Diffusion Causal Model

Yassine BEN JEMAA, Victor GERTNER

August 13, 2024

Contents

1 Introduction

In causal inference, we aim to intervene to quantify the causal effect of a given action. In doing so, we generate counterfactuals that constitute hypothetical scenarios. Thus, we are not only capturing the conditional distribution of our data, but we are also trying to capture the underlying physical mechanism that generated the data within a model. In this context, significant progress has been made in causal estimation in low dimensions. However, it remains challenging to achieve this in a framework where our data is of high dimensions. Therefore, we will explore causal estimation in high dimensions, employing a deep learning model. In this framework, we will examine how advances in energy-based generative models enable us to perform causal estimation. We will find that diffusion models fit perfectly into the realm of causal estimation for two main reasons: the stochastic nature of diffusion models directly corresponds to the uncertainty that causal models take into account. Additionally, we will see that the diffusion process allows us to easily perform interventions, thus providing an effective method for producing counterfactual data.

2 Diffusion Model

In this section, we'll introduce the concept of diffusion models and the notations we've chosen. Diffusion generative models have gained in popularity thanks to their performance, particularly for high-dimensional tasks (images, etc.). We're interested here in the Denoising Diffusion Probabilistic Model, whose aim is to learn the denoising of an iamge initially noised by Gaussian noise over a period of time. It can be divided into 2 actions:

The Forward Process: In this step, we gradually add noise β_t to the image x_0, considered as the initial image. Thus, by properties of conditional distribution on a normal distribution, we obtain an image x_t following the distribution:

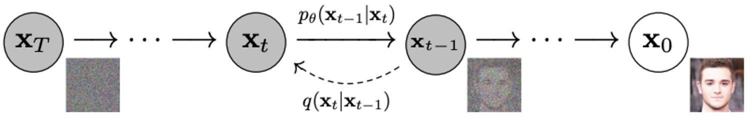
$$p(x_t \mid x_0) = \mathcal{N}\left(x_t; \sqrt{\alpha_t} x_0, (1-\alpha_t)\mathbf{I}\right)$$

with α_t defined as:

$$\alpha_t := \prod_{j=0}^{t} (1-\beta_j)$$

The Backward Process: This step involves learning the noise added during the forward process. We start with a distribution $p(x_t) = \int p_{\text{data}}(x) p(x_t|x)\, dx$ to arrive at $p(x_0) \approx p_{\text{data}}$. The instances t are connected by a Markov Chain that gradually removes the noise until the image is deblurred.

The learning of the inverse process is done through a neural network ϵ_θ trained to denoise images, with the main objective of approximating $\nabla_{x_t} \log p_t(x_t|x_0)$.



$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$

Figure 1: Forward et Backward Process

After training the neural network, we can then find the relationship that links an image at time $t-1$ to its previous instance at time t:

$$\mathbf{x}(t-1) = \frac{1}{\sqrt{1-\beta_t}}\left[\mathbf{x}_t + \beta_t\, \boldsymbol{\epsilon}_{\theta^*}(\mathbf{x}_t, t)\right] + \sqrt{\beta_t}\mathbf{z}, \quad t = T \cdots 0\ ,\ \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$$

For the experimentation phase, a diffusion model is already initially pre-trained.

3 Causality in the General Framework

It is important, before delving into counterfactuals within the framework of diffusion models, to understand them in a general context. Let $(G := (S, p_U))$ be constituted by a collection $(S = (f^{(1)}, f^{(2)}, ..., f^{(K)}))$ of structural assignments (referred to as mechanisms), defined as follows:

$$x^{(k)} := f^{(k)}(pa^{(k)}, u^{(k)}),$$

Where $X = \{x^{(1)}, x^{(2)}, ..., x^{(K)}\}$ are the known endogenous random variables,, $pa^{(k)}$ is the set of parents of $x^{(k)}$ (its direct causes), and $U = \{u^{(1)}, u^{(2)}, ..., u^{(K)}\}$ are the exogenous variables..

- An endogenous variable is a variable that appears as a dependent variable in at least one equation of the structural model. Here, we can see clearly how $x^{(k)}$ are endogenous variables.
- An exogenous variable is a variable that never appears as a dependent variable in the equations of a structural model. Here, we notice that the variables $u^{(k)}$ fit this definition as their only link is respectively with $x^{(k)}$, and it is only in the direction from $u^{(k)}$ to $x^{(k)}$.

The distribution $p(U)$ of the exogenous variables represents the uncertainty associated with the variables that have not been accounted for by the causal model. Furthermore, we consider them to be mutually independent. Thus, to align more closely with the framework presented in the course, we can transform these functional relationships into graphical relationships: the vertices represent the variables, and the edges among endogenous variables represent causal (directional) relationships between them.

Thus, once a Structural Causal Model (SCM) is defined, it is possible to perform interventions: a deliberate modification of one or more variables in the causal system represented by the graph, followed by an attempt to estimate the effects of this intervention.

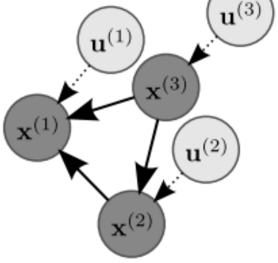


Figure 2: Exemple d'un SCM avec des variables endogènes $x^{(k)}$ et leurs variables exogènes respectives $u^{(k)}$.

4 Causality in diffusion models: Diff-SCM

4.1 Theoretical Framework

Thus, now that we have understood the significance of causality, we will link it with diffusion models: **Diffusion Causal Models for Counterfactual Estimation**[5].

As mentioned earlier, diffusion models have facilitated significant advances in generating high-dimensional data. Here, we will understand how this can be advantageous in the context of causality.

To proceed, we will start with three working hypotheses:

i) The SCM is known, and the intervention is identifiable. (i.e., this means that we can observe the effects of an intervention on a given variable in the model.)

ii) The variables on which the counterfactuals will be estimated must contain enough information to recover their causes; that is, an anticausal predictor can be formed.

iii) All endogenous variables in the training set are annotated. This allows for precise modeling of cause-and-effect relationships between different variables.

(It is worth recalling the definition of an anticausal predictor here: In this context, we consider the effect as an input, and we attempt to predict the value of the causal variable that caused it.

Consider, for example, the task of predicting the class label of a handwritten digit from its image. The causal structure is as follows: a person intends to write the digit 7, let's say, and this intention triggers a motor pattern producing an image of the digit 7 - in this sense, the class label Y causes the image X. [2])

In this model, we consider that our causal variables follow the dynamics of an Ito process $x_t^{(k)}, \forall t \in [0, T]$ transitioning from an endogenous variable for $t = 0$ to its respective exogenous

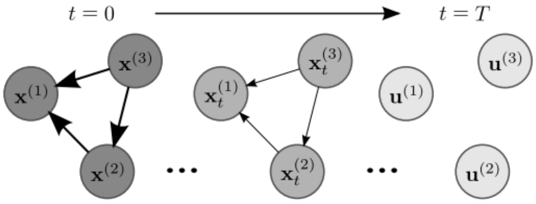


Figure 3: The diffusion process weakens the relationship between endogenous variables until they become completely independent at $t = T$. Arrows with solid lines indicate the causal relationship between variables and their direction, while the thickness of the arrow indicates the strength of the relationship. Note that the time t is a fiction used as a reference for the diffusion process and is not a causal variable.

noise $u^{(k)}$. Thus, during this Ito process, causal relationships between our variables gradually disappear, leaving only noise in our latent space. Therefore, we can define the diffusion model as follows:

Thus, for each node in our causal model, we can define an SDE based on [6]:

$$dx^{(k)} = -\frac{1}{2}\beta_t x^{(k)} dt + \sqrt{\beta_t} dw \text{ Where : } p(x_0^{(k)}) = \prod_{j=k}^{K} p(x^{(j)}|pa^{(j)}) \text{ And } p(x_T^{(k)}) = p(u^{(k)}) \quad (1)$$

Here, w represents a Brownian motion. This equation is obtained by taking the limit of the following Markov chain:

$$x_i = \sqrt{1-\beta_i} x_{i-1} + \sqrt{\beta_i} z_{i-1} \quad (2)$$

The theory behind this is complex, for more details, refer to [6].
On the other hand, the generative process is the solution of the reverse-Time SDE:

$$dx^{(k)} = \left[-\frac{1}{2}\beta_t + \beta_t \Delta_{x_t^{(k)}} \log p(x_t^{(k)}) \right] dt + \sqrt{\beta_t} \bar{w} \quad (3)$$

Thus, as we can see, we recover our data by iteratively adding the gradient of the distribution of our data with respect to our input variable. Therefore, we can see this process, this time, as making causal relationships stronger at each iteration, i.e., transitioning from exogenous noise that lacks causal relationships to our causal variables containing noise and parent/child relationships.

4.2 Counterfactual Estimation

4.2.1 From the DDIM Algorithm

Thus, in order to create an algorithm for simulating Counterfactuals, it is important to first understand the classical procedure proposed by [3] within the framework of diffusion models. There are therefore two steps:

- Generating new data from the latent space: **Sampling**
- Embedding a data point into the latent space: **Reverse Sampling**

Sampling

Here is the Sampling algorithm

Algorithm 1 Sampling with DDIM - Image Generation

```
Models: trained diffusion model $\epsilon_\theta$.
Input: $x_T \sim \mathcal{N}(0, I)$
Output: $x_0$ - Image
for $t \leftarrow T$ to 0 do do
    $x_{t-1} \leftarrow \sqrt{\alpha_{t-1}}\left(\frac{x_t-\sqrt{1-\alpha_t}\epsilon_\theta(x_t,t)}{\sqrt{\alpha_t}})\right) + \sqrt{\alpha_{t-1}}\epsilon_\theta(x_t, t))$
end for=0
```

As we can see the first term $\sqrt{\alpha_{t-1}}\left(\frac{x_t-\sqrt{1-\alpha_t}\epsilon_\theta(x_t,t)}{\sqrt{\alpha_t}})\right)$ is representing the "predicted x_0 from the actual state. And the second term $\sqrt{\alpha_{t-1}}\epsilon_\theta(x_t, t))$ reminds the model the direction pointing to x_t

Reverse-Sampling

Here is the Reverse-Sampling algorithm

Algorithm 2 Reverse-Sampling with DDIM - Inferring the Noisy Latent

```
Models: trained diffusion model $\theta$.
Input: $x_0$ - Image
Output: $x_T \sim \mathcal{N}(0, I)$
for $t \leftarrow T$ to 0 do do
    $x_{t+1} \leftarrow \sqrt{\alpha_{t+1}}\left(\frac{x_t-\sqrt{1-\alpha_t}\epsilon_\theta(x_t,t)}{\sqrt{\alpha_t}})\right) + \sqrt{\alpha_{t+1}}\epsilon_\theta(x_t, t))$
end for=0
```

Thus, these basic algorithms form the foundation of the suite. It is important to understand that these were chosen instead of DDPM because they provide an almost-invertible transformation between x_T and x_0, making it more efficient; i.e., fewer steps are needed to create the embedding or to sample.

4.2.2 Until the proposed algorithm

Thus, in addition to the Sampling/Reverse-Sampling part, we now need to consider the framework imposed by Pearl. Indeed, Pearl imposes a schema for simulating counterfactuals:

- **Abduction**: Estimation of the exogenous noise U
- **Action**: Alteration of the graph by removing the edges between the intervened variable and its parents. (These should no longer have influence during this simulation)
- **Prediction**: Estimation of the counterfactual using the abduced noise and intervention values.

Thus, for the first step, in the context of a diffusion model, estimating the exogenous noise amounts to performing a Reverse-Sampling phase, i.e., going into the latent space learned previously.

For the second step, we will only use graphs with two states $x^1 \to x^2$, so this step is not considered.

Finally, to estimate the counterfactual, we can adapt the sampling algorithm: indeed, simulating x^2_{CF} after intervening on x^1 ($do(x^1 = x^1_{CF})$) amounts to sampling according to $p(x^2|do(x^1 = x^1_{CF}); x^2 = x^2_{CF})$.

Thus, we can now revert to equation (3) by applying the fact that the effect on x^2 of an intervention on x^1 is equivalent to solving a reverse diffusion process for $x_t^{(2)}$ using the gradient of an anticausal predictor with respect to $x_t^{(2)}$.

Based on this, here is the proposed algorithm following the logic of DDIM:

Algorithm 3 Inference of the counterfactual for a variable $x^{(k)}$ from a diffusion model ϵ_θ and an anticausal predictor p_ϕ.

```
function InferenceContrefactuel($x_{0,F}^{(k)}, x_{0,CF}^{(j)}, s$)
    Abduction du Bruit Exogène
    $u^{(k)} \leftarrow$ Récupérer $u^{(k)}$ à partir de $x_{0,F}^{(k)}$
    for $t \leftarrow 0$ à $T$ do
        $x_{t+1,F}^{(k)} \leftarrow \sqrt{\alpha_{t+1}} \frac{\left(x_{t,F}^{(k)} - \sqrt{1-\alpha_t}\theta(x_{t,F}^{(k)}, t)\right)}{\sqrt{\alpha_t}} + \sqrt{\alpha_{t+1}}\epsilon_\theta(x_{t,F}^{(k)}, t)$
    end for

    Génération sous Intervention
    for $t \leftarrow T$ à $0$ do
        $\epsilon \leftarrow \epsilon_\theta(x_t^{(k)}, t) - s\sqrt{1-\alpha_t}\nabla_{x_t^{(k)}} \log p_\phi(x_{0,CF}^{(j)}|x_t^{(k)})$
        $x_{t-1}^{(k)} \leftarrow \sqrt{\alpha_{t-1}} \frac{\left(x_t^{(k)} - \sqrt{1-\alpha_t}\epsilon\sqrt{\alpha_t}\right)}{\sqrt{\alpha_t}} + \sqrt{\alpha_{t-1}}\epsilon$
    end for
    $x_{0,CF}^{(k)} \leftarrow x_0^{(k)}$
end function=0
```

So, we can see that we have adapted our two previous algorithms to make them into one. It is important to explain this. We notice that for the first part: the abduction of the exogenous noise, nothing has changed except for the variable notations.

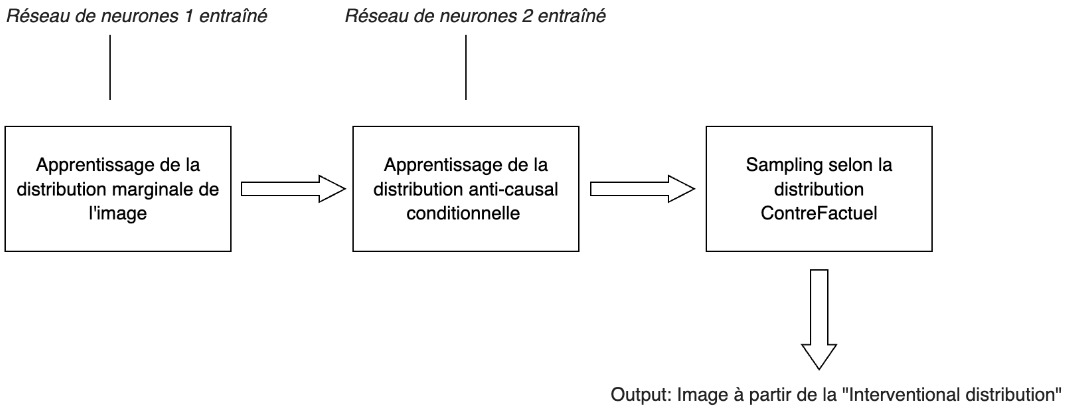
In the second part, we see that the term epsilon is calculated not only from the neural network of our diffusion model but also from the term $\nabla_{x_t^{(k)}} \log p_\phi(x_{0,CF}^{(j)}|x_t^{(k)})$. Indeed, in this sum, epsilon typically allows us to reconstruct an image belonging to the distribution of our data from the noise, and the second term adds a direction to this reconstruction: that of the counterfactual. This echoes what we just said above about the way to diffuse towards a counterfactual from (3).

In order to have more control over belonging to our distribution / resembling the counterfactual, a parameter s is added, which allows us to put more or less weight on the gradient. We will see its influence in practice later.

5 Experimentation

5.1 Implementation

In the general case, the causal graph is rarely known, just like the Counterfactual, which is only a simple hypothesis. Thus, to construct the model, we will follow the following implementation:


Réseau de neurones 1 entraîné
Réseau de neurones 2 entraîné
Apprentissage de la distribution marginale de l'image
Apprentissage de la distribution anti-causal conditionnelle
Sampling selon la distribution ContreFactuel
Output: Image à partir de la "Interventional distribution"

The chosen databases are **MNIST** and **ImageNet**.

5.2 Choice of Metrics

To evaluate the results, two metrics will be used: IM1 and IM2 [1]. These metrics calculate the level of "realism" of the generated image as well as the proximity of the image to the database and the label used during the intervention.

$$\mathbf{IM1}(x_{\text{CF}}^{(1)}, x_{\text{F}}^{(2)}, x_{\text{CF}}^{(2)}) = \frac{\left\| x_{\text{CF}}^{(1)} - \text{AE}_{x_{\text{CF}}^{(2)}}(x_{\text{CF}}^{(1)}) \right\|_2^2}{\left\| x_{\text{CF}}^{(1)} - \text{AE}_{x_{\text{F}}^{(2)}}(x_{\text{CF}}^{(1)}) \right\|_2^2 + \epsilon}$$

$$\mathbf{IM2}(x_{\text{CF}}^{(1)}, x_{\text{CF}}^{(2)}) = \frac{\left\| \text{AE}_{x_{\text{F}}^{(2)}}(x_{\text{CF}}^{(1)}) - \text{AE}(x_{\text{CF}}^{(1)}) \right\|_2^2}{\left\| x_{\text{CF}}^{(1)} \right\|_1 + \epsilon}$$

In this case, IM1 measures the ratio between the reconstruction errors of x_{CF} using $AE_{x^2_{CF}}$

and $AE_{x_F^2}$, while IM2 compares the similarity of counterfactual instances reconstructed using $AE_{x_F^2}$ and an autoencoder trained on all classes, AE.

A lower value for IM1 means that x_{CF} is better reconstructed by the autoencoder that has only seen examples of the counterfactual class x_{CF}^2 than by the autoencoder trained on the original class x_F^2. This implies that x_{CF} is closer to the counterfactual class x_{CF}^2 compared to x_F^2, and thus more realistic.

A low value of IM2 means that the auto-encoded images x_{CF} are very similar when using $AE_{x_F^2}$ or AE. Thus, the data distribution of class x_F^2 describes x_{CF} as well as the distribution over all classes, meaning that the generated image has ”moved away” from the original class in its latent space. This implies that the counterfactual is well interpretable.

5.3 Results

For the MNIST dataset, diffusion models are trained by U-Nets. From a qualitative point of view, the generated images are satisfactory. We notice that the characters retain the ”characteristics” of the original image (fine/thick writing, size, etc.). From a quantitative point of view, IM1 and IM2 are of the order of magnitude of 0.94 and 0.04, which is better than the results obtained in previous research [4].

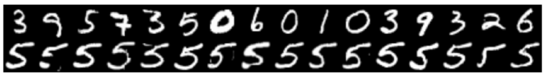


Figure 4: MNIST Dataset: do(5)

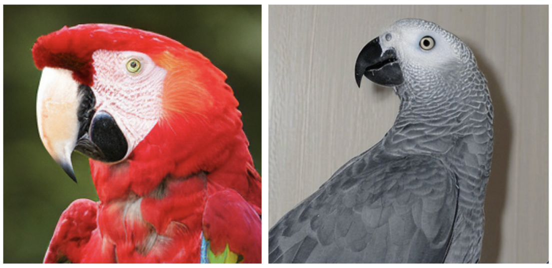


Figure 5: ImageNet Dataset: do(other_species)

For the ImageNet dataset, transformers are used instead of U-Nets for training the diffusion model [7].

5.4 Potential Improvement of Measurement

A problem with the metrics used is that they use the l_1 measure to evaluate similarity. However, this measure can be poor in the case of pixels since an intervention can modify the structure of the image while maintaining other factors unchanged. We then introduce a new metric, Counterfactual Latent Divergence (CLD), which calculates the LogSumExp of the two probability measures on the true class and the counterfactual class.

Additionally, this metric will allow fine-tuning of the hyperparameter s introduced in **Algorithm 3**.

Typically, a small s means that counterfactuals are only reconstructions of factual data, resulting in a high CLD. A high s "removes" the diffusion model from ϵ and only considers the intervention parameter, also resulting in a high CLD.

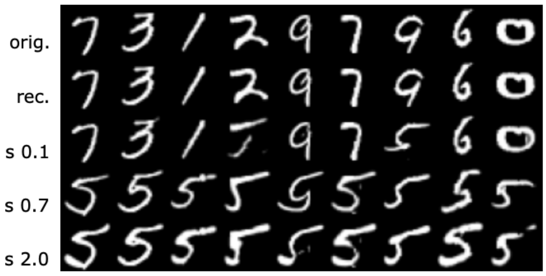


Figure 6: Influence of s on MNIST (from the article)

6 Conclusion

In summary, Diff-SCM integrates generative diffusion models with structural causal models, providing a novel approach to causal estimation through distribution gradients. The former serves as both an intervention algorithm and a counterfactual estimation method, tackling the challenge of counterfactual evaluation. Additionally, several metrics was proprosed for quantifying either the obtained image or the latent space distance.

By demonstrating good performance on MNIST, Diff-SCM holds potential for diverse applications, including critical domains like medical imaging for cancer detection.

References

[1] Janis Klaise Arnaud Van Looveren. Interpretable counterfactual explanations guided by prototypes. 2020.

[2] Jonas Peters Eleni Sgouritsa Kun Zhang Bernhard Scholkopf, Dominik Janzing. On causal and anticausal learning. 2012.

[3] Stefano Ermon Jiaming Song, Chenlin Meng. Denoising diffusion implicit models. 2020.

[4] Oscar Key Lisa Schut. Generating interpretable counterfactual explanations by implicit minimisation of epistemic and aleatoric uncertainties. 2021.

[5] Pedro Sanchez. Diffusion causal models for counterfactual estimation. 2022.

[6] Yang Song. Score-based generative modeling through stochastic differential equations. 2021.

[7] Saining Xie William Peebles. Scalable diffusion models with transformers. 2022.