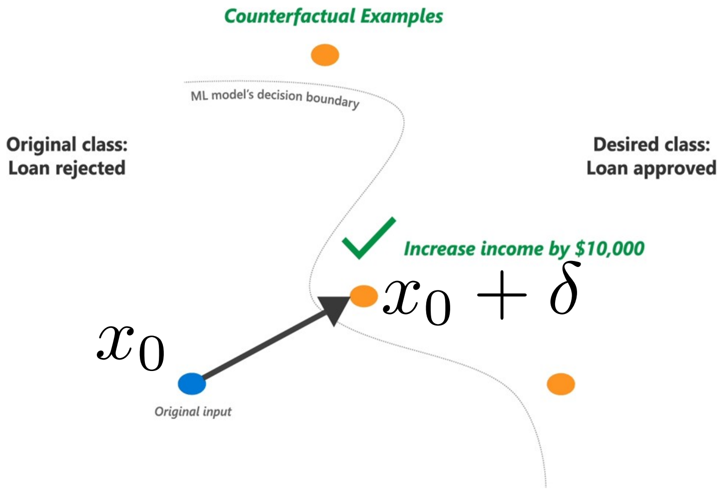
VCNet: A self-explaining model for realistic counterfactual generation

HyAIAI June 8th 2022

Table of contents

Counterfactual explanation



<u>Counterfactual explanation for machine learning models:</u>

The smallest change of feature values that changes the prediction to a given output.
(Watcher et al., Harvard Journal of Law & Technology 2018)

First thesis contribution

Optimize the following cost function:

$$\min_{\delta} \left(c \cdot f_{\tau}(x_0, \delta) + \boxed{||\delta||} + \gamma \cdot L_{AE} + \theta \cdot L_{\text{proto}} \right)$$

Three desired properties:

- **Sparsity:** change only few features

- **Closeness to the data manifold:** counterfactuals close to the training data distribution

- **Closeness according to the counterfactual class:** counterfactuals close to the training data distribution but according to the predicted class

Arnaud Van Looveren and Janis Klaise, Interpretable Counterfactual Explanations Guided by Prototypes, 2021,ECML

First thesis contribution

Optimize the following cost function:

$$\min_{\delta} \left(c \cdot f_{\tau}(x_0, \delta) + ||\delta|| + \gamma \cdot \boxed{L_{AE}} + \theta \cdot L_{\text{proto}} \right)$$

Three desired properties:

- **Sparsity:** change only few features

- **Closeness to the data manifold:** counterfactuals close to the training data distribution

- **Closeness according to the counterfactual class:** counterfactuals close to the training data distribution but according to the predicted class

Arnaud Van Looveren and Janis Klaise, Interpretable Counterfactual Explanations Guided by Prototypes, 2021,ECML

First thesis contribution

<u>Optimize the following cost function:</u>

$$\min_{\delta}\left(c \cdot f_{\tau}(x_0, \delta) + ||\delta|| + \gamma \cdot L_{AE} + \theta \cdot \boxed{L_{\text{proto}}}\right)$$

<u>Three desired properties:</u>

- **Sparsity:** change only few features

- **Closeness to the data manifold:** counterfactuals close to the training data distribution

- **Closeness according to the counterfactual class:** counterfactuals close to the training data distribution but according to the predicted class

Arnaud Van Looveren and Janis Klaise, Interpretable Counterfactual Explanations Guided by Prototypes, 2021,ECML

First this contribution

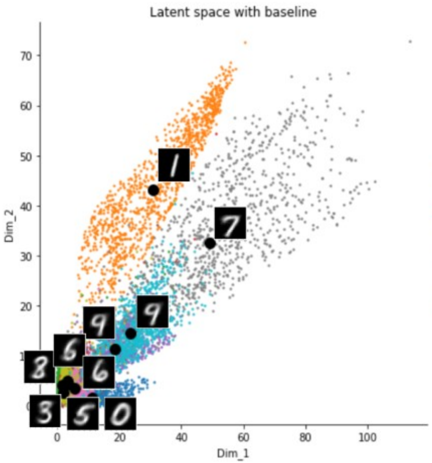
Goal: Generate counterfactuals that are **more representative** of the **predicted class**.

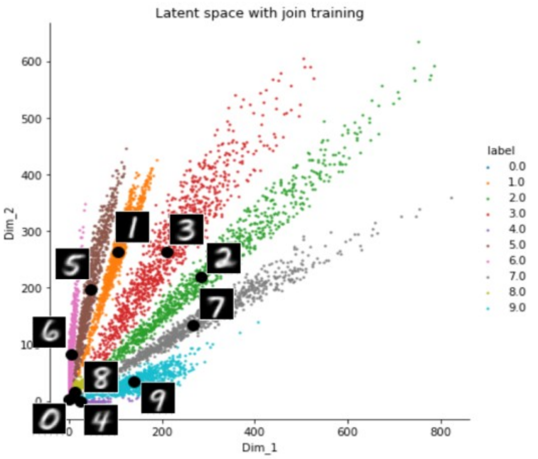
x_0 c_1 c_2

x_0 The example to explain, predicted class 5

c_1, c_2 Counterfactuals both predicted class 6

The idea: Generate more representative prototypes by using a **supervised autoencoder,** then we obtain more representative counterfactuals.





Publications

[1] Post-hoc counterfactual generation with supervised autoencoder, 2021, AIMLAI ECML

[2] Générer des explications contrefactuelles à l'aide d'un autoencodeur supervisé, 2022, EGC

Note: [2] is an extended version of [1] that contains evaluations on numerical data.

<u>In the next slides:</u> limitations + existing solutions that have motivated the choices in our contribution.

Limitations

1) Scalability issue: For each new example to explained, it is necessary to optimize the cost function again, which can be **costly in terms of computing time**, thus making the process **unscalable**.

A solution: Train a **model** to generate counterfactuals.
Then, counterfactuals are obtained by forwarding the example to explain through the model.
Such models are often based on **generative models** such as VAE or GAN [1,2,3]

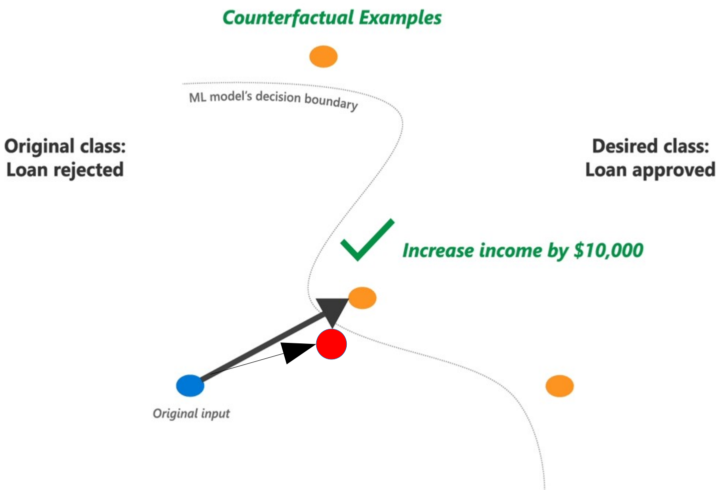
[1] Nemirovsky et al., Countergan: Generating realistic counterfactuals with residual generative adversarial network, arXiv, 2020

[2] Downs et al., Cruds: Counterfactual recourse using disentangled subspaces, WHI ECML, 2020

[3] Van looveren et al., Conditional generative models for counterfactual explanations, arXiv, 2021

Limitations

2) Validity issue: A counterfactual is said to be valid if it **succeeds** in **reaching** a **different prediction** (the other side of the decision border is successfully reached).
In a post-hoc paradigm, the counterfactual search process is completely uninformed from the decision process, which can leads to counterfactual validity issue.



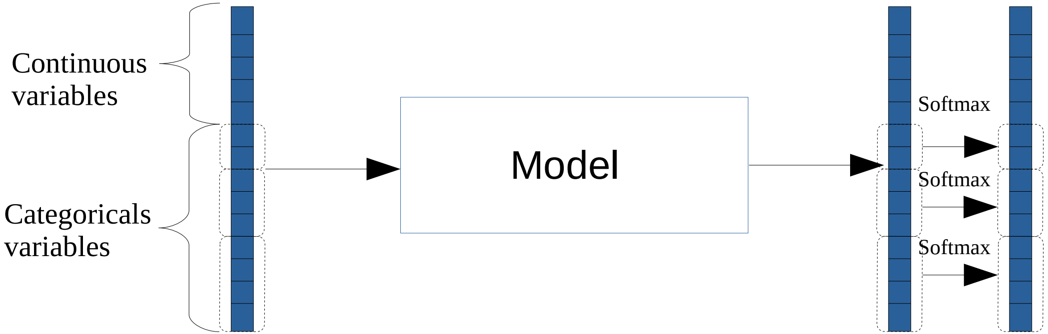
A solution: Learn **jointly** the **prediction task** and the **explanation task**

Guo et al., End-to-End Training of Counterfactual Aware Predictions, 2021, ICML Workshop on Algorithmic Recourse

Limitations

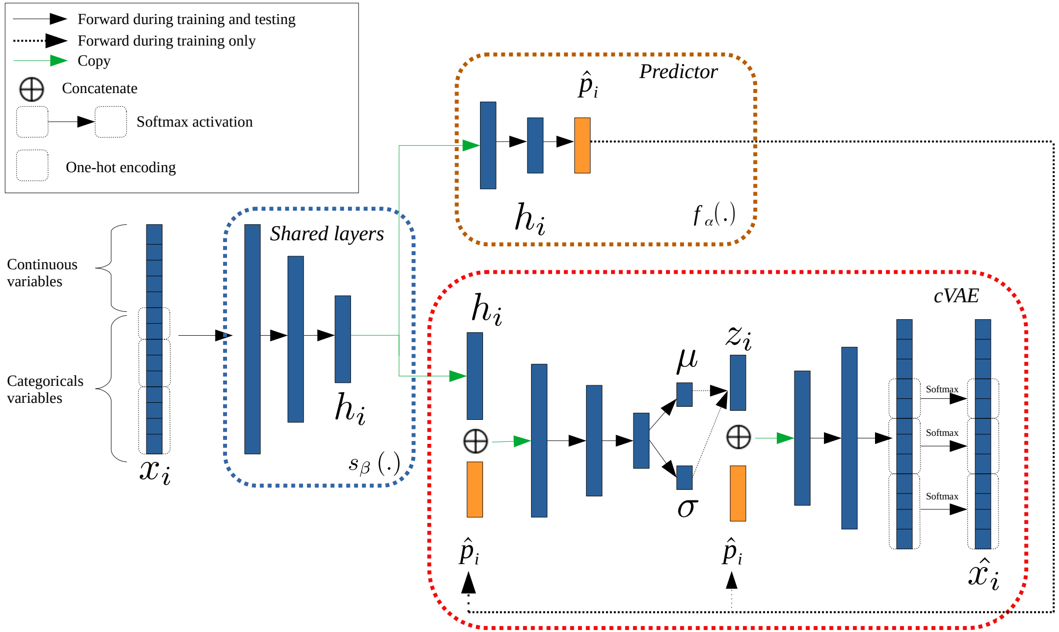
3) Handle categorical variables: Our approach does not handle categorical data.
More specifically, the approach initially proposed by Van looveren et al. turned out to be **inapplicable** in the context of a **supervised autoencoder**.

<u>A solution:</u> Encode each categorical variable with a **one-hot encoding**.
Train a model to generate counterfactuals and add a **softmax activation** function for each one-hot categorical variable in order to obtain a one-hot encoding format by taking the argmax.
This ensures the counterfactual satisfies the categorical data format.

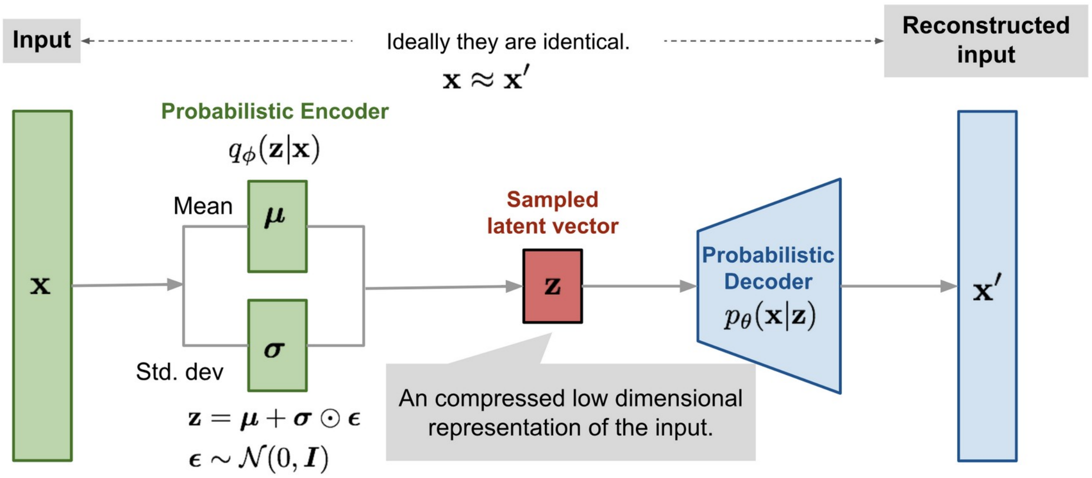


A new contribution: VCnet

Vcnet is based on a conditional variational autoencoder (cVAE)



Variational autoencoder (VAE)



- Learn a latent distribution and not a latent representation.

- Often gaussian distributions

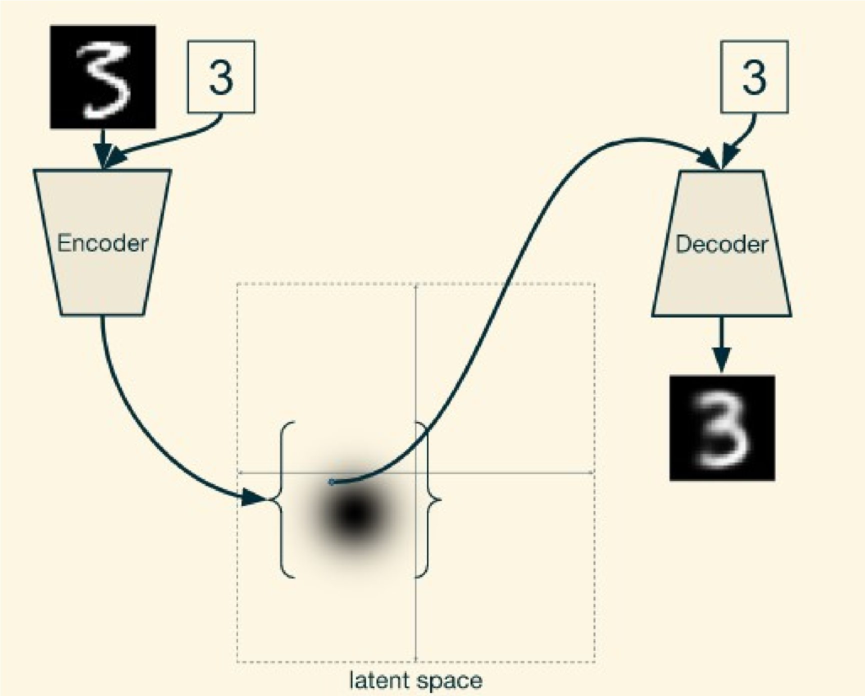
<u>Loss function:</u> $\mathcal{L}_{\mathrm{VAE}}(\theta, \phi) = -\mathbb{E}_{q_\phi(z|x)}[\log(p_\theta(x \mid z))] + \mathrm{D}_{KL}\left[q_\phi(z \mid x) \middle\| p(z)\right]$

Reconstruction error term Regularization term

$p \sim \mathcal{N}(0, I)$

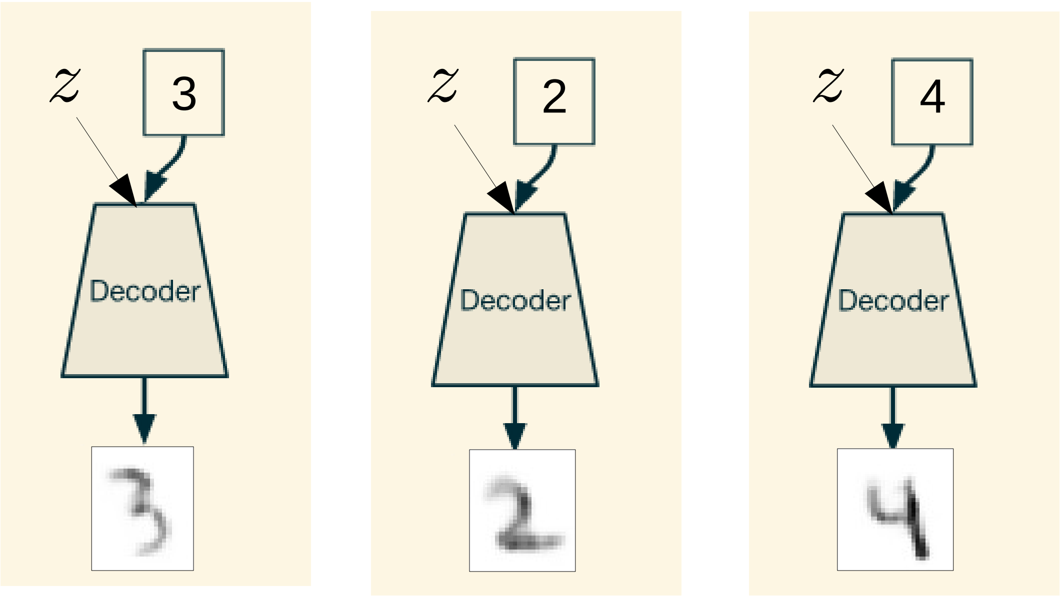
Conditional variational autoencoder (cVAE)

Conditional variational autoencoder (cVAE):



- Generative model: Generate an example according to a class

Interest of a cVAE



Given a sample latent vector z, change the class at inference.

Kingma et al., semi-supervised learning with deep generative models, 2014, NIPS

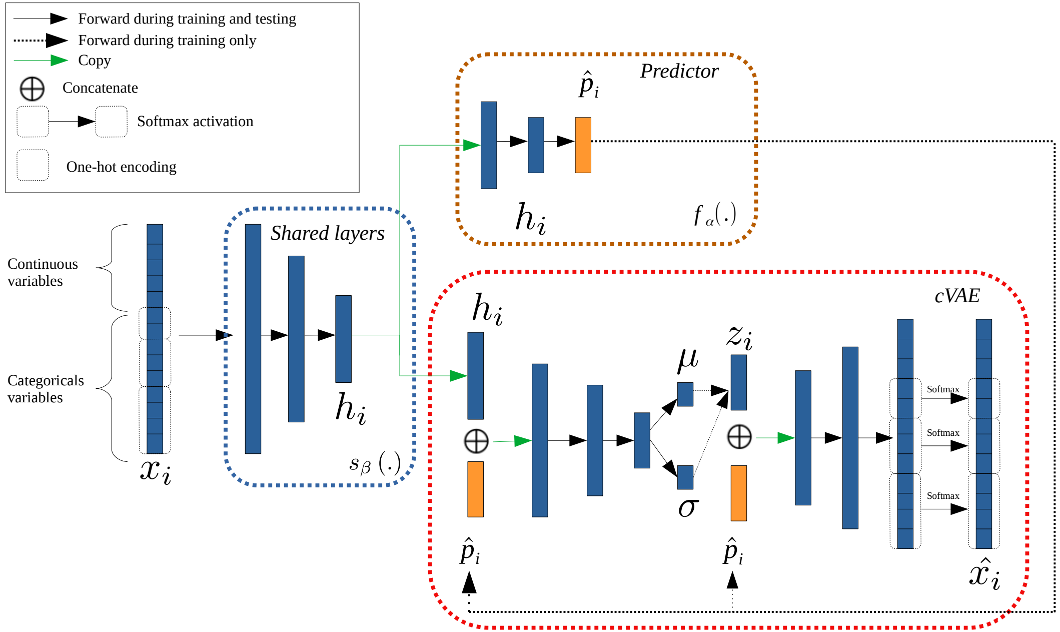
Interest of a cVAE

$z, 2$ $z, 3$ $z, 4$

z encode latent features that are not relevant for classification.
Here we obtain digits with the **same “handwritting”** but with a **different class.**
In this case the two numbers are **close** because they **share the same handwriting** but **different** because they do **not share the same class.**

This is exactly what is expected for **counterfactual generation**, as we want to generate an example that have a **close representation** but which is **representative of a different class**.

VCnet architecture



<u>It's composed of 3 blocks:</u>

a) A prediction block (for the prediction task)

b) A cVAE block that will be used as a counterfactual generator during testing

c) A shared block that built a shared representation common for a) and b)

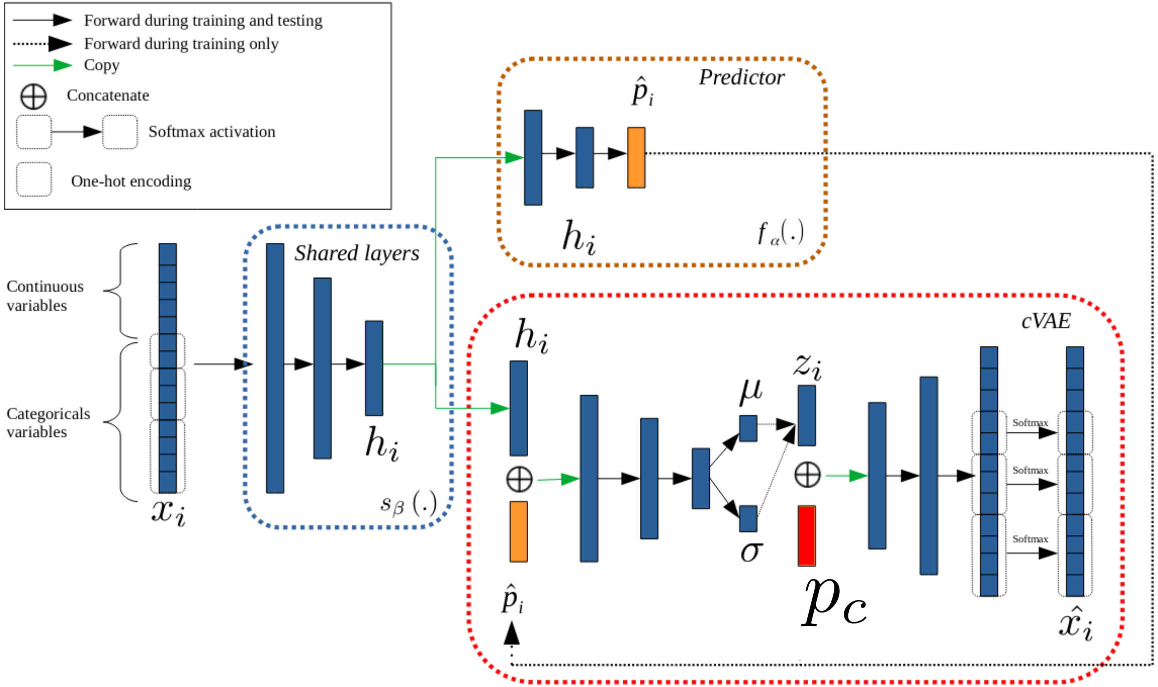
Training procedure

The network is learned in a **single optimization process** thanks to back-propagation.

The loss is a weighted sum of a **cVAE loss** and a **classification loss**:

$$\mathcal{L}\left(\theta, \alpha, \beta, \phi; D\right) = \sum_{i=1}^{n} \mathcal{L}_{cVAE}\left(\theta, \phi, \beta; x_i\right) + \lambda_2 \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{pred}\left(\alpha, \beta; x_i, y_i\right)$$

Counterfactual generation



At inference time, a vector p_c is passed to the decoder instead of $\hat{p}_i$

How to choose p_c?

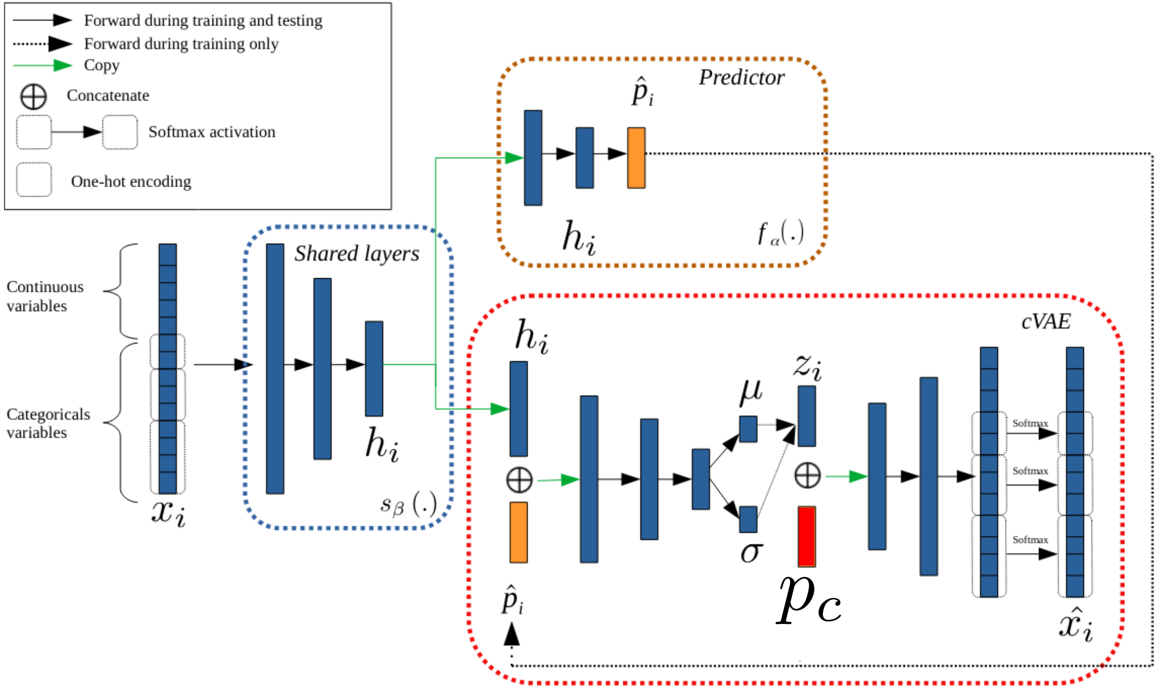
Because we want to generate an example with a **different predicted class** we need a probability vector such that the **class with maximum probability** is **different** from the one of the **prediction**. We decided to use a one-hot vector where the probability is 0 for the predicted class and 1 for the opposite class (this works only in a binary classification setup).

<u>Ex:</u>

$$\hat{p}_i = [0.2, 0.8]$$

$$p_c = [1, 0]$$

Intuition



Intuitively, z_i encode **non class relevant** information about x_i and p_c encodes **information** related to the **desired class**.

Evaluation

To the best of our knowledge, **CounterNet** is the only method that proposes to **learn jointly counterfactuals and predictions**.
Thus, we compare the quality of the counterfactuals produced with those of CounterNet on different datasets through state-of-the-art metrics.
We plan to compare also with the first contribution.

<u>Metrics:</u>

- **Accuracy:** model performance metric
- **Validity:** 1 if the counterfactuals achieves a different predicted class else 0
- **Proximity:** L1 distance between the example to explain and the counterfactual
- **Prediction gain:** differences between predicted class for the example and predicted class for counterfactual
- **Proximity score:** normalized distance of the counterfactual to an existing example with the same predicted class

Results

		VCNet	CounterNet
Adult	Validity	**1.0**	0.99
	Proximity	7.71 ± 2.11	**7.16± 2.13**
	Prediction gain	**0.76 ± 0.15**	0.61±0.17
	Proximity score	**0.04 ± 0.11**	0.31± 0.28
	Accuracy	**0.83**	**0.83**
OULAD	Validity	**1.0**	0.99
	Proximity	11.66±2.46	11.96±2.40
	Prediction gain	**0.93±0.12**	0.74±0.13
	Proximity score	**0.38±0.18**	0.46±0.16
	Accuracy	**0.93**	**0.93**
HELOC	Validity	**1.0**	0.99
	Proximity	5.60±2.11	**4.41±1.80**
	Prediction gain	**0.64±0.13**	0.56±0.15
	Proximity score	**0.23±0.21**	0.49±0.35
	Accuracy	0.71	**0.72**
Student	Validity	0.96	**1.0**
	Proximity	19.90±3.21	19.86±2.78
	Prediction gain	**0.86±0.27**	0.76±0.05
	Proximity score	**0.70±0.08**	0.73±0.06
	Accuracy	0.90	**0.92**
Titanic	Validity	0.92	**0.99**
	Proximity	15.43±3.79	**15.15±4.05**
	Prediction gain	**0.69±0.31**	0.66±0.15
	Proximity score	**0.71±0.21**	0.80± 0.16
	Accuracy	0.82	**0.83**
Breast-cancer	Validity	**1.0**	**1.0**
	Proximity	5.27±1.47	**1.51±1.01**
	Prediction gain	**0.95 ± 0.11**	0.69±0.15
	Proximity score	**0.28±0.03**	0.72±0.48
	Accuracy	**0.96**	**0.96**

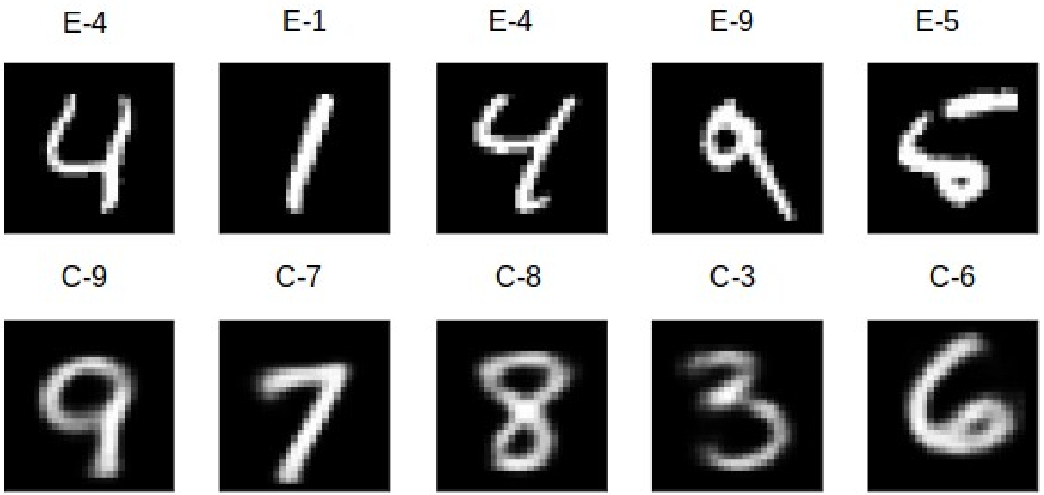
- Comparable levels of accuracies
- 100% of validity on 4 of the 6 datasets
- A better prediction gain and proximity score for every dataset
- A higher proximity

Higher prediction gain = more confidence in the class change of the counterfactual

Higher proximity score = counterfactuals are closer to an existing example of the same predicted class

Higher proximity = counterfactuals are obtained at the cost of larger changes of the input space

Easily adaptable to image data



Summary

1) A first contribution to generate counterfactuals according to three properties

2) Three identified limitations (validity/scalability/categorical variables)

3) A self-explainable model for counterfactual generation (Vcnet)

- A scalable process (counterfactuals are obtained in a single forward pass)
- A treatment for categorical data based on softmax functions

- Realistic counterfactuals
- 100% of validity on 4 of the 6 datasets

This work has been submitted to **ECML 2022**

Thank you !