IT2023 Lecture Diary (Oct 27, 2023)

1 Generative Adversarial Network

Generative adversarial network is a kind of generative model. In the last few lectures, I have introduced its basic working principle has introduced and mentioned that the *factor analysis* model is a generative model.

1.1 Generative model

Basically, a generative model is a multiple-input-multiple-output (MIMO) model. The input is a multi-dimension random vector $\mathbf{z}$. Precisely, the input variables are called the hidden variables or the latent variables.

1.2 Fitting the sample distribution

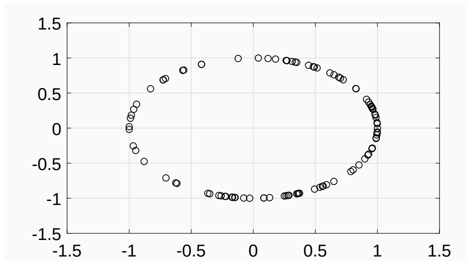
If the probability distribution $P(\mathbf{z})$ is known, the problem is to find a generative model $G(\mathbf{z}, \mathbf{w})$ its output vector distribution $P(G(\mathbf{z}, \mathbf{w}))$ conforms to the sample distribution, i.e. $P(\mathbf{x})$. In other words, the problem is find a generative model $G(\mathbf{z}, \mathbf{w})$ fitting the samples.

1.3 In search of the best-fit model

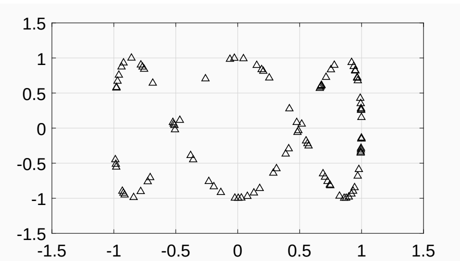
Clearly, in search of a generative model fitting the samples is a difficult problem. Let say, the dimension of the sample vector $\mathbf{x}$ is five. The number of possible classes of generative models is infinite. It could be a class of 1-input-5-output models. It could be a class of 2-input-5-output models. It could be a class of 3-input-5-output models and so on. For each class of generative models, we need to search for the model parameters so that the model fits well to the samples.

In these regards, the problems incurred in searching for a generative model are at least four folds.

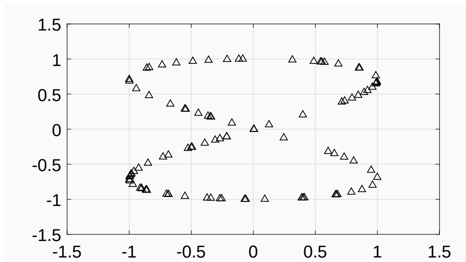
1. The number of random variables (i.e. hidden factors) in $\mathbf{z}$.
2. The probability distribution of the random input vector $\mathbf{z}$, i.e. $P(\mathbf{z})$.
3. The class of generative models. Equivalently, the structure of the class of generative models.
4. The parameters of a generative model in a class of models.
5. The search algorithm for the parameters of a generative model which fits well the samples.



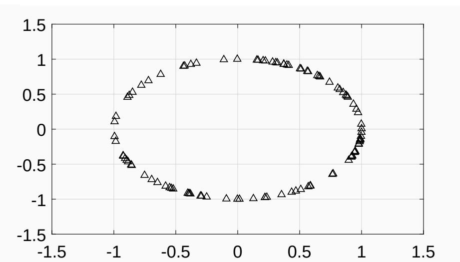
(a) Samples.



(b) Generative model I.



(c) Generative model II.



(d) Generative model III.

Figure 1: (a) Sampled data. (b) Data generated by the model $x_1 = \sin(z), x_2 = \cos(3z)$. (c) Data generated by the model $x_1 = \sin(2z), x_2 = \cos(z)$. (d) Data generated by the model $x_1 = \sin(z), x_2 = \cos(z)$. Here, z is a random variable uniformly distributed on $[0, 10]$.

2 Revisit Factor Analysis

Factor analysis is clearly a class of generative models. If you have background on *factor analysis* model, you should recall that the probability distribution of the random input vector $P(\mathbf{z})$ is assumed to be a Gaussian distribution (multivariate normal distribution).

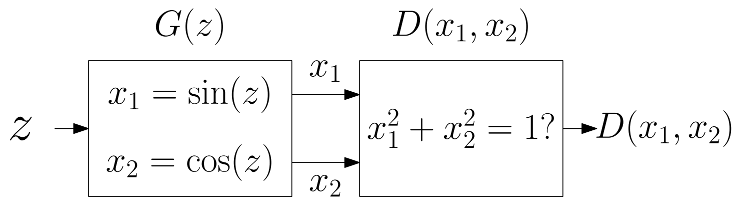
The algorithm for searching the model parameters is either based on *maximum likelihood* (ML) method, the method of *least square* (LS) or the method of *generalized least square* (GLS). Students usually underlook or even ignore the other two problems. If you did apply *factor analysis* in a management research, you should be aware that the results obtained are far from complete.

It should be noted that the Gaussian distribution is a distribution for continuous variable. Hence, an implicit assumption on the input vector is that the input variables are all continuous variables. Another implicit assumption is on the samples. As the model input variables are assumed to be continuous variables, the output variables must be continuous variables as well. In return, another implicit assumption is that the samples are sampled from a population with a probability density function of a continuous variables.

Question: (i) What if $P(\mathbf{z})$ is not a Gaussian distribution? (ii) What if $\mathbf{z}$ is a binary vector? (iii) Would SPSS or SAS has algorithm for searching the parameters for a *factor analysis* model with binary input vectors and discrete output vector?

3 Generative Model Fitting Data on Circular Samples

Figure 1 shows the shapes of data generated by three generative models. By *visual inspection*, Model III is the best-fit model among all three models. Human visual inspection is certainly not the only method (and not the best method) to determine is a model is good-fit for the samples if the distribution of the samples (resp. data being generated) is hardly visualized.



$$D(x_1, x_2) = \exp\left(-100\left(1 - \sqrt{x_1^2 + x_2^2}\right)^2\right)$$

Figure 2: A generative adversarial network for generating random samples on a circle.

Figure 2 shows a generative adversarial network in which the human inspector is replaced by a discriminator. In a visual inspection, a human determines if the shapes of the distributions of the samples and the generated data are the same. Instead of inspecting the shapes, the discriminator inspects if its input data (no matter from the generator or from the samples) is real. If the input data is real, its output is one. Otherwise, its output is zero.

3.1 With known information on the shape of the samples

By visual inspection on Figure 1a, we can assume that the samples are distributed on a circle with radius one. Therefore, one could define the class of discriminators as a two-input-one-output model with one parameter, i.e. $y(x_1, x_2) = D(x_1, x_2, \alpha)$, where the value of D is in the range $[0, 1]$.

$$D(x_1, x_2, \alpha) = \exp\left(-\alpha\left(1 - \sqrt{x_1^2 + x_2^2}\right)^2\right), \tag{1}$$

where α is the model parameter for the discriminator. The class of generative models could thus be defined as a single-input-two-output model, i.e. $\mathbf{x}(z) = G(z, a_1, a_2)$, where a_1, a_2 are the model parameters.

$$G(z, a_1, a_2) = \begin{bmatrix} x_1(z) \\ x_2(z) \end{bmatrix} = \begin{bmatrix} \sin(a_1 z) \\ \cos(a_2 z) \end{bmatrix}, \tag{2}$$

where the probability distribution $P(z) = 1/10$ for $z \in [0, 10]$. For this GAN, three parameters, respectively α, a_1 and a_2, are needed to be determined. Figure 2 shows a possible model, in which $\alpha = 100$, $a_1 = a_2 = 1$.

In accordance with the above example, we let $\mathbf{w}_G = (a_1, a_2)$ and $\mathbf{w}_D = \alpha$. Some interesting questions could be raised.

Questions: (i) What if the shape of the sample distribution is complicated? (ii) What if the class of generative models is defined as three-input-two-output model? (iii) Would there be other possible classes for the generative model $G(\mathbf{z}, \mathbf{w}_G)$?

3.2 Different $P(z)$

For the generative model, one important factor if the distribution of the random variable z. In the above example, it is assumed that $P(z)$ is a uniform distribution on $[0, 10]$. Figure 3 shows the shapes of the data being generated by same generative model, $x_1 = \sin(z)$ and $x_2 = \cos(z)$, but their distributions are not the same as the $P(z)$ for the generative models in Figure 1.

- Figure 1(a) shows the case that z is uniformly distributed on $[0, \pi]$, i.e. $P(z) = 1/\pi$ for $\pi \in [0, \pi]$. The data generated are located on the right half circle only.

- Figure 1(b) shows the case that z is uniformly distributed on $[\pi, 2\pi]$, i.e. $P(z) = 1/\pi$ for $\pi \in [\pi, 2\pi]$. The data generated are located on the left half circle only.
- Figure 1(c) shows the case that the distribution of z is a Beta distribution in $[-\pi, \pi]$. The data generated are located densely on the upper half circle.
- Figure 1(d) shows the case that the distribution of z is a Beta distribution in $[0, 2\pi]$. The data generated are located densely on the lower half circle.

3.3 Arbitrary $P(z)$

From the above observations, one should be aware that the definition (equivalently, the mode structure) of a class of generative models relies on $P(z)$. If $P(z)$ is **arbitrary**, complicated model structure should be defined. However, in search of a suitable complicated model structure is yet another difficult problem.

4 Important Property of the Generator

Given a set of samples (high dimension), say $\mathbf{x}_1, \cdots, \mathbf{x}_N$, and the dimension of each sample is n. Suppose that the generator is defined as a multiple-input-multiple-output model with m inputs and n outputs. If the generator is successfully obtained, the following properties will be ensured.

1. The probability distribution of the data generated by the generator is similar to the distribution of the samples.
2. Each sample $\mathbf{x}_k$ can be mapped to one and only one input $\mathbf{z}_k$.
3. If $\mathbf{x}_i$ and $\mathbf{x}_j$ are close to each other, their corresponding inputs $\mathbf{z}_i$ and $\mathbf{z}_j$ must be close to each others.

Note that m is usually much smaller than n. Like the example presented in the previous sections, $m = 1$ and $n = 2$. The number in the range of z controls the outputs x_1 and x_2.

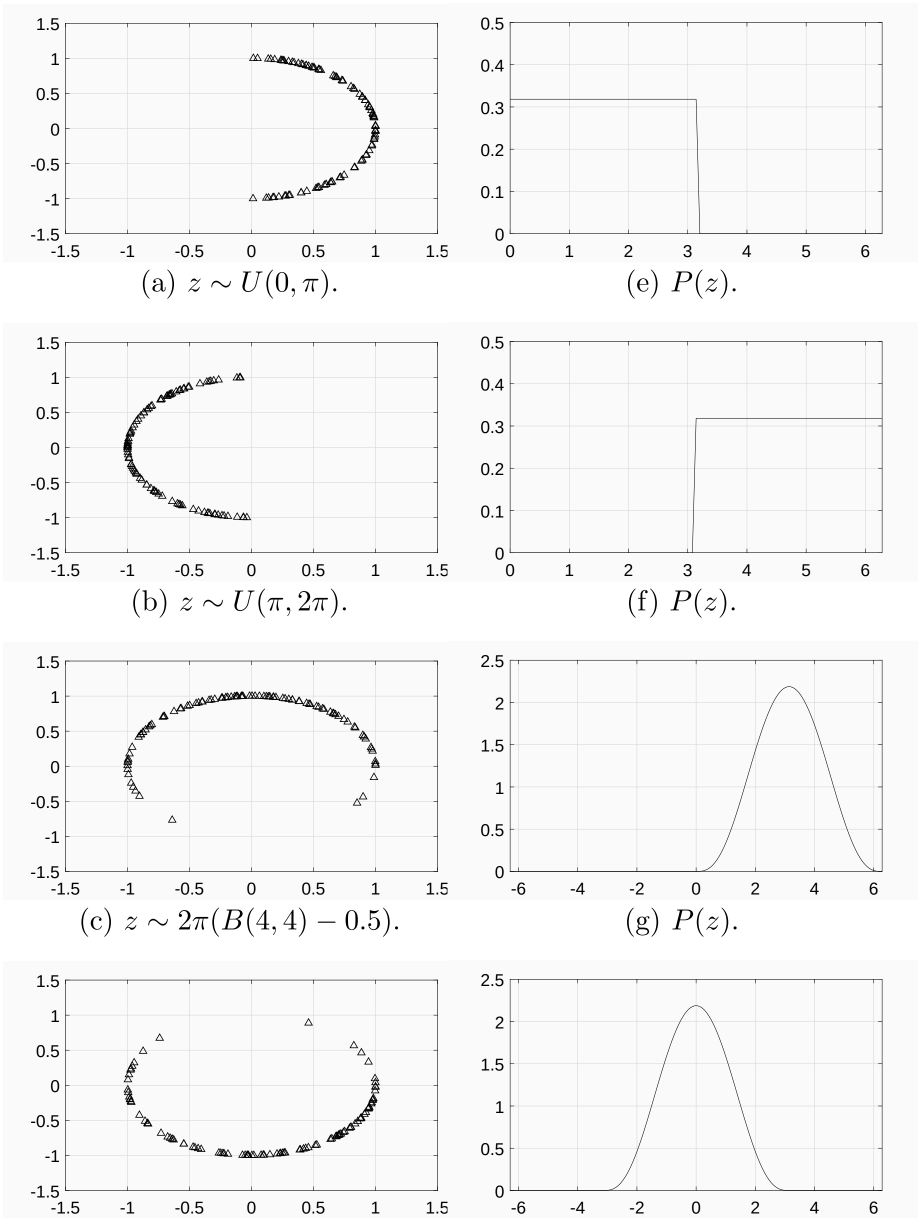
Once the generator has been obtained, each sample (x_1, x_2) can thus been mapped to a particular z. Clearly, if (x_1, x_2) and (x'_1, x'_2) are close to each other, their corresponding z and z' must be close to each other. This property of the generator is so-called *topographic preservation*. That is to say, the samples which are close to each other in the n dimensional space their corresponding m dimensional representatives are also close to each other in the low dimension m space, see Figure 4.

5 Generative Model for Images

For a set of image samples, each image is of very high dimension. For a black-and-white image, the size is of 28×28 (tiny image) or even 1024×1024 (natural image). For a color image, the dimension is even higher, $(3 \times 28) \times (3 \times 28) = 7056$ and $(3 \times 1024) \times (3 \times 1024) = 9437184$. With such high dimension, the probability distribution of the image samples $P(\mathbf{x})$ unlikely follows a simple Gaussian distribution. The distribution $P(\mathbf{x})$ should be of complicated shape.

5.1 Number of hidden random variables

The generative model should be of a large number of inputs (hundred and even thousand) and complicated model structure (equivalently, a model with very large number of parameters up to hundreds of billion). No matter what, the ultimate purpose of the model to be obtained is to generate a set of images its distribution similar to that of the distribution of the image samples, see Figure 5.



(a) $z \sim U(0, \pi)$. (e) $P(z)$.

(b) $z \sim U(\pi, 2\pi)$. (f) $P(z)$.

(c) $z \sim 2\pi(B(4, 4) - 0.5)$. (g) $P(z)$.

(d) $z \sim 2\pi B(4, 4)$. (h) $P(z)$.

Figure 3: The data generated by the generative model, $x_1(z) = \sin(z)$ and $x_2 = \cos(z)$, as shown in Figure 1(d) but with different distributions of $P(z)$. (a) $P(z)$ is a uniform distribution on $[0, \pi]$. (b) $P(z)$ is a uniform distribution on $[\pi, 2\pi]$. (c) $P(z)$ is a Beta distribution on $[-\pi, \pi]$. (d) $P(z)$ is a Beta distribution on $[0, 2\pi]$. The corresponding distributions of $P(z)$ are shown in (e), (f), (g), and (h) on the right column.

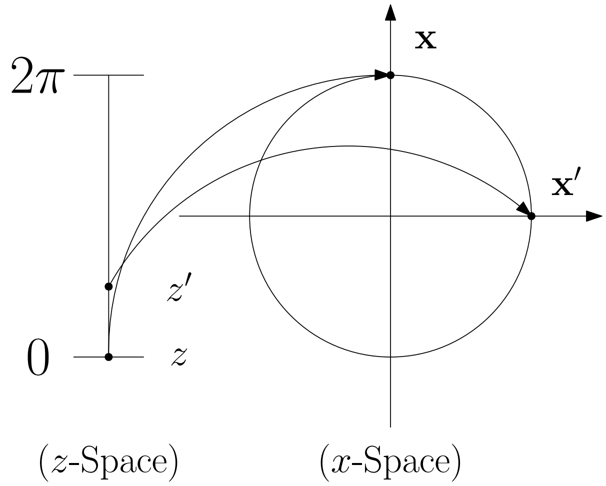


Figure 4: Illustration of the property of topographic preservation.

5.2 Fake image generation

Once the generative model has been obtained, one can investigate the change of each of the input, or a combination of inputs, on the visual change of an image. If it is succeeded, the generative model could eventually be applied to generate fake images. Clearly, it is yet another difficult problem for a researcher.

5.3 Number of training images

For a color image with size 1024×1024, the dimension of an image vector is 9437184. With such very high dimension, one problem encountered is on the number of training images to be ready for obtaining the generative model. Usually, the number of training data should be at least ten (or even hundred) times the dimension of a data. Thus, the number of training images should be at least ten (or even hundred) times the dimension of an image data, i.e. $95M$ to $950M$ images. Collecting such a large image dataset is yet another problem in building such a generative model.

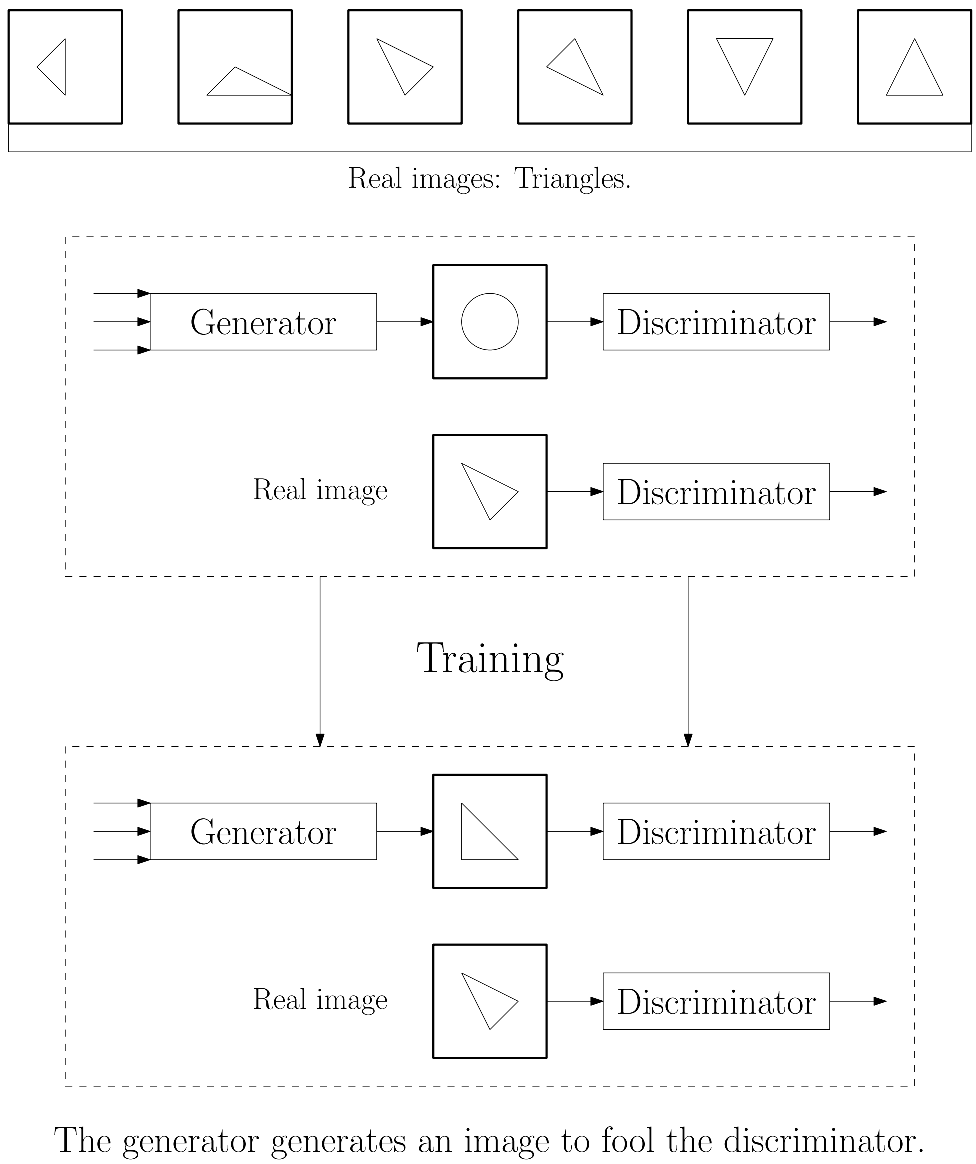


Figure 5: Idea behind a generative adversarial network and the goal of training.