Application of a counterfactual algorithm in a Variational Autoencoder

Giovani de Almeida Valdrighi

FGV EMAp

July 16, 2020

Outline

Introduction

- The task of learning relations between X and Y is a advanced area with efficient methods. However it may led us to mistakes when we apply it in scenarios where more information is necessary than the correlation between variables. There is necessary a study of causal relations.
- Judea Pearl, winning of Turing Award in 2011, is one of the biggest names encouraging the study of causality. He argues that "To Build Truly Intelligent Machines, Teach Them Cause and Effect".

Introduction

- According to Judea Pearl, there is three steps of knowledge: seeing, doing and imagining. **Seeing** refers to the ability to learn associations between variables. **Doing** refers to the ability of predicting outcomes of your actions on the enviroment. **Imagining** refers to the ability to know what could have been if the event ocurred in a different way.

Introduction

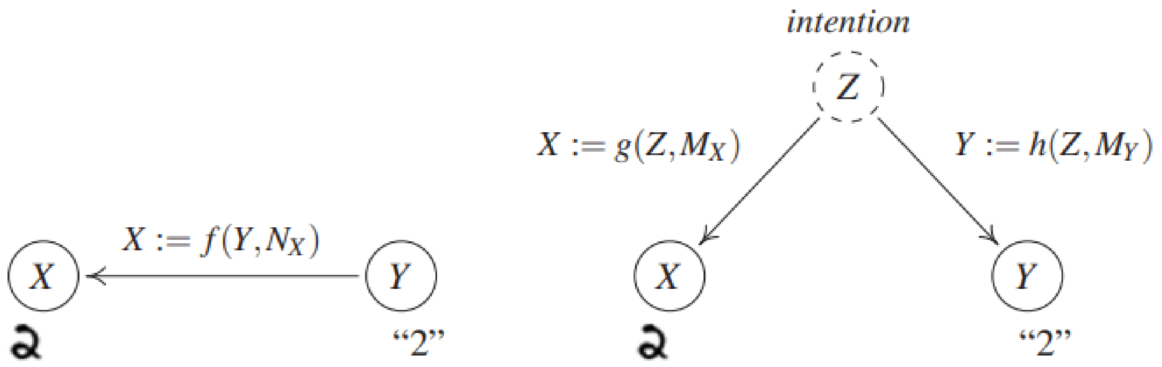

Figure 1: Two possible causal structures for the MNIST dataset that would generate the same observed data. X is the written number, Y is the number e Z is the writter intention.

Without a causal structure, it's possible to predict Y from X using observed data. However, to answers questions about interventions and counterfactuais, the causal structure is required.

Background

Structural causal model

A *structural causal model* model $\mathfrak{C} := (S, P_N)$ consists of a collection S of d structural assignments:

$$X_i := f_i(PA_i, N_i), \qquad i = 1, \dots, d$$

Where $PA_i \subseteq X_1, \dots, X_d \backslash X_i$ is called the parents of X_i and P_N is a joint distribution over the noise variables $N_1, \dots, N_d$.

Background

Intervention distribution

Consider an SCM $\mathfrak{C} := (S, P_N)$ and $P_X^{\mathfrak{C}}$. When there is an intervention in X_k replacing:

$$X_k := g(PA_k, N_k)$$

g can be a constant function. It creates a new SCM $\mathfrak{C}'$ that entails the *intervention distribution*, it is denoted by

$$P_X^{\mathfrak{C}'} = P_X^{\mathfrak{C};do(X_k := g(PA_k, N_k))}$$

Background

Counterfactual

Consider an SCM $\mathfrak{C} := (S, P_N)$ and $P_X^{\mathfrak{C}}$. For some observation x we define the *counterfactual* SCM by replacing the noise distribution

$$\mathfrak{C}_{X=x} = (S, P_N^{\mathfrak{C}|X=x})$$

The counterfactual statements are the invertion statements in the counterfactual SCM above.

Algorithm 1 Counterfactual inference on SCM

```
    Input
Structural causal model $\mathfrak{C}$
Observed variables $X = x$
Counterfactual variables $X = \overline{x}$
Number of samples n
    Output
n samples from $P^{\mathfrak{C};X=x;do(X=\overline{x})}$
1: #create observed and intervenioned models
2: obsModel ← $Condition(\mathfrak{C}, X = x)$
3: intModel ← $Do(\mathfrak{C}, X = \overline{x})$
4: #infer the noise distribution from observantion model
5: noisePost ← $Infer(P_N, model =$ obsModel$)$
6: # draw samples from the internvention model with the new
   noise dsitribution
7: for i in i:n do
8:     samples[i] ← intModel($noisePost$)
   return samples
```

Experiment

- To analyse the effiencity of the method, we need to apply it in a model that we know what is the real counterfactual result. For this taks, I used a dataframe from Deepmind called dSprites.
- dSprites
    - 737280 total images of 64×64
    - with latents factors:
    - shape: square, ellipse, heart
    - scale: 6 values in $[0.5, 1]$
    - orientation: 40 values in $[0, 2\pi]$
    - position X: 32 values in $[0, 1]$
    - position Y: 32 values in $[0, 1]$

Experiment

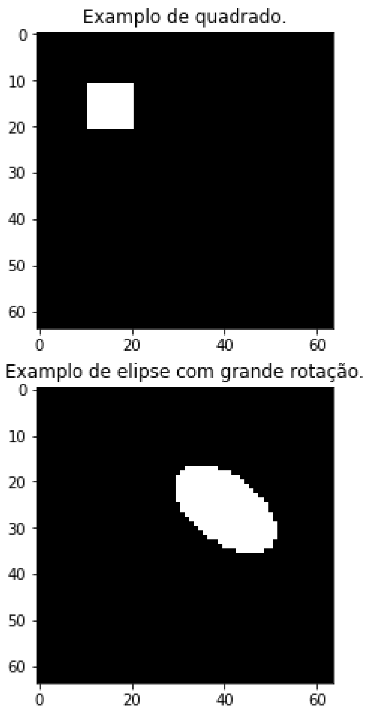

Figure 2: Example of images from the dataset, showing the different shapes, scales, orientations, and position.

Experiment

- There is no causal relation in the dataset, as every combination of latents factors is present. So we define our SCM $\mathfrak{C}$, represented in the next DAG.

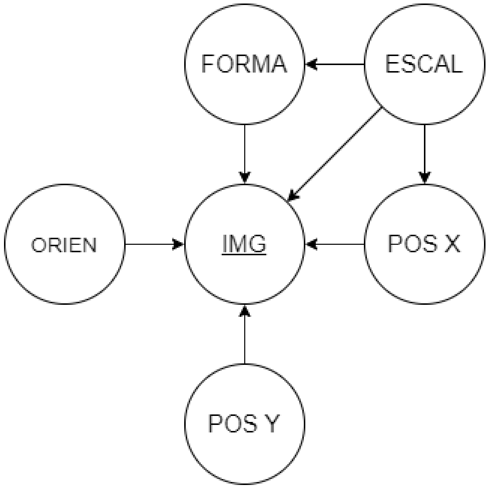

Figure 3: FORMA is shape and ESCAL is scale. The image is caused by every factor and there is causal relations between them.

Experiment

The causal story that was created for the dataset contains the structural assigments:

- The image is caused by every factor.
- Position Y and orientation are independent from the others variables.
- Scale is a cause of shape, if the scale is big, the shape always is a square.
- Scale is a cause of position X, the bigger the scale, the bigger is the probability for large values of position X.

Every structural assignment is simple and can be expressed by simple distributions and functions. However, we don't know a function g that image $= g(\text{shape, scale, orientation, pos.X, pos.Y})$. To express this function in the SCM we will train a variational autoencoder.

Variational autoencoder

Lets consider some dataset X_i of **i.i.d.** variables that are generated from some random process involving an unoberserved variable z. In the process there is as distribution $p_\theta(z)$ and a distribution $p_\theta(x|z)$, however, we don't know the true values of θ and the variable z. The variational autoencoder is an unsupervised learning method composed by an encoder and a decoder. The encoder is an neural network that approximates the distribution $p(z|x)$ and the decoder approximates the distribution $p(x|z)$. In this work, the probability distribution $p(x|z)$ is the fuction g necessary to relate the latent factos and the image.

Variational autoencoder

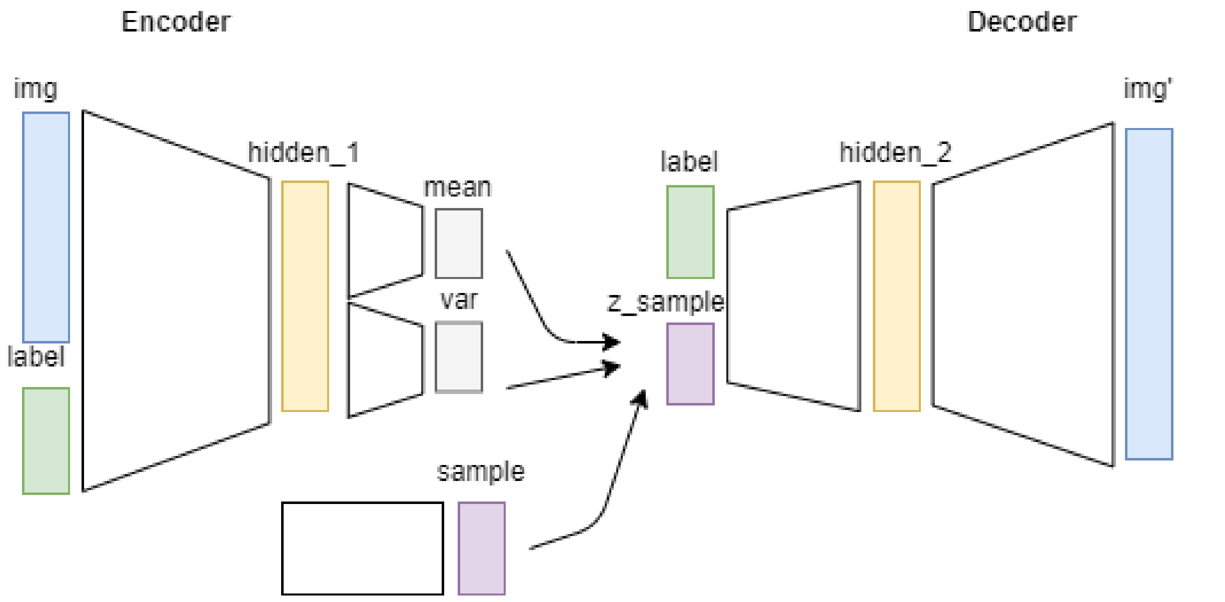

Figure 4: Architeture of variational autoencoder used. The label variable contains the latents factor of the image and is an entry for the encoder and also for the decoder.

Results

In the causal story created, there is a big relation between shape = square and bigger values of $pos.X$ (that occurs because of the effect of the scale). With the shape free, we can look at the expected value of the image in this particular case.

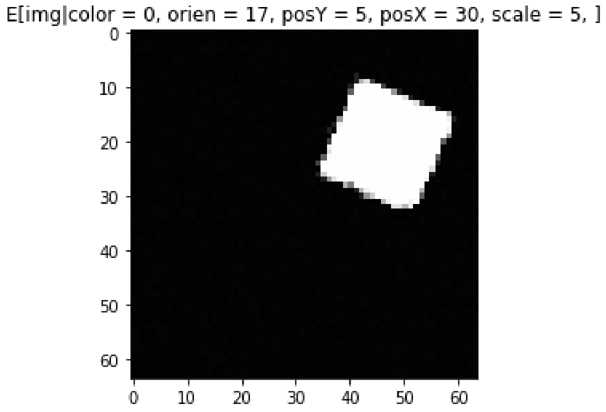

Figure 5: Expected value of the image, conditioned on orientation = 17, position Y = 5, position X = 30, scale = 5. The shape variable is free, but the expected value of the image shows clearly that it is a square.

Results

Because of the correlation, we can expect that if we condition on low values os position X, the shape will not be a square.

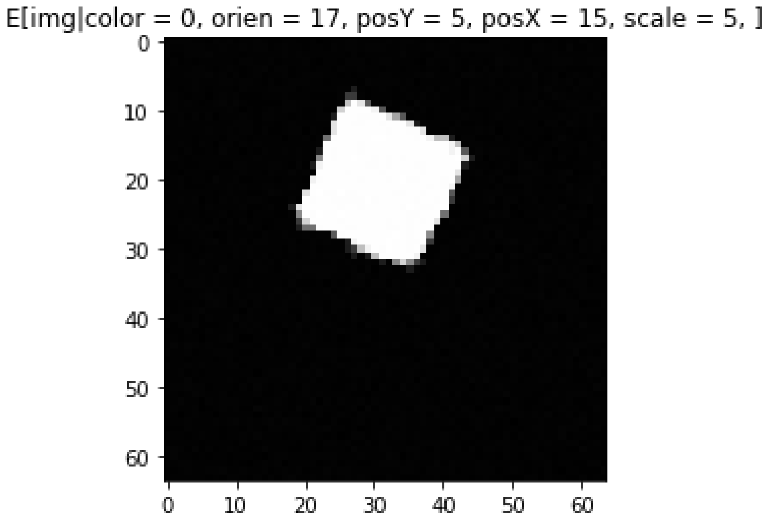

Figure 6: Expected value of the image, conditioned on orientation = 17, position Y = 5, position X = 15, scale = 5. The shape variable is free and the pos.X is low, however it still being a square.

Results

When we apply a counterfactual query, with the observed values being orientation = 17, position Y = 5, position X = 30, scale = 5, and make the scale = 1, the result is that the shape is not a square, because we affect one of the causes of the variable, the shape.

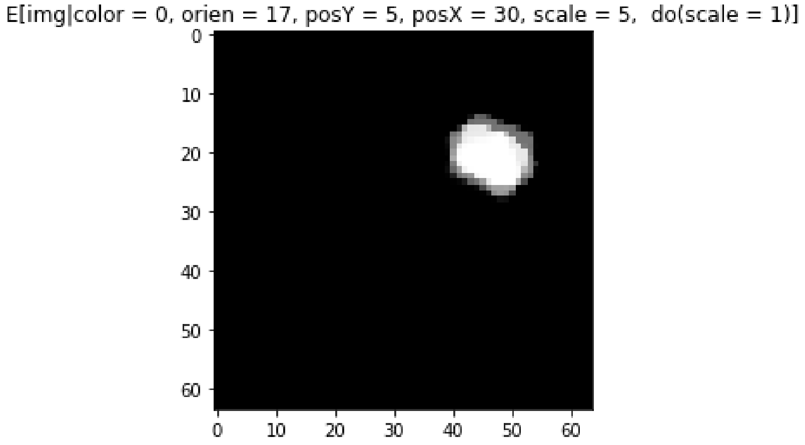

Figure 7: Expected value of the image in an counterfactual query. We intervened in one of the causes of the variable shape, and now the shape is not defined anymore, is a combination of the 3 differents values.

Limitation

- The DAG is required as an input for the algorithm.
- The method for inference of the noise prior distribution is simple.

Where to go next. . .

- Study what are the currently methods for learning of the DAG and see how are they being used and how effectively.
- Change the currently work to a better and more general method of inferece of the noise prior distribution.
- Apply the currently method in real data (education, health, criminality) where a DAG is known and analyse interventions and counterfactual situations.