

Variational Mixture of Normalizing Flows

Guilherme Grijó Pen Freitas Pires

November 22, 2019

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

Supervisor: Prof. Mário A. T. Figueiredo



Table of Contents

- 1 Introduction and Motivation
- 2 Mixture Models
- 3 Normalizing Flows
- 4 Variational Inference
- 5 Variational Mixture of Normalizing Flows
- 6 Conclusions

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

Introduction and Motivation

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

- 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

- 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

- 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)

- Mixture Models

Outline

- Mixture Models
- Normalizing Flows

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

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

- 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

- 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

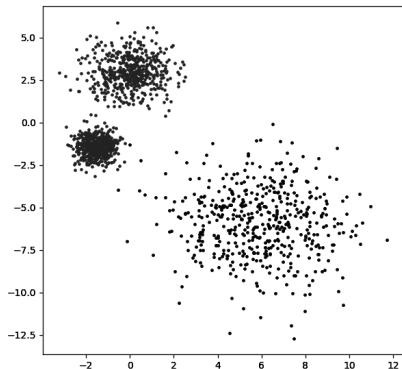
- Mixture Models
- Normalizing Flows
 - The chosen family for the mixture model components
- Variational Inference
 - The chosen framework for estimating the parameters of the proposed model
- Variational Mixture of Normalizing Flows
- Experiments and results

- 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

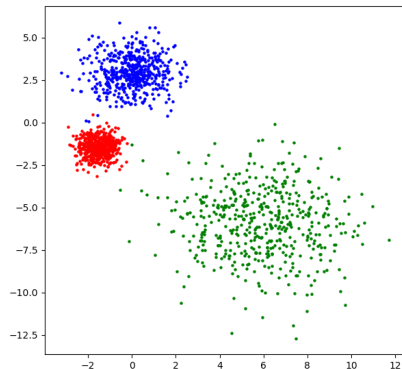
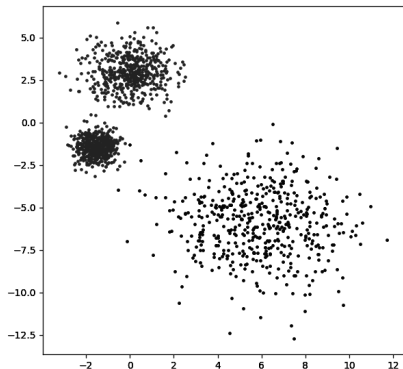
Table of Contents

- 1 Introduction and Motivation
- 2 Mixture Models**
- 3 Normalizing Flows
- 4 Variational Inference
- 5 Variational Mixture of Normalizing Flows
- 6 Conclusions

Mixture Models: Mixture of Gaussians



Mixture Models: Mixture of Gaussians



Mixture Models: Definition

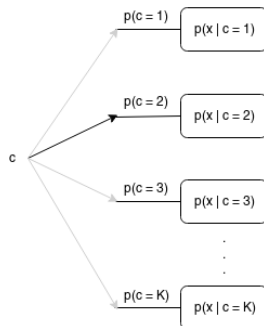
- Mixture model: used to model data that contains **subgroups**.

Mixture Models: Definition

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

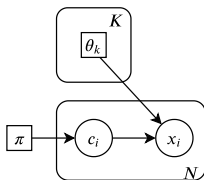
Mixture Models: Definition

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



Mixture Models: Joint

For N data points, $X = \{\mathbf{x}_i : i = 1, 2, \dots, N\}$, and hidden variables $C = \{c_i : i = 1, 2, \dots, N\}$



$$\begin{aligned} p(\mathbf{x}, \mathbf{c}) &= \\ &= \prod_{i=1}^N \sum_{k=1}^K p_c(c_i = k) p_{\mathbf{x}|\mathbf{c}}(\mathbf{x}_i | c_i = k, \theta_k) \\ &= \prod_{i=1}^N \sum_{k=1}^K \pi_k p_{\mathbf{x}|\mathbf{c}}(\mathbf{x}_i | c_i = k, \theta_k) \end{aligned}$$

Mixture Models: Difficult case

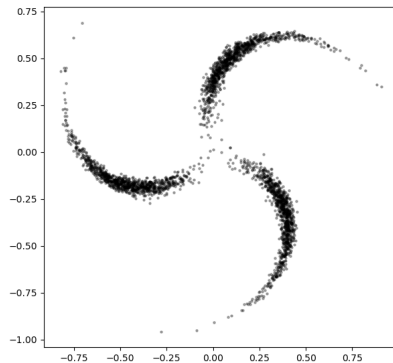


Table of Contents

- 1 Introduction and Motivation
- 2 Mixture Models
- 3 Normalizing Flows**
- 4 Variational Inference
- 5 Variational Mixture of Normalizing Flows
- 6 Conclusions

Normalizing Flows: Change of Variables

Given

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

Normalizing Flows: Change of Variables

Given

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

then:

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

Normalizing Flows: Change of Variables

Given

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

then:

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

This can be **optimized w.r.t. $\boldsymbol{\theta}$** , to approximate an **arbitrary distribution**

Normalizing Flows: Change of Variables

Requirements for feasibility

Normalizing Flows: Change of Variables

Requirements for feasibility

- Ⓐ 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

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
- c **Gradient** of $\det \left(\frac{d}{dz} g(\mathbf{z}; \boldsymbol{\theta}) \right)$ 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., $g = h_{L-1} \circ h_{L-2} \circ \dots \circ h_1 \circ h_0$

Normalizing Flows: Change of Variables

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

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

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 \mathbf{z} into $(\mathbf{z}_1, \mathbf{z}_2)$,

Normalizing Flows: Affine Coupling Layer

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

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

Normalizing Flows: Affine Coupling Layer

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

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

- The respective Jacobian matrix:

$$J_{f(\mathbf{z})} = \begin{bmatrix} \frac{\partial \mathbf{x}_1}{\partial \mathbf{z}_1} & \frac{\partial \mathbf{x}_1}{\partial \mathbf{z}_2} \\ \frac{\partial \mathbf{x}_2}{\partial \mathbf{z}_1} & \frac{\partial \mathbf{x}_2}{\partial \mathbf{z}_2} \end{bmatrix} = \begin{bmatrix} \text{diag}(\exp(s(\mathbf{z}_2))) & \frac{\partial \mathbf{x}_1}{\partial \mathbf{z}_2} \\ \mathbf{0} & I \end{bmatrix}$$

Table of Contents

- 1 Introduction and Motivation
- 2 Mixture Models
- 3 Normalizing Flows
- 4 Variational Inference**
- 5 Variational Mixture of Normalizing Flows
- 6 Conclusions

Variational Inference: Preamble

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

Variational Inference: Preamble

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

Variational Inference: Preamble

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

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

Variational Inference: Preamble

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

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

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

Variational Inference: Preamble

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

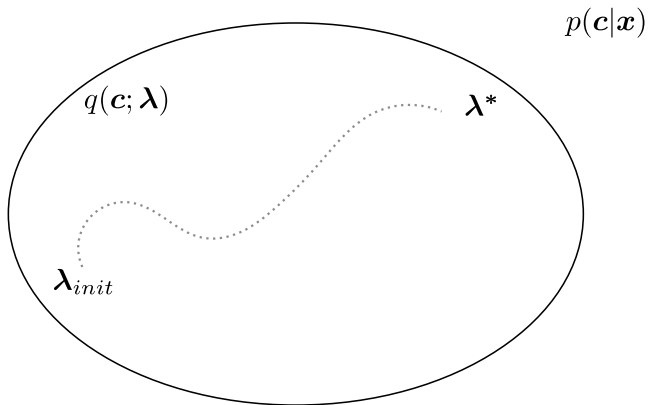
$$\begin{aligned} p(\mathbf{c}|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathbf{c})p(\mathbf{c})}{p(\mathbf{x})} \\ &= \frac{p(\mathbf{x}|\mathbf{c})p(\mathbf{c})}{\int p(\mathbf{x}|\mathbf{c}')p(\mathbf{c}')d\mathbf{c}'} \end{aligned}$$

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

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

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

Variational Inference: Goal



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

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

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

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

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

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

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

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

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

Table of Contents

- 1 Introduction and Motivation
- 2 Mixture Models
- 3 Normalizing Flows
- 4 Variational Inference
- 5 Variational Mixture of Normalizing Flows**
- 6 Conclusions

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 \mathbf{x}_i

VMoNF: Definition

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

VMoNF: Definition

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

VMoNF: Definition

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

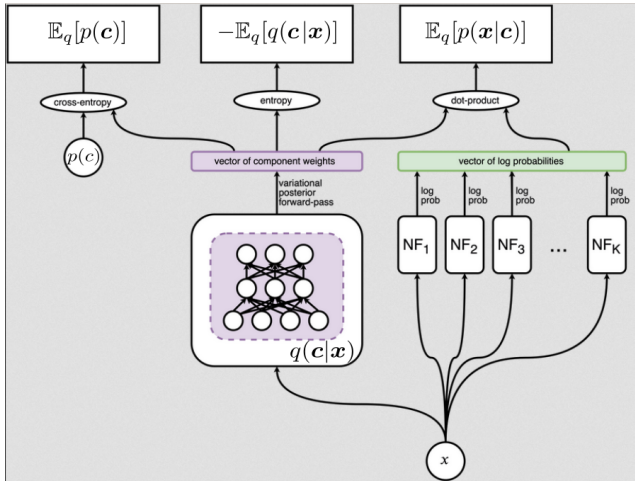
VMoNF: Definition

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

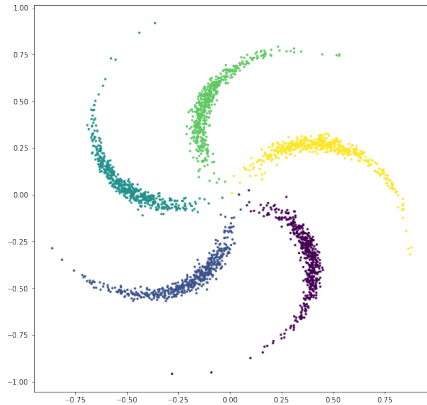
VMoNF: Definition

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

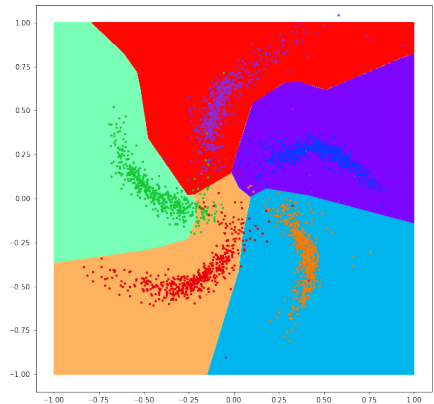
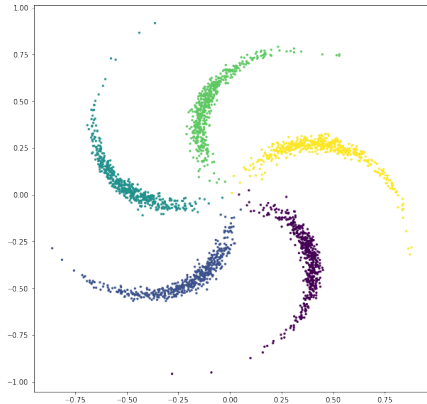
VMoNF: Overview



VMoNF: Experiments - Pinwheel (5 wings)

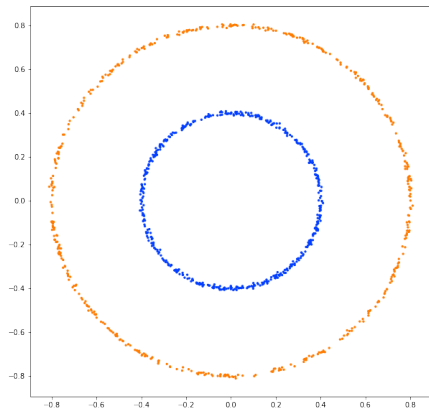


VMoNF: Experiments - Pinwheel (5 wings)

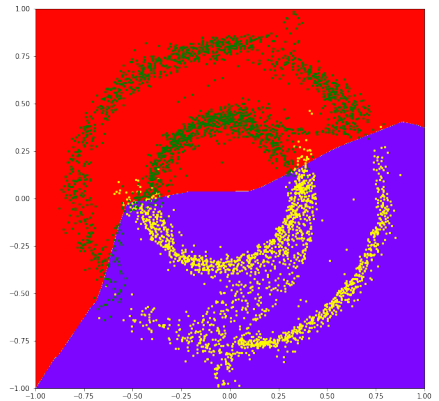
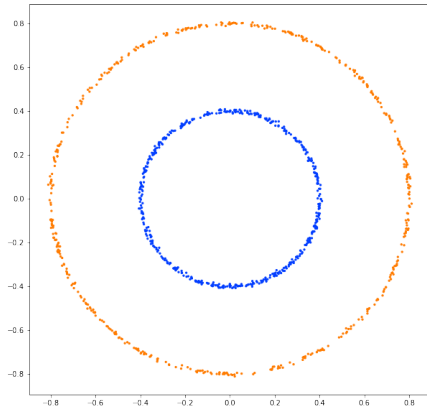


Training Animation

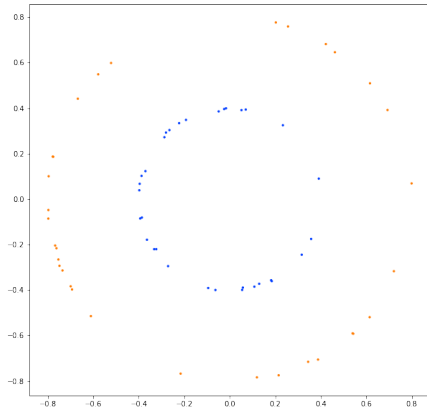
VMoNF: Experiments - 2 Circles



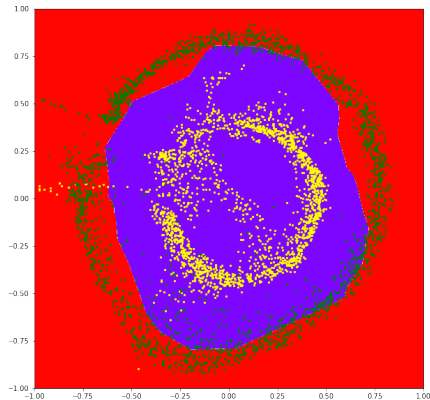
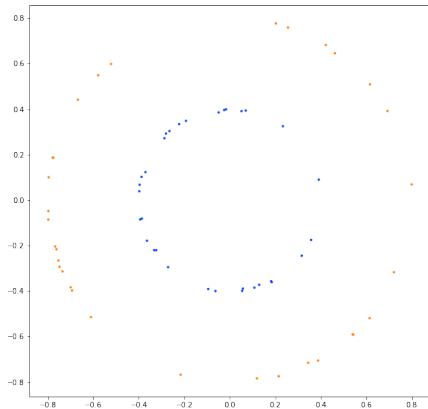
VMoNF: Experiments - 2 Circles



VMoNF: Experiments - 2 Circles (semi supervised)



VMoNF: Experiments - 2 Circles (semi supervised)



Note: 32 labeled points, 1024 unlabeled points

VMoNF: Experiments - MNIST

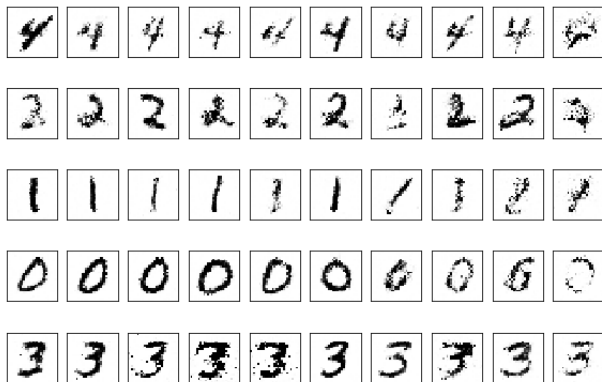


Table of Contents

- 1 Introduction and Motivation
- 2 Mixture Models
- 3 Normalizing Flows
- 4 Variational Inference
- 5 Variational Mixture of Normalizing Flows
- 6 Conclusions**

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 annihilation

Thank you!