}^{\text{new}} x_j^{(i)}}{d \sum_{i=1}^{n} z_{ik}^{\text{new}}}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathbb{E}[\log \Pr(X, Z|\pi, \mu)] + \lambda(\sum_{k=1}^{K} \pi_k - 1)}{\partial \pi_k} &= \frac{1}{\pi_k} \sum_{i=1}^{n} z_{ik}^{\text{new}} + \lambda = 0 \\
\therefore \pi_k &= -\frac{\sum_{i=1}^{N} z_{ik}^{\text{new}}}{\lambda}
\end{aligned}$$

Plug it into $\mathbb{E}[\log \Pr(X, Z|\pi, \mu)]$ to replace π_k and take the derivative w.r.t λ

$$\begin{aligned}
\frac{\partial \mathbb{E}[\log \Pr(X, Z|\pi, \mu)] + \lambda(\sum_{k=1}^{K} \pi_k - 1)}{\partial \lambda} &= -\frac{1}{\lambda} \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik}^{\text{new}} - 1 = 0 \\
\therefore \lambda &= -\sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik}^{\text{new}} \\
\therefore \pi_k^{\text{new}} &= \frac{\sum_{i=1}^{N} z_{ik}^{\text{new}}}{\sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik}^{\text{new}}}
\end{aligned}$$

5 Variational Autoencoder (VAE): 19pts

In this problem you will implement a Variational Autoencoder (VAE) to model points sampled from an unknown distribution. This will be done by constructing an encoder network and a decoder network. The encoder network $f_{\text{enc}} : X \subset \mathbb{R}^2 \to \mathbb{R}^h \times \mathbb{R}^h$ takes as input a point $\boldsymbol{x}$ from the input space and outputs parameters $(\boldsymbol{\mu}, \boldsymbol{\xi})$ where $\boldsymbol{\xi} = \log \boldsymbol{\sigma}^2$. The decoder network $f_{\text{dec}} : \mathbb{R}^h \to \mathbb{R}^2$ takes as input a latent vector $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ and outputs an element $\hat{\boldsymbol{x}} \in \mathbb{R}^2$ that we would hope is similar to members of the input space X. You will train this model by minimizing the (regularized) empirical risk

$$\widehat{\mathcal{R}}_{\mathrm{VAE}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(\hat{\boldsymbol{x}}_i, \boldsymbol{x}_i) + \lambda \mathrm{KL}\left(\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}_i), \exp(\boldsymbol{\xi}(\boldsymbol{x}_i)/2)), \mathcal{N}(0, I)\right).$$

Particularly, we have

$$\mathrm{KL}\left(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \mathcal{N}(0, I)\right) = -\frac{1}{2}\left[h + \sum_{j=1}^{h}\left(\log \sigma_j^2 - \mu_j^2 - \sigma_j^2\right)\right],$$

1. (3pts) Use the empirical risk discussed above to implement a VAE in the class `VAE`. Use ReLU activations between each layer, except on the last layer of the decoder use sigmoid. Use the Adam optimizer to optimize in the `step()` function. Make use of the PyTorch library for this. Use `torch.optim.Adam()`, there is no need to implement it yourself. Please refer to the docstrings in `hw4.py` for more implementation details.

2. (5pts) Implement the `fit` function using the `step()` function from the `VAE` class. See the docstrings in `hw4.py` for more details.

3. (11pts) Fit a `VAE` on the data generated by `generate_data` in `hw4_utils.py`. Use a learning rate $\eta = 0.01$, latent space dimension $h = 6$, KL-divergence scaling factor $\lambda = 5 \times 10^{-5}$, and train for 8000 iterations. Use least squares as the loss, that is, let $\ell(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2$. **Include separate plots of each of the following with a legend in your written submission**:

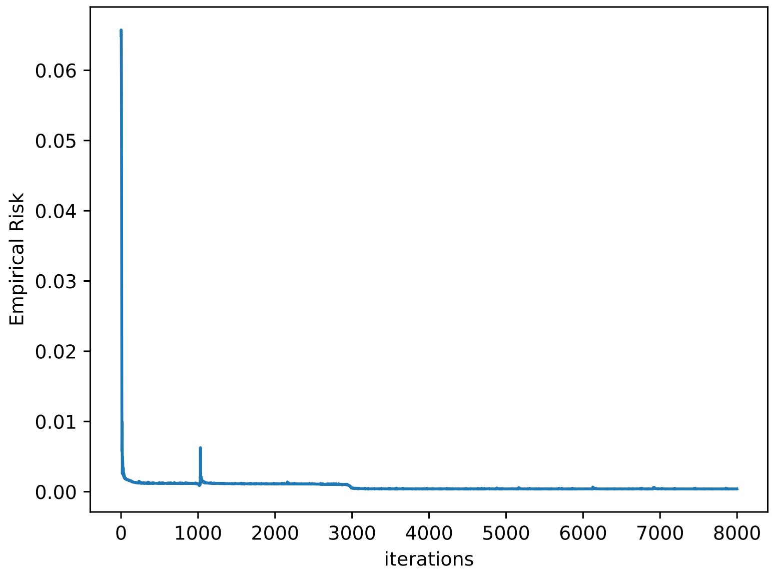
   (a) Your empirical risk $\widehat{\mathcal{R}}_{\mathrm{VAE}}$ on the training data vs iteration count;

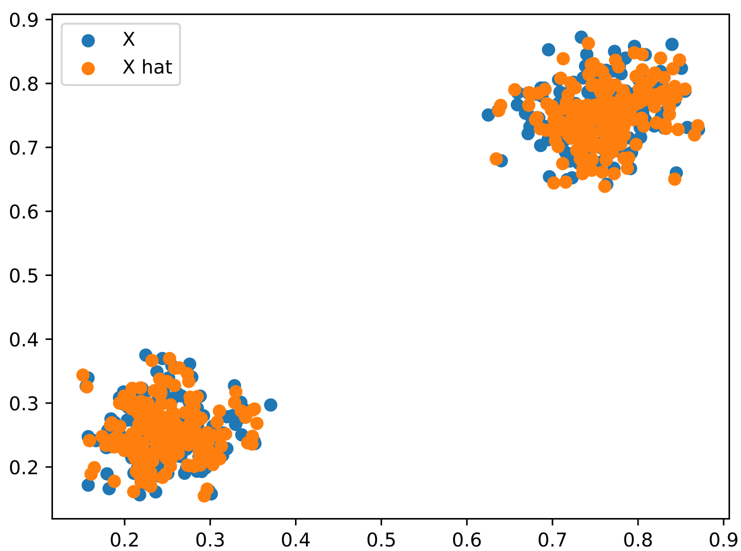
   (b) The data points $(\boldsymbol{x})_{i=1}^{n}$ along with their encoded and decoded approximations $\hat{\boldsymbol{x}}$;

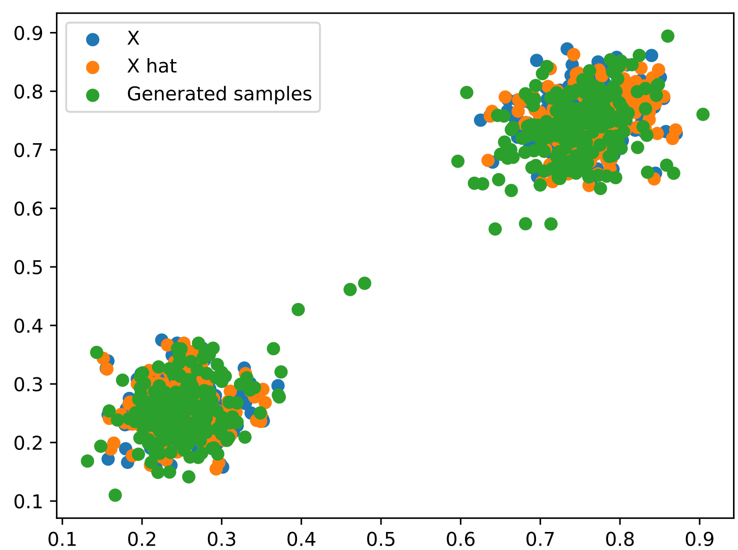
   (c) The data points $(\boldsymbol{x})_{i=1}^{n}$ along with their encoded and decoded approximations $\hat{\boldsymbol{x}}$, and n generated points $f_{\mathrm{dec}}(\boldsymbol{z})$ where $\boldsymbol{z} \sim \mathcal{N}(0, I)$.

   After you are done training, save your neural network to a checkpoint file using `torch.save(model.cpu().state_dict(), "vae.pb")`. **You will submit this checkpoint file `"vae.pb"` to the autograder with your code submission.**

   Answer:


Empirical Risk
iterations
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
1000
2000
3000
4000
5000
6000
7000
8000


X
X hat
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


X
X hat
Generated samples
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9