Diffeomorphic Explanations with Normalizing Flows

Jan E. Gerken

CHALMERS
UNIVERSITY OF TECHNOLOGY

Technical University Berlin
8 October 2021

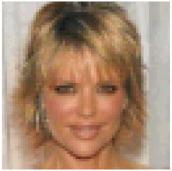
Based on joint work with
Ann-Kathrin Dombrowski and Pan Kessel

Published as a Contributed Talk at
2021 ICML Workshop on Invertible Neural Networks
Normalizing Flows, and Explicit Likelihood Models

Counterfactual explanations

- ***Counterfactual of a sample***: Data point close to original but with different classification
- Difference between original and counterfactual reveals features which led to classification
- Example from CelebA dataset, classified as not-blonde:

original x



counterfactual x'

$|x - x'|$

Adversarial examples

- For a classifier $f : X \to [0, 1]^C$, can generate adversarial example in class k by computing

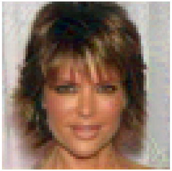
$$\text{argmax}_x\, f_k(x)$$

approximately by gradient ascent

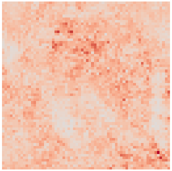
$$x^{(t+1)} = x^{(t)} + \lambda \frac{\partial f_k}{\partial x}(x^{(t)})$$

- Problem: The adversarial example does not lie on the data manifold, so is not a counterfactual

original x
blonde $p \approx 0.01$



adversarial example x'
blonde $p \approx 0.99$



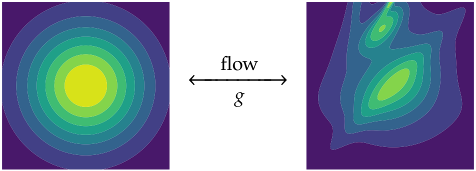
$|x - x'|$

Normalizing flows

- Generative model g which maps base space Z to data space X ***bijectively***
- Probability distribution q_Z in Z is simple, e.g. uniform or normal
- Probability distribution q_X in X is given by change of variables

$$q_X(x) = q_Z(g^{-1}(x)) \left| \det \frac{\partial z}{\partial x} \right|$$

- g is realized as a neural network with bijective, easily invertible building blocks



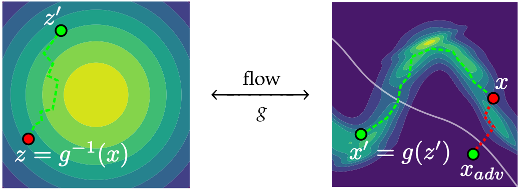
base space Z data space X

Counterfactuals from normalizing flows

- Generate counterfactual by performing gradient ascent in base space of normalizing flow:

$$z^{(t+1)} = z^{(t)} + \lambda \frac{\partial (f \circ g)_k}{\partial z}(z^{(t)})$$

- In this way, stay on data manifold:



base space Z data space X

Gradient ascent in base space

- Gradient ascent in Z for class k of the classifier f with learning rate λ:

$$z^{(t+1)} = z^{(t)} + \lambda \frac{\partial (f \circ g)_k}{\partial z}(z^{(t)})$$

- Using change-of-variable under the flow:

Theorem

Gradient ascent in the base space Z is given by

$$x^{(t+1)} = x^{(t)} + \lambda \, \boldsymbol{\gamma^{-1}(x^{(t)})} \, \frac{\partial f_k}{\partial x}(x^{(t)}) + \mathcal{O}(\lambda^2)$$

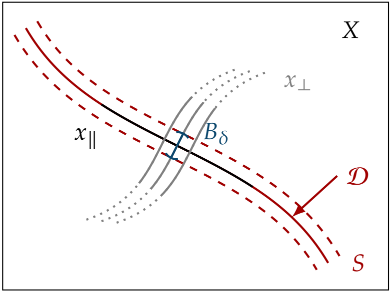
where $\gamma^{-1}(x) = \left(\frac{\partial g}{\partial z} \frac{\partial g}{\partial z}^T\right)(g^{-1}(x))$ is the inverse of the induced metric on X from Z under the flow g.

Coordinates in data space

- Model data manifold S by submanifold $\mathcal{D}$ and balls B_δ with small radii δ,

$$S = \mathcal{D} \times B_{\delta_1} \times \cdots \times B_{\delta_n}$$

- Use ***Gaussian normal coordinates*** on S



The induced metric in normal coordinates

Theorem

In Gaussian normal coordinates, the inverse induced metric γ^{-1} takes the form

$$\gamma^{-1} = \begin{pmatrix} \gamma_{\mathcal{D}}^{-1} & & & \\ & \gamma_{B_{\delta_1}}^{-1} & & \\ & & \ddots & \\ & & & \gamma_{B_{\delta_n}}^{-1} \end{pmatrix}$$

and, for well-trained flows, $\gamma_{B_{\delta_i}}^{-1} \longrightarrow 0$ for $\delta_i \longrightarrow 0$.

$\Rightarrow$ Since γ^{-1} multiplies the learning rate in the gradient ascent update,

$$x^{(t+1)} = x^{(t)} + \lambda\, \boldsymbol{\gamma^{-1}(x^{(t)})}\, \frac{\partial f_k}{\partial x}(x^{(t)}) + \mathcal{O}(\lambda^2)\,,$$

the directions orthogonal to $\mathcal{D}$ are scaled by a vanishing factor.

$\Rightarrow$ We stay on the data manifold.

Sketch of proof

i.e. $\mathrm{KL}(p_X, q_X) < \epsilon$

- For well-trained flows g, almost all probability mass is concentrated in $S = \mathcal{D} \times B_{\delta_1} \times \cdots \times B_{\delta_{N-n}}$

$$1 - \epsilon < \int_S q_X(x)\,\mathrm{d}x = \int_S \sqrt{\det|\gamma|}\, q_Z(g^{-1}(x))\,\mathrm{d}x$$

- When $\delta_i \longrightarrow 0$, the integration domain shrinks to zero, but the value of the integral is bounded from below

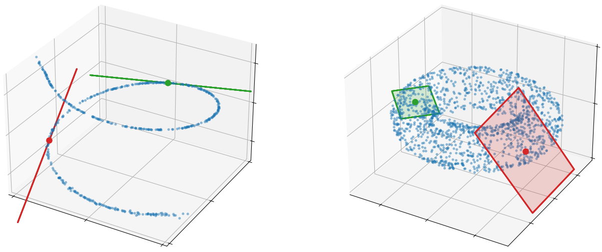
- Hence, the metric $\gamma_{B_{\delta_i}}$ has to diverge, i.e. $\gamma^{-1}_{B_{\delta_i}} \longrightarrow 0$.

Tangent space of data manifold

- From induced metric, can infer tangent space of data manifold
- Perform singular value decomposition of the Jacobian $\frac{\partial g}{\partial z} = U\,\Sigma\,V$
- Rewrite the inverse induced metric as

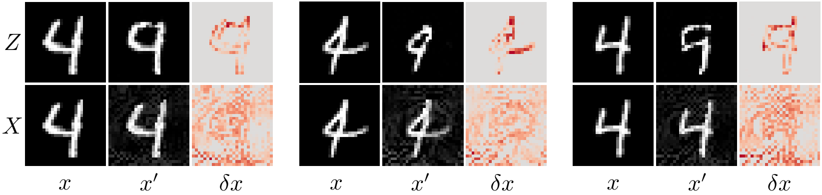
$$\gamma^{-1} = \frac{\partial g}{\partial z}\frac{\partial g}{\partial z}^T = U\,\Sigma^2\,U^T$$

- For N dimensional data manifold: N large singular values
- Corresponding left-singular vectors span the tangent space of the data manifold
- For toy data:



Experiments with MNIST

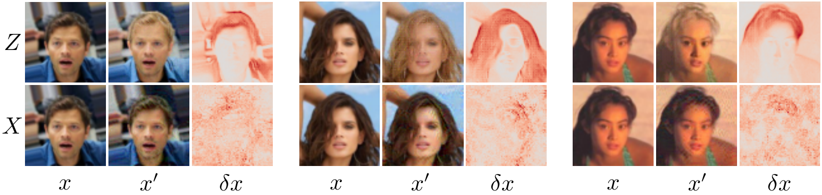
- Classifier: CNN with 10 classes (test accuracy: 99%)
- Flow: RealNVP
- Task: Change classification of 4 to 9



- Top row: Counterfactual computed in base space
- Bottom row: Adersarial example computed in data space

Experiments with CelebA

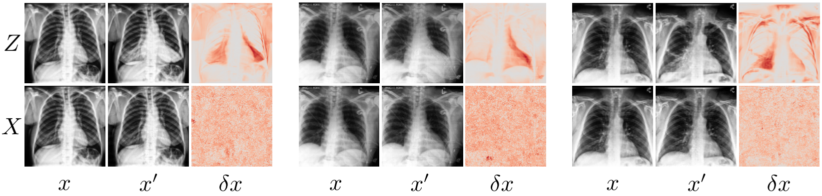
- Classifier: Binary CNN trained on ***blonde/not blonde*** attribute (test accuracy: 94%)
- Flow: Glow
- Task: Change classification from ***not blonde*** to ***blonde***



- Top row: Counterfactual computed in base space
- Bottom row: Adersarial example computed in data space

Experiments with CheXpert

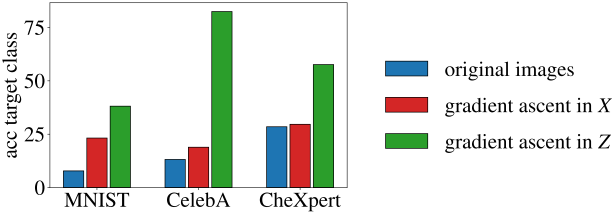
- Classifier: Binary CNN trained on ***cardiomegaly/healthy*** attribute (test accuracy: 86%)
- Flow: Glow
- Task: Change classification from ***healthy*** to ***cardiomegaly***



- Top row: Counterfactual computed in base space
- Bottom row: Adersarial example computed in data space

Quantitative evaluation

- Compare classification accuracies for different samples with linear SVMs



- Gradient ascent in Z produces higher probability in the target class

  $\Rightarrow$ Optimization in Z leads to better generalization

Conclusion

Summary

- Counterfactual examples have a different classification than the original sample and lie on the data manifold
- However, a gradient ascent optimization leads to adversarial examples which lie off the data manifold
- Normalizing flows are bijective generative models
- A gradient ascent optimization in the base space of a normalizing flow stays on the data manifold and leads to counterfactuals
- The reason is that the induced metric makes the learning rate in orthogonal directions small

Future questions

- Other ways to quantify the quality of counterfactuals?
- Can one replace the normalizing flow by a GAN?
- Can one learn something about the invertibility of normalizing flows using this construction?