Variational Mixture of Normalizing Flows

Guilherme Grijó Pen Freitas Pires

November 21, 2019

Thesis to obtain the Master of Science degree in
Electrical and Computer Engineering

Supervisor: Prof. Mário A. T. Figueiredo

TÉCNICO LISBOA

Table of Contents

- Goal of this work: **mixture of flexible distributions**.

Introduction and Motivation

- Goal of this work: **mixture of flexible distributions**.
- Two questions:

Introduction and Motivation

- Goal of this work: **mixture of flexible distributions**.
- Two questions:
  - What should the **mixture components** be?

Introduction and Motivation

- Goal of this work: **mixture of flexible distributions**.
- Two questions:
  - What should the **mixture components** be?
  - How should their **parameters** be **estimated**?

- Deep generative models: an active area of research

Introduction and Motivation

- Deep generative models: an active area of research
  - **Implicit distributions**: Generative adversarial networks [Goodfellow et al., 2014], Variational Autoencoder [Kingma and Welling, 2014]
    - No explicit access to the density function

Introduction and Motivation

- Deep generative models: an active area of research
  - **Implicit distributions**: Generative adversarial networks [Goodfellow et al., 2014], Variational Autoencoder [Kingma and Welling, 2014]
    - No explicit access to the density function
  - **Explicit distributions**: Normalizing flows [Rezende and Mohamed, 2015]
    - Explicit access to the density function
    - No approach to introduce discrete structure (multi-modality)

TÉCNICO LISBOA

Outline

- Mixture Models

Outline

- Mixture Models
- Normalizing Flows

Outline

- Mixture Models
- Normalizing Flows
    - The chosen family for the mixture model components

Outline

- Mixture Models
- Normalizing Flows
  - The chosen family for the mixture model components
- Variational Inference

Outline

- Mixture Models
- Normalizing Flows
  - The chosen family for the mixture model components
- Variational Inference
  - The chosen framework for estimating the parameters of the proposed model

TÉCNICO LISBOA

Outline

- Mixture Models
- Normalizing Flows
    - The chosen family for the mixture model components
- Variational Inference
    - The chosen framework for estimating the parameters of the proposed model
- Variational Mixture of Normalizing Flows

Outline

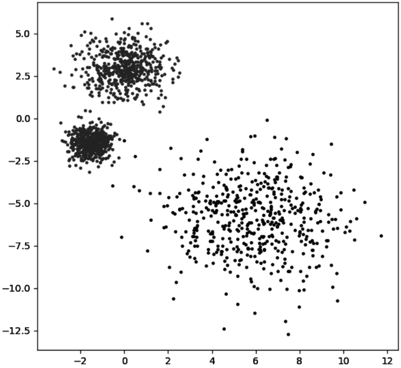
- Mixture Models
- Normalizing Flows
  - The chosen family for the mixture model components
- Variational Inference
  - The chosen framework for estimating the parameters of the proposed model
- Variational Mixture of Normalizing Flows
- Experiments and results

Outline

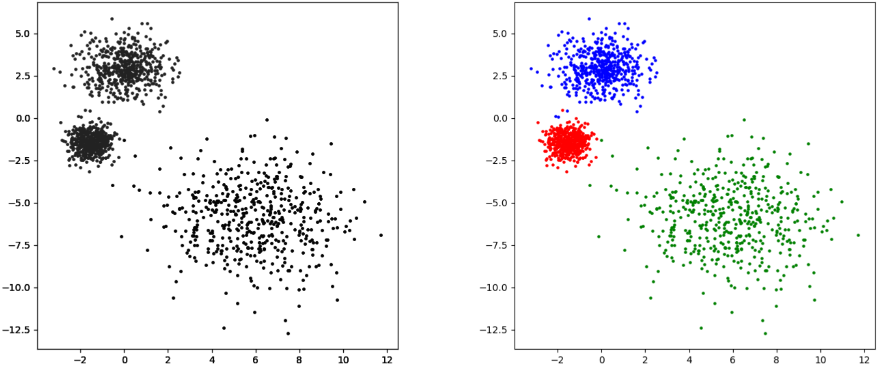
- Mixture Models
- Normalizing Flows
  - The chosen family for the mixture model components
- Variational Inference
  - The chosen framework for estimating the parameters of the proposed model
- Variational Mixture of Normalizing Flows
- Experiments and results
- Conclusions and future work

Table of Contents

Mixture Models: Mixture of Gaussians



Mixture Models: Mixture of Gaussians



Mixture Models: Definition

- Mixture model: used to model data that contains **subgroups**.

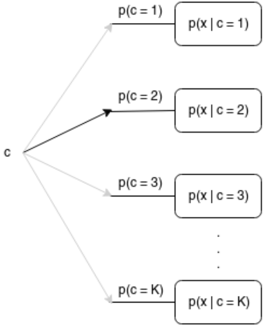
TÉCNICO LISBOA

Mixture Models: Definition

- Mixture model: used to model data that contains **subgroups**.
- "Subgroup-conditional" distributions (typically) in the same family

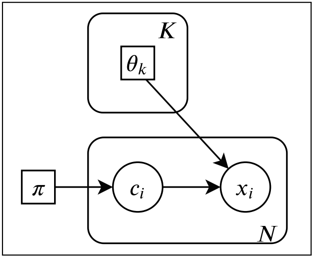
Mixture Models: Definition

- Mixture model: used to model data that contains **subgroups**.
- "Subgroup-conditional" distributions (typically) in the same family



Mixture Models: Joint

For N data points, $X = \{\boldsymbol{x_i} : i = 1, 2, ..., N\}$, and hidden variables $C = \{c_i : i = 1, 2, ..., N\}$



$$p(\boldsymbol{x}, \boldsymbol{c}) =$$

$$= \prod_{i=1}^{N} \sum_{k=1}^{K} p_c(c_i = k) p_{\boldsymbol{x}|c}(\boldsymbol{x_i} | c_i = k, \theta_k)$$

$$= \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k p_{\boldsymbol{x}|c}(\boldsymbol{x_i} | c_i = k, \theta_k)$$

TÉCNICO LISBOA

Mixture Models: Difficult case

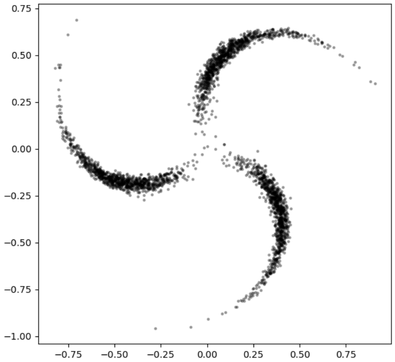


Table of Contents

Normalizing Flows: Change of Variables

Given

$$\begin{cases} \boldsymbol{z} \sim p(\boldsymbol{z}) \\ \boldsymbol{x} = g(\boldsymbol{z}; \boldsymbol{\theta}) \end{cases}$$

Normalizing Flows: Change of Variables

Given

$$\begin{cases} \boldsymbol{z} \sim p(\boldsymbol{z}) \\ \boldsymbol{x} = g(\boldsymbol{z}; \boldsymbol{\theta}) \end{cases}$$

then:

$$\begin{aligned} f_X(\boldsymbol{x}) &= f_Z(g^{-1}(\boldsymbol{x}; \boldsymbol{\theta})) \left| \det \left(\frac{d}{d\boldsymbol{x}} g^{-1}(\boldsymbol{x}; \boldsymbol{\theta}) \right) \right| \\ &= f_Z(g^{-1}(\boldsymbol{x}; \boldsymbol{\theta})) \left| \det \left(\frac{d}{d\boldsymbol{z}} g(\boldsymbol{z}; \boldsymbol{\theta}) \Big|_{\boldsymbol{z} = g^{-1}(\boldsymbol{x}; \boldsymbol{\theta})} \right) \right|^{-1} \end{aligned}$$

TÉCNICO LISBOA

Normalizing Flows: Change of Variables

Given

$$\begin{cases} \boldsymbol{z} \sim p(\boldsymbol{z}) \\ \boldsymbol{x} = g(\boldsymbol{z}; \boldsymbol{\theta}) \end{cases}$$

then:

$$\begin{aligned} f_X(\boldsymbol{x}) &= f_Z(g^{-1}(\boldsymbol{x}; \boldsymbol{\theta})) \left| \det \left(\frac{d}{d\boldsymbol{x}} g^{-1}(\boldsymbol{x}; \boldsymbol{\theta}) \right) \right| \\ &= f_Z(g^{-1}(\boldsymbol{x}; \boldsymbol{\theta})) \left| \det \left(\frac{d}{d\boldsymbol{z}} g(\boldsymbol{z}; \boldsymbol{\theta}) \Big|_{\boldsymbol{z}=g^{-1}(\boldsymbol{x};\boldsymbol{\theta})} \right) \right|^{-1} \end{aligned}$$

This can be **optimized w.r.t. θ**, to approximate an **arbitrary distribution**

Normalizing Flows: Change of Variables

Requirements for feasibility

Normalizing Flows: Change of Variables

Requirements for feasibility

a. Base density - **closed form** and **easy to sample** from

Normalizing Flows: Change of Variables

Requirements for feasibility

a. Base density - **closed form** and **easy to sample** from

b. **Determinant** of the **Jacobian** of g - computationally cheap

Requirements for feasibility

a. Base density - **closed form** and **easy to sample** from

b. **Determinant** of the **Jacobian** of g - computationally cheap

c. **Gradient** of $\det\left(\frac{d}{d\boldsymbol{z}} g(\boldsymbol{z};\boldsymbol{\theta})\right)$ w.r.t $\boldsymbol{\theta}$ - computationally cheap

Normalizing Flows: Change of Variables

- Normalizing Flows: **composition** of several “good” transformations

Normalizing Flows: Change of Variables

- Normalizing Flows: **composition** of several "good" transformations

- I.e., $g = h_{L-1} \circ h_{L-2} \circ \ldots \circ h_1 \circ h_0$

Normalizing Flows: Change of Variables

- Normalizing Flows: **composition** of several "good" transformations
- I.e., $g = h_{L-1} \circ h_{L-2} \circ \ldots \circ h_1 \circ h_0$
- Applying the formula to g, and taking the logarithm:

$$\log f_X(\boldsymbol{x}) = \log f_Z(g^{-1}(\boldsymbol{x})) - \sum_{\ell=0}^{L-1} \log \left| \det \left(\frac{d}{d\boldsymbol{x}_\ell} h_\ell(\boldsymbol{x}_\ell) \right) \right|.$$

TÉCNICO LISBOA

Normalizing Flows: Affine Coupling Layer

- An example: Affine Coupling Layer [Dinh, Sohl-Dickstein, and Bengio, 2017]

Normalizing Flows: Affine Coupling Layer

- An example: Affine Coupling Layer [Dinh, Sohl-Dickstein, and Bengio, 2017]
- Splitting $\boldsymbol{z}$ into $(\boldsymbol{z_1}, \boldsymbol{z_2})$,

Normalizing Flows: Affine Coupling Layer

- An example: Affine Coupling Layer [Dinh, Sohl-Dickstein, and Bengio, 2017]
- Splitting $\boldsymbol{z}$ into $(\boldsymbol{z_1}, \boldsymbol{z_2})$,

$$\begin{cases} \boldsymbol{x_1} & = \boldsymbol{z_1} \odot \exp\left(s(\boldsymbol{z_2})\right) + t(\boldsymbol{z_2}) \\ \boldsymbol{x_2} & = \boldsymbol{z_2}. \end{cases}$$

Normalizing Flows: Affine Coupling Layer

- An example: Affine Coupling Layer [Dinh, Sohl-Dickstein, and Bengio, 2017]
- Splitting $\boldsymbol{z}$ into $(\boldsymbol{z}_1, \boldsymbol{z}_2)$,

$$\begin{cases} \boldsymbol{x}_1 & = \boldsymbol{z}_1 \odot \exp\left(s(\boldsymbol{z}_2)\right) + t(\boldsymbol{z}_2) \\ \boldsymbol{x}_2 & = \boldsymbol{z}_2. \end{cases}$$

- The respective Jacobian matrix:

$$J_{f(z)} = \begin{bmatrix} \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{z}_1} & \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{z}_2} \\ \frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{z}_1} & \frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{z}_2} \end{bmatrix} = \begin{bmatrix} \operatorname{diag}\Big(\exp\left(s(\boldsymbol{z}_2)\right)\Big) & \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{z}_2} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$$

Table of Contents

Variational Inference: Preamble

- Joint probability distribution $p(\boldsymbol{x}, \boldsymbol{c})$.

Variational Inference: Preamble

- Joint probability distribution $p(\boldsymbol{x}, \boldsymbol{c})$.
- $\boldsymbol{x}$ is observed and $\boldsymbol{c}$ is latent.

Variational Inference: Preamble

- Joint probability distribution $p(\boldsymbol{x}, \boldsymbol{c})$.
- $\boldsymbol{x}$ is observed and $\boldsymbol{c}$ is latent.
- Inference about $\boldsymbol{c}$, given $\boldsymbol{x}$, by **Bayes' Law**:

$$
\begin{aligned}
p(\boldsymbol{c}|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|\boldsymbol{c})p(\boldsymbol{c})}{p(\boldsymbol{x})} \\
&= \frac{p(\boldsymbol{x}|\boldsymbol{c})p(\boldsymbol{c})}{\int p(\boldsymbol{x}|\boldsymbol{c}')p(\boldsymbol{c}')d\boldsymbol{c}'}
\end{aligned}
$$

Variational Inference: Preamble

- Joint probability distribution $p(\boldsymbol{x}, \boldsymbol{c})$.
- $\boldsymbol{x}$ is observed and $\boldsymbol{c}$ is latent.
- Inference about $\boldsymbol{c}$, given $\boldsymbol{x}$, by **Bayes' Law**:

$$\begin{aligned} p(\boldsymbol{c}|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|\boldsymbol{c})p(\boldsymbol{c})}{p(\boldsymbol{x})} \\ &= \frac{p(\boldsymbol{x}|\boldsymbol{c})p(\boldsymbol{c})}{\int p(\boldsymbol{x}|\boldsymbol{c}')p(\boldsymbol{c}')d\boldsymbol{c}'} \end{aligned}$$

- Problem: The integral is normally **intractable**

Variational Inference: Preamble

- Joint probability distribution $p(\boldsymbol{x}, \boldsymbol{c})$.
- $\boldsymbol{x}$ is observed and $\boldsymbol{c}$ is latent.
- Inference about $\boldsymbol{c}$, given $\boldsymbol{x}$, by **Bayes' Law**:

$$
\begin{aligned}
p(\boldsymbol{c}|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|\boldsymbol{c})p(\boldsymbol{c})}{p(\boldsymbol{x})} \\
&= \frac{p(\boldsymbol{x}|\boldsymbol{c})p(\boldsymbol{c})}{\int p(\boldsymbol{x}|\boldsymbol{c'})p(\boldsymbol{c'})d\boldsymbol{c'}}
\end{aligned}
$$

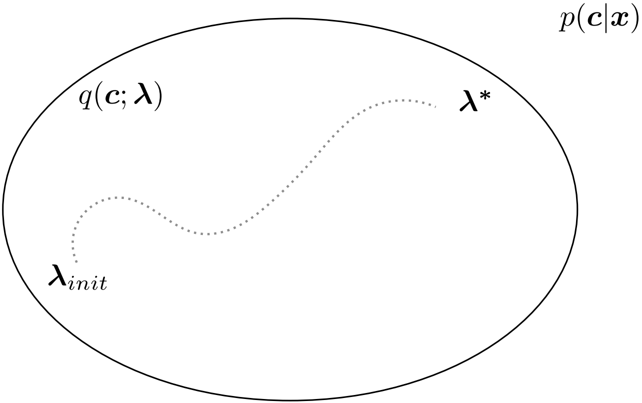
- Problem: The integral is normally **intractable**
  - **Variational inference**: an **approximate inference** framework to overcome this intractability.

Variational Inference: Goal

Given a family $q(\boldsymbol{c}; \boldsymbol{\lambda})$, find the parameters $\boldsymbol{\lambda}^*$ such that:

$$\boldsymbol{\lambda}^* = \operatorname*{argmin}_{\boldsymbol{\lambda}} KL(q(\boldsymbol{c}; \boldsymbol{\lambda}) || p(\boldsymbol{c}|\boldsymbol{x}))$$

Variational Inference: Goal



Variational Inference: ELBO

$$KL(q(\boldsymbol{c})||p(\boldsymbol{c}|\boldsymbol{x})) = \int q(\boldsymbol{c}) \log \frac{q(\boldsymbol{c})}{p(\boldsymbol{c}|\boldsymbol{x})} d\boldsymbol{c}$$

$$
\begin{aligned}
KL(q(\boldsymbol{c})||p(\boldsymbol{c}|\boldsymbol{x})) &= \int q(\boldsymbol{c}) \log \frac{q(\boldsymbol{c})}{p(\boldsymbol{c}|\boldsymbol{x})} d\boldsymbol{c} \\
&= \int q(\boldsymbol{c})(\log q(\boldsymbol{c}) - \log p(\boldsymbol{c}|\boldsymbol{x}))d\boldsymbol{c}
\end{aligned}
$$

Variational Inference: ELBO

$$
\begin{aligned}
KL(q(\boldsymbol{c})||p(\boldsymbol{c}|\boldsymbol{x})) &= \int q(\boldsymbol{c}) \log \frac{q(\boldsymbol{c})}{p(\boldsymbol{c}|\boldsymbol{x})} d\boldsymbol{c} \\
&= \int q(\boldsymbol{c})(\log q(\boldsymbol{c}) - \log p(\boldsymbol{c}|\boldsymbol{x})) d\boldsymbol{c} \\
&= \int q(\boldsymbol{c})(\log q(\boldsymbol{c}) - (\log p(\boldsymbol{x}, \boldsymbol{c}) - \log p(\boldsymbol{x}))) d\boldsymbol{c}
\end{aligned}
$$

Variational Inference: ELBO

$$
\begin{aligned}
KL(q(\boldsymbol{c})||p(\boldsymbol{c}|\boldsymbol{x})) &= \int q(\boldsymbol{c}) \log \frac{q(\boldsymbol{c})}{p(\boldsymbol{c}|\boldsymbol{x})} d\boldsymbol{c} \\
&= \int q(\boldsymbol{c})(\log q(\boldsymbol{c}) - \log p(\boldsymbol{c}|\boldsymbol{x}))d\boldsymbol{c} \\
&= \int q(\boldsymbol{c})(\log q(\boldsymbol{c}) - (\log p(\boldsymbol{x},\boldsymbol{c}) - \log p(\boldsymbol{x})))d\boldsymbol{c} \\
&= \mathbb{E}_q[\log q(\boldsymbol{c})] - \mathbb{E}_q[\log p(\boldsymbol{x},\boldsymbol{c})] + \log p(\boldsymbol{x})
\end{aligned}
$$

Variational Inference: ELBO

$$KL(q(\boldsymbol{c})||p(\boldsymbol{c}|\boldsymbol{x})) + \mathbb{E}_q[\log p(\boldsymbol{x}, \boldsymbol{c})] - \mathbb{E}_q[\log q(\boldsymbol{c})] = \log p(\boldsymbol{x})$$

Variational Inference: ELBO

$$\overbrace{KL(q(\boldsymbol{c})||p(\boldsymbol{c}|\boldsymbol{x}))}^{\geqslant 0} + \mathbb{E}_q[\log p(\boldsymbol{x}, \boldsymbol{c})] - \mathbb{E}_q[\log q(\boldsymbol{c})] = \log p(\boldsymbol{x})$$

$$\overbrace{KL(q(\boldsymbol{c})||p(\boldsymbol{c}|\boldsymbol{x}))}^{\geqslant 0} + \mathbb{E}_q[\log p(\boldsymbol{x}, \boldsymbol{c})] - \mathbb{E}_q[\log q(\boldsymbol{c})] = \log p(\boldsymbol{x})$$

$$\mathbb{E}_q[\log p(\boldsymbol{x}, \boldsymbol{c})] - \mathbb{E}_q[\log q(\boldsymbol{c})] \leqslant \log p(\boldsymbol{x})$$

$$\overbrace{KL(q(\boldsymbol{c})||p(\boldsymbol{c}|\boldsymbol{x}))}^{\geqslant 0} + \mathbb{E}_q[\log p(\boldsymbol{x}, \boldsymbol{c})] - \mathbb{E}_q[\log q(\boldsymbol{c})] = \log p(\boldsymbol{x})$$
$$\mathbb{E}_q[\log p(\boldsymbol{x}, \boldsymbol{c})] - \mathbb{E}_q[\log q(\boldsymbol{c})] \leqslant \log p(\boldsymbol{x})$$

$$\begin{aligned}\mathrm{ELBO}(q) &= \mathbb{E}_q[\log p(\boldsymbol{x}, \boldsymbol{c})] - \mathbb{E}_q[\log q(\boldsymbol{c})] \\ &= \mathbb{E}_q[\log p(\boldsymbol{x}|\boldsymbol{c})] + \mathbb{E}_q[\log p(\boldsymbol{c})] - \mathbb{E}_q[\log q(\boldsymbol{c})]\end{aligned}$$

Table of Contents

VMoNF: Introduction

Is it possible to **combine** the ideas from the previous sections, to obtain a mixture of flexible models?

VMoNF: Definition

- Mixture of K normalizing flows

VMoNF: Definition

- Mixture of K normalizing flows
- Variable c_i selects the component for sample $\boldsymbol{x}_i$

VMoNF: Definition

- Mixture of K normalizing flows
- Variable c_i selects the component for sample $\boldsymbol{x}_i$
- $p(c|\boldsymbol{x})$ is unknown.

VMoNF: Definition

- Mixture of K normalizing flows
- Variable c_i selects the component for sample $\boldsymbol{x}_i$
- $p(c|\boldsymbol{x})$ is unknown.
  - Approximate it with $q(c|\boldsymbol{x})$: **neural network**

VMoNF: Definition

- Mixture of K normalizing flows
- Variable c_i selects the component for sample $\boldsymbol{x}_i$
- $p(c|\boldsymbol{x})$ is unknown.
  - Approximate it with $q(c|\boldsymbol{x})$: **neural network**
- Recall $ELBO(q) = \mathbb{E}_q[\log p(\boldsymbol{x}|c)] + \mathbb{E}_q[\log p(c)] - \mathbb{E}_q[\log q()]$

TÉCNICO LISBOA

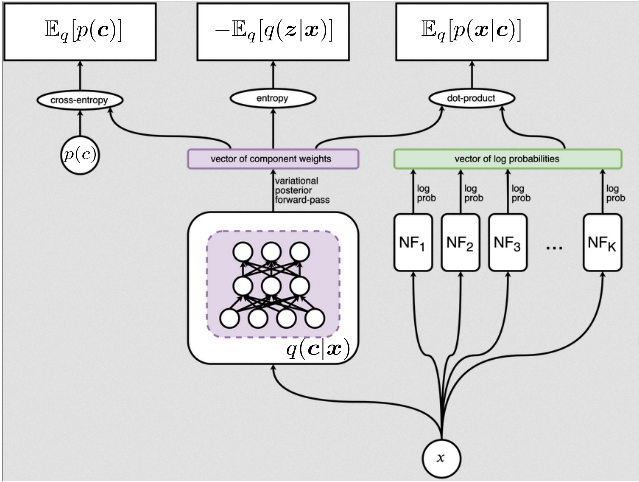
VMoNF: Definition

- Mixture of K normalizing flows
- Variable c_i selects the component for sample $\boldsymbol{x}_i$
- $p(c|\boldsymbol{x})$ is unknown.
  - Approximate it with $q(c|\boldsymbol{x})$: **neural network**
- Recall $ELBO(q) = \mathbb{E}_q[\log p(\boldsymbol{x}|c)] + \mathbb{E}_q[\log p(c)] - \mathbb{E}_q[\log q()]$
- The components $p(\boldsymbol{x}|c)$ are **normalizing flows**

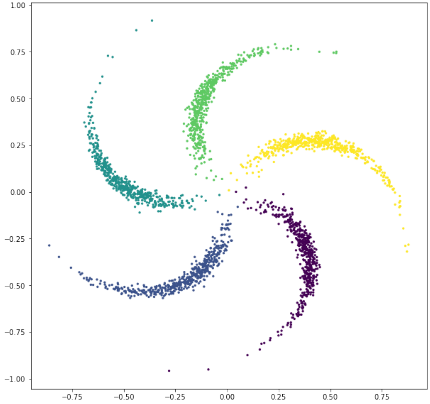
VMoNF: Definition

- Mixture of K normalizing flows
- Variable c_i selects the component for sample $\boldsymbol{x}_i$
- $p(c|\boldsymbol{x})$ is unknown.
  - Approximate it with $q(c|\boldsymbol{x})$: **neural network**
- Recall $ELBO(q) = \mathbb{E}_q[\log p(\boldsymbol{x}|c)] + \mathbb{E}_q[\log p(c)] - \mathbb{E}_q[\log q()]$
- The components $p(\boldsymbol{x}|c)$ are **normalizing flows**
- Optimize the ELBO, by **jointly** learning the variational posterior and the generative components.

VMoNF: Overview

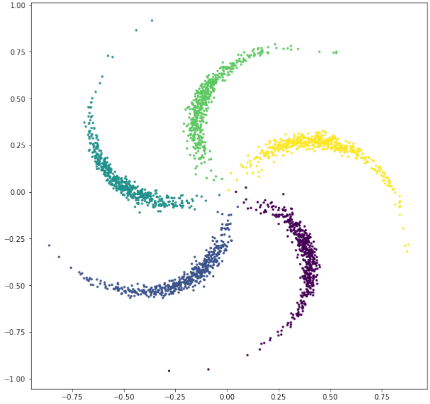


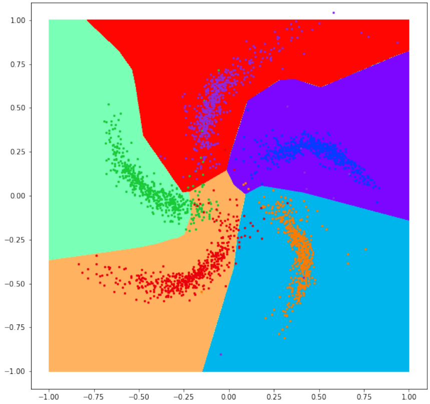
VMoNF: Experiments - Pinwheel (5 wings)



TÉCNICO LISBOA

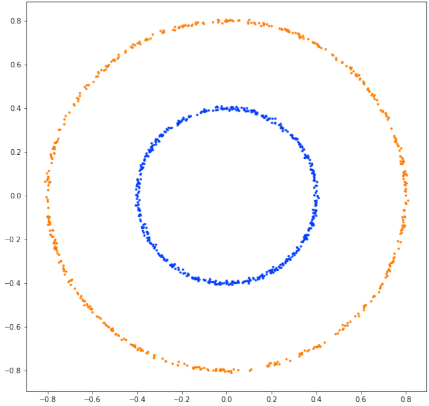
VMoNF: Experiments - Pinwheel (5 wings)



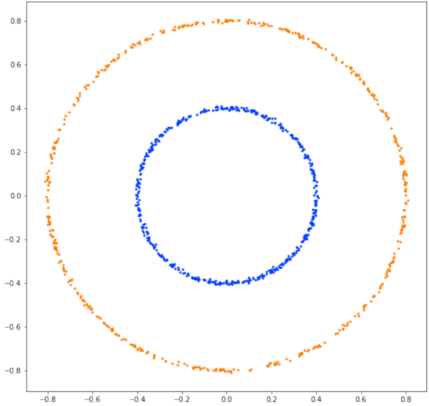


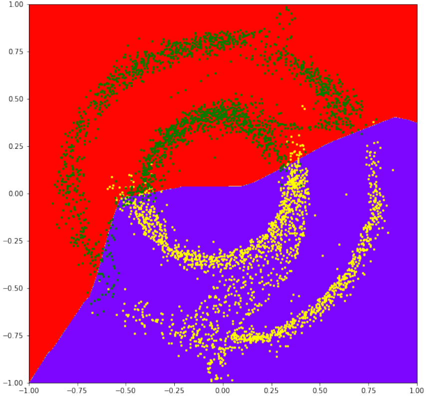
Trainining Animation

VMoNF: Experiments - 2 Circles

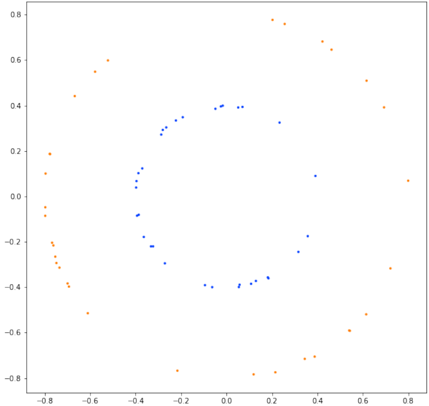


VMoNF: Experiments - 2 Circles

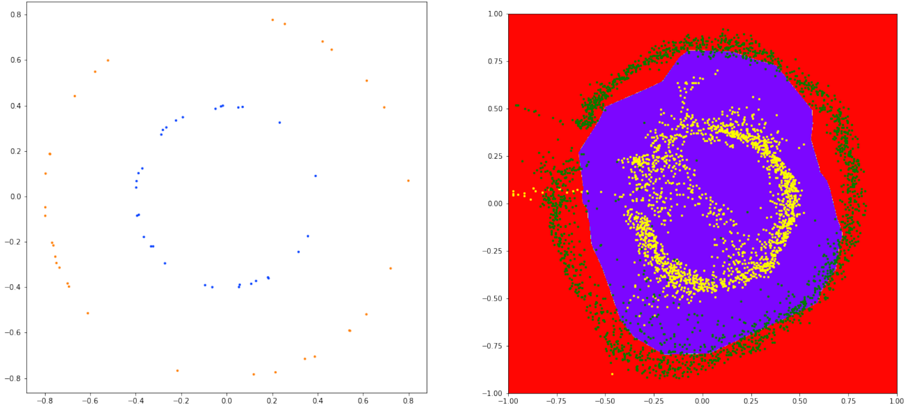




VMoNF: Experiments - 2 Circles (semi supervised)



VMoNF: Experiments - 2 Circles (semi supervised)



Note: 32 labeled points, 1024 unlabeled points

VMoNF: Experiments - MNIST

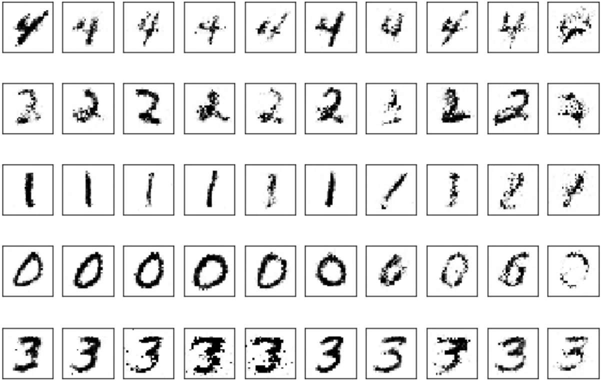


Table of Contents

Conclusions and Future Work

- Similar work is being pursued and published in prominent venues: [Dinh, Sohl-Dickstein, et al., 2019], [Izmailov et al., 2019]

Conclusions and Future Work

- Similar work is being pursued and published in prominent venues: [Dinh, Sohl-Dickstein, et al., 2019], [Izmailov et al., 2019]

- Investigate the effect of a consistency loss regularization term

Conclusions and Future Work

- Similar work is being pursued and published in prominent venues: [Dinh, Sohl-Dickstein, et al., 2019], [Izmailov et al., 2019]
- Investigate the effect of a consistency loss regularization term
- Weight-sharing between components

Conclusions and Future Work

- Similar work is being pursued and published in prominent venues: [Dinh, Sohl-Dickstein, et al., 2019], [Izmailov et al., 2019]
- Investigate the effect of a consistency loss regularization term
- Weight-sharing between components
- Balance between complexities

Conclusions and Future Work

- Similar work is being pursued and published in prominent venues: [Dinh, Sohl-Dickstein, et al., 2019], [Izmailov et al., 2019]
- Investigate the effect of a consistency loss regularization term
- Weight-sharing between components
- Balance between complexities
- (Controlled) component anihilation

Thank you!