Generative Modeling of Sequential Data

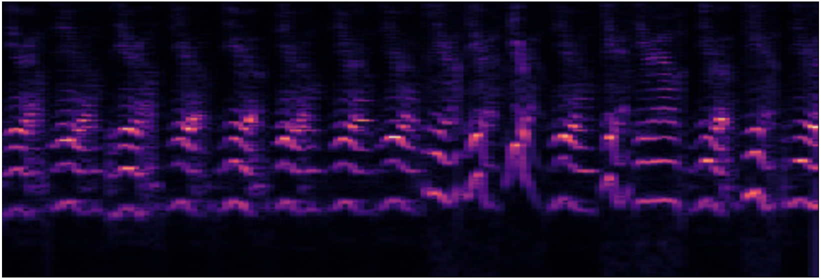
Cem Subakan

University of Illinois at Urbana-Champaign

April 6'th, 2018

What is Generative Modeling?

What is Generative Modeling?



What is Generative Modeling, and Why?

- Learning by assuming a generative process
  - E.g. fitting a multivariate Gaussian, mixture model, NMF, etc.
- Short answer to why question: Extracting structure out of data, understanding data

What is Generative Modeling, and Why?

- Learning by assuming a generative process
  - E.g. fitting a multivariate Gaussian, mixture model, NMF, etc.
- Short answer to why question: Extracting structure out of data, understanding data
  - Clustering (mixture models)

What is Generative Modeling, and Why?

- Learning by assuming a generative process
  - E.g. fitting a multivariate Gaussian, mixture model, NMF, etc.
- Short answer to why question: Extracting structure out of data, understanding data
  - Clustering (mixture models)
  - Document Analysis (LDA)

What is Generative Modeling, and Why?

- Learning by assuming a generative process
  - E.g. fitting a multivariate Gaussian, mixture model, NMF, etc.
- Short answer to why question: Extracting structure out of data, understanding data
  - Clustering (mixture models)
  - Document Analysis (LDA)
  - Audio Source Separation (NMF)

What is Generative Modeling, and Why?

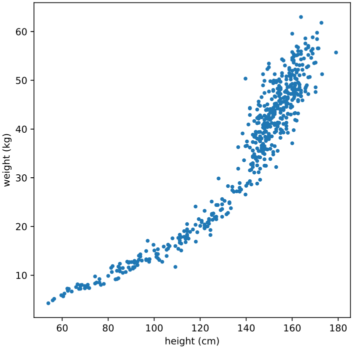
- Learning by assuming a generative process
  - E.g. fitting a multivariate Gaussian, mixture model, NMF, etc.
- Short answer to why question: Extracting structure out of data, understanding data
  - Clustering (mixture models)
  - Document Analysis (LDA)
  - Audio Source Separation (NMF)
  - Image in-painting

What is Generative Modeling, and Why?

- Learning by assuming a generative process
  - E.g. fitting a multivariate Gaussian, mixture model, NMF, etc.
- Short answer to why question: Extracting structure out of data, understanding data
  - Clustering (mixture models)
  - Document Analysis (LDA)
  - Audio Source Separation (NMF)
  - Image in-painting
  - Generating random images (my favorite)

Generative Modeling in Action

Weight and Heights of the members of an African tribe



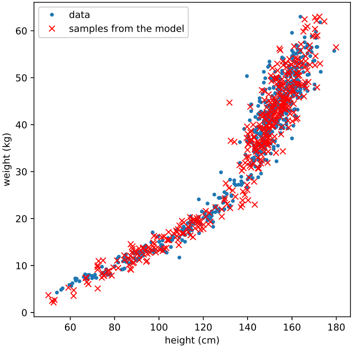
$$
\begin{aligned}
h_n &\sim \text{Discrete}([\pi, 1-\pi]) \\
x_n | h_n &\sim \mathcal{N}(\mu_{h_n}, \Sigma_{h_n})
\end{aligned}
$$

Learning:

$$\max_\theta \sum_n \log p(x_n|\theta)$$

Generative Modeling in Action

Weight and Heights of the members of an African tribe



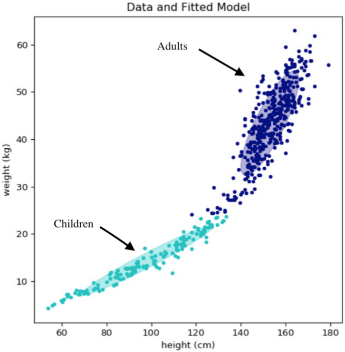
$$
\begin{aligned}
h_n &\sim \text{Discrete}([\pi, 1-\pi]) \\
x_n | h_n &\sim \mathcal{N}(\mu_{h_n}, \Sigma_{h_n})
\end{aligned}
$$

Learning:

$$\max_\theta \sum_n \log p(x_n|\theta)$$

Generative Modeling in Action

Weight and Heights of the members of an African tribe

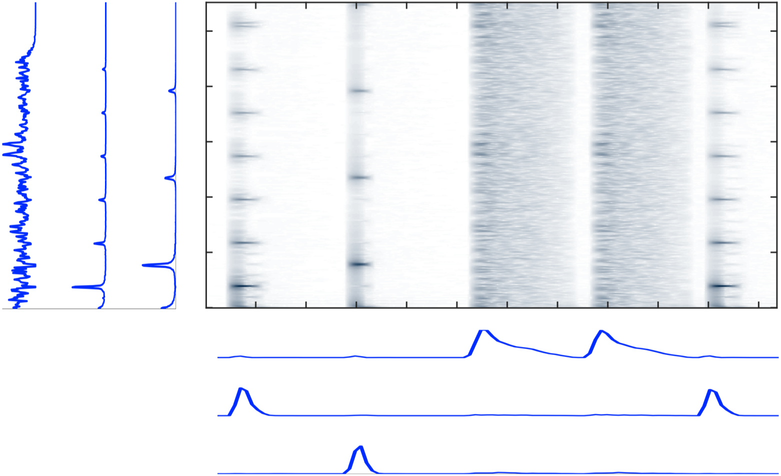


$$
\begin{aligned}
h_n &\sim \text{Discrete}([\pi, 1-\pi]) \\
x_n | h_n &\sim \mathcal{N}(\mu_{h_n}, \Sigma_{h_n})
\end{aligned}
$$

Learning:

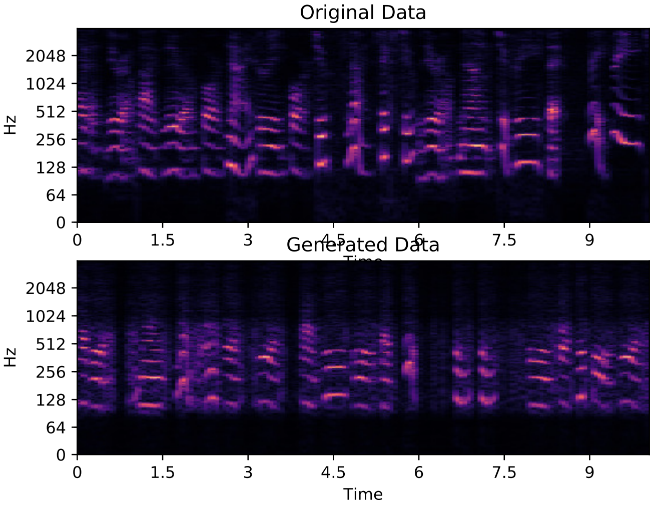
$$
\max_\theta \sum_n \log p(x_n|\theta)
$$

A sequence example



Hugely popular NMF model: $X = WH$
(figure stolen from Paris)

Learning distributions over sequences



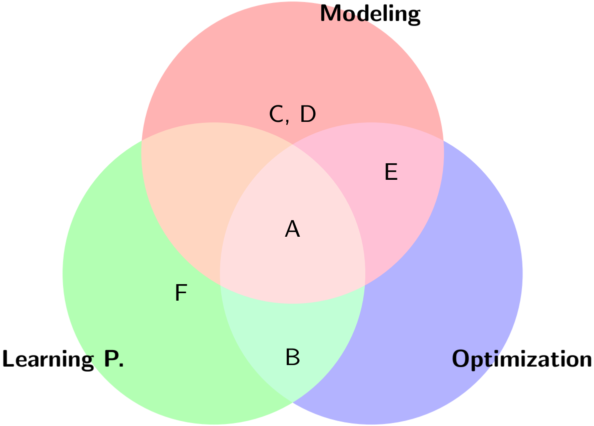
Generated with the method in Chapter 4 of this thesis.

Major issues when generative modeling

- **Modeling/Representation**
  - How we represent the data (what model/distribution we use)
- **Learning Paradigm**
  - The cost function used to measure between model distribution and underlying data distribution (e.g. maximum likelihood, adversarial training, method of moments)
- **Optimization**
  - Given the model and the learning paradigm, the procedure with which we obtain the model parameters.

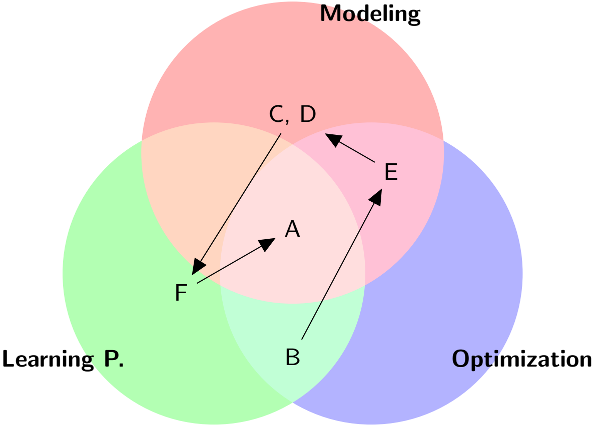
Contributions in this thesis

- **A** - Learning with multi-modal latent representations in implicit generative models (UAI 2018 submission - (New)
- **B** - Method of Moments Framework for HMMs with special structure (NIPS 2014, WASPAA 2015)
- **C** - Convolutional neural nets for source separation (MLSP 2017 best paper award)
- **D** - Diagonal RNNs in symbolic music modeling (WASPAA 2017)
- **E** - Identifiable Factorial HMMs (NIPS 2015, ICASSP 2017 submissions)
- **F** - GANs for source separation (ICASSP 2018) - (New)



Contributions in this thesis

- **A** - Learning with multi-modal latent representations in implicit generative models (UAI 2018 submission - (New)
- **B** - Method of Moments Framework for HMMs with special structure (NIPS 2014, WASPAA 2015)
- **C** - Convolutional neural nets for source separation (MLSP 2017 best paper award)
- **D** - Diagonal RNNs in symbolic music modeling (WASPAA 2017)
- **E** - Identifiable Factorial HMMs (NIPS 2015, ICASSP 2017 submissions)
- **F** - GANs for source separation (ICASSP 2018) - (New)



Plan

Learning in Latent Variable Models

- Typical objective is Maximum Likelihood:

$$
\begin{aligned}
&\max_\theta \mathbb{E}_x \log p(x|\theta) \\
&= \max_\theta \mathbb{E}_x \log \sum_h p(x, h|\theta)
\end{aligned}
$$

- **Observations:** x.
- **Hidden Variables:** h.
- **Parameters (to be optimized)**: θ.

Learning in Latent Variable Models

- Typical objective is Maximum Likelihood:

$$\max_\theta \mathbb{E}_x \log p(x|\theta)$$
$$= \max_\theta \mathbb{E}_x \log \sum_h p(x, h|\theta)$$

- **Observations:** x.
- **Hidden Variables:** h.
- **Parameters (to be optimized)**: θ.
- **In general not convex.**

Learning in Latent Variable Models

- Typical objective is Maximum Likelihood:

$$\max_{\theta} \mathbb{E}_x \log p(x|\theta)$$
$$= \max_{\theta} \mathbb{E}_x \log \sum_h p(x, h|\theta)$$

- **Observations:** x.
- **Hidden Variables:** h.
- **Parameters (to be optimized)**: θ.
- **In general not convex.**
- This poses a challenge in terms of optimization. In general, it is difficult to train latent variable models. Can we devise methods to more easily reach solutions around the global optimum?

Method of Moments

- The idea is to estimate the models parameters θ by solving a system of non-linear equations formed with moments $\mathbb{E}[g_k(x)]$, $k \in \{1, \dots K\}$:

$$\begin{aligned}\mathbb{E}[g_1(x)] &= f_1(\theta) \\ &\vdots \\ \mathbb{E}[g_K(x)] &= f_K(\theta)\end{aligned}$$

Method of Moments

- The idea is to estimate the models parameters θ by solving a system of non-linear equations formed with moments $\mathbb{E}[g_k(x)]$, $k \in \{1, \dots K\}$:

$$\begin{aligned} \mathbb{E}[g_1(x)] &= f_1(\theta) \\ &\vdots \\ \mathbb{E}[g_K(x)] &= f_K(\theta) \end{aligned}$$

- Canonical Example: $x \sim \mathcal{G}(a, b)$:

$$\begin{aligned} \mathbb{E}[x] &= ab \\ \mathbb{E}[x^2] &= ab^2 + a^2b^2 \end{aligned} \quad \rightarrow \quad \begin{aligned} \widehat{b} &= (\mathbb{E}[x^2] - \mathbb{E}[x]^2)/\mathbb{E}[x] \\ \widehat{a} &= \mathbb{E}[x]^2/(\mathbb{E}[x^2] - \mathbb{E}[x]^2) \end{aligned}$$

Method of Moments

- The idea is to estimate the models parameters θ by solving a system of non-linear equations formed with moments $\mathbb{E}[g_k(x)]$, $k \in \{1, \dots K\}$:

$$\begin{aligned}\mathbb{E}[g_1(x)] =& f_1(\theta)\\ &\vdots\\ \mathbb{E}[g_K(x)] =& f_K(\theta)\end{aligned}$$

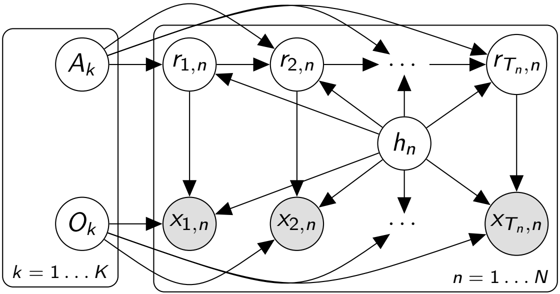
- Canonical Example: $x \sim \mathcal{G}(a, b)$:

$$\begin{aligned}\mathbb{E}[x] =& ab\\ \mathbb{E}[x^2] =& ab^2 + a^2 b^2\end{aligned} \quad \rightarrow \quad \begin{aligned}\widehat{b} =& (\mathbb{E}[x^2] - \mathbb{E}[x]^2)/\mathbb{E}[x]\\ \widehat{a} =& \mathbb{E}[x]^2/(\mathbb{E}[x^2] - \mathbb{E}[x]^2)\end{aligned}$$

- Can we do this for latent variable models?

Spectral Learning of Mixture of HMMs

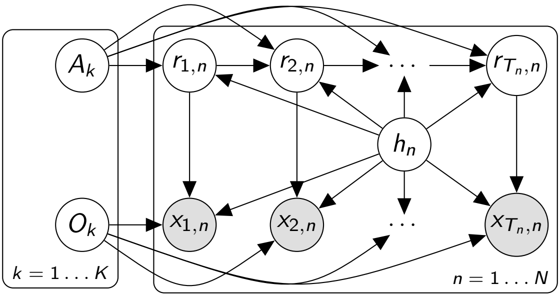
[MHMM, Smyth 97]



$$
\begin{aligned}
h_n &\sim Categorical(\pi_n) \\
\mathbf{x}_n &\sim HMM(A_n, O_n)
\end{aligned}
$$

Spectral Learning of Mixture of HMMs

[MHMM, Smyth 97]



$$
\begin{aligned}
h_n &\sim Categorical(\pi_n) \\
\mathbf{x}_n &\sim HMM(A_n, O_n)
\end{aligned}
$$

- Learning Goal: Estimate π_n, A_n, O_n, given $\mathbf{x}_{1:N}$

Global view for Mixture of HMMs

- An MHMM with *local* parameters $\theta_{1:K} = (O_{1:K}, A_{1:K}, \nu_{1:K}, \pi)$ is an HMM with *global* parameters $\bar{\theta} = (\bar{O}, \bar{A}, \bar{\nu})$, where:

$$\bar{O} = \begin{bmatrix} O_1 & \dots & O_K \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} A_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & A_2 & \dots & \mathbf{0} \\ & & \ddots & \\ \mathbf{0} & \mathbf{0} & \dots & A_K \end{bmatrix}, \quad \bar{\nu} = \begin{bmatrix} \pi_1 \nu_1 \\ \pi_2 \nu_2 \\ \vdots \\ \pi_K \nu_K \end{bmatrix} .$$

Global view for Mixture of HMMs

- An MHMM with *local* parameters $\theta_{1:K} = (O_{1:K}, A_{1:K}, \nu_{1:K}, \pi)$ is an HMM with *global* parameters $\bar{\theta} = (\bar{O}, \bar{A}, \bar{\nu})$, where:

$$\bar{O} = \begin{bmatrix} O_1 & \dots & O_K \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} A_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & A_2 & \dots & \mathbf{0} \\ & & \ddots & \\ \mathbf{0} & \mathbf{0} & \dots & A_K \end{bmatrix}, \quad \bar{\nu} = \begin{bmatrix} \pi_1 \nu_1 \\ \pi_2 \nu_2 \\ \vdots \\ \pi_K \nu_K \end{bmatrix} .$$

- Estimating the global parameters $\bar{\theta}$ with a moment algorithm would introduce permutation $\mathcal{P}$ and noise to the estimates.
- How to impose this structural constraint on the estimator?

Two stage estimation for HMMs

HMM-Mixture model equivalence, [Kontorovich et al., 13]

An HMM with state marginals $p(h_t)$ is equivalent to a mixture model with mixing weights $\pi := \frac{1}{T} \sum_{t=1}^{T} p(h_t)$, and the same emission parameters.

Two stage estimation for HMMs

HMM-Mixture model equivalence, [Kontorovich et al., 13]

An HMM with state marginals $p(h_t)$ is equivalent to a mixture model with mixing weights $\pi := \frac{1}{T}\sum_{t=1}^{T} p(h_t)$, and the same emission parameters.

- First compute (estimate) $\widehat{O}$, and $\widehat{\pi}$.
- Then solve the convex problem:

$$
\begin{aligned}
\widehat{A} = \arg\min_{A} \|M_2 - \widehat{O} A \mathrm{diag}(\widehat{\pi}) \widehat{O}\|_F \\
s.t.\ 1^\top A = 1^\top, \\
A \geq 0. \\
(1 - \mathcal{M}) \odot A = 0,
\end{aligned}
$$

where $\mathcal{M}$ encodes the block diagonal structure.

Two stage estimation for HMMs

HMM-Mixture model equivalence, [Kontorovich et al., 13]

An HMM with state marginals $p(h_t)$ is equivalent to a mixture model with mixing weights $\pi := \frac{1}{T}\sum_{t=1}^{T} p(h_t)$, and the same emission parameters.

- First compute (estimate) $\widehat{O}$, and $\widehat{\pi}$.
- Then solve the convex problem:

$$\begin{aligned} \widehat{A} = \arg\min_{A} \|M_2 - \widehat{O}A\mathrm{diag}(\widehat{\pi})\widehat{O}\|_F \\ s.t.\ 1^\top A = 1^\top, \\ A \geq 0. \\ \cancel{(1-\mathcal{M}) \odot A = 0,} \end{aligned}$$

  where $\mathcal{M}$ encodes the block diagonal structure.
- **Problem:** $\widehat{O}$ is still permuted.

Two stage estimation for HMMs

HMM-Mixture model equivalence, [Kontorovich et al., 13]

An HMM with state marginals $p(h_t)$ is equivalent to a mixture model with mixing weights $\pi := \frac{1}{T}\sum_{t=1}^{T} p(h_t)$, and the same emission parameters.

- First compute (estimate) $\widehat{O}$, and $\widehat{\pi}$.
- Then solve the convex problem:

$$\begin{aligned}\widehat{A} = \arg\min_A \; & \|M_2 - \widehat{O}A\mathrm{diag}(\widehat{\pi})\widehat{O}\|_F \\ s.t.\; & 1^\top A = 1^\top, \\ & A \geq 0. \\ & \cancel{(1-\mathcal{M})\odot A = 0,}\end{aligned}$$

  where $\mathcal{M}$ encodes the block diagonal structure.
- **Problem:** $\widehat{O}$ is still permuted.
- But $\widehat{A}$ is de-permutable! (if we remove the block diagonal constraint)

Two stage estimation framework with structural constraints:

Two stage estimation framework

- Get rough/permuted estimates for the parameters $\widehat{O}, \widehat{A}, \widehat{\pi}$.
- De-permute A. (Solve the graph problem dictated by model)
- Solve:

$$
\begin{aligned}
&\min_{A} \|M_2 - \widehat{O} A \mathrm{diag}(\widehat{\pi}) \widehat{O}\|_F \\
&s.t.\ 1^\top A = 1^\top, \\
&\qquad A \geq 0. \\
&\qquad f(\mathcal{M}, A) = 0
\end{aligned}
$$

- f, and $\mathcal{M}$ depend on the model.

Two stage estimation framework with structural constraints:

Two stage estimation framework

- Get rough/permuted estimates for the parameters $\widehat{O}, \widehat{A}, \widehat{\pi}$.
- De-permute A. (Solve the graph problem dictated by model)
- Solve:

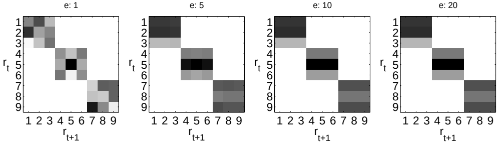
$$\begin{aligned} &\min_A \|M_2 - \widehat{O} A \mathrm{diag}(\widehat{\pi}) \widehat{O}\|_F \\ &s.t.\ 1^\top A = 1^\top, \\ &\quad\ \ A \geq 0. \\ &\quad\ \ f(\mathcal{M}, A) = 0 \end{aligned}$$

- f, and $\mathcal{M}$ depend on the model.

For MHMM $\mathcal{M}$ is the complement of a binary block diagonal matrix.

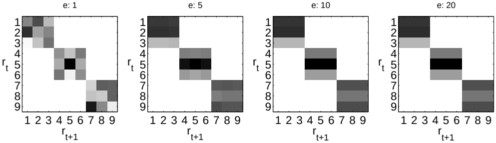
Mixture of HMMs: De-permutation

- $\lim_{e\to\infty} \bar{A}^e = [\bar{v}_1 1_M^\top,\ \bar{v}_2 1_M^\top,\ \ldots,\ \bar{v}_K 1_M^\top]$, where $\bar{v}_k$ is the k'th eigenvector of $\bar{A}$.

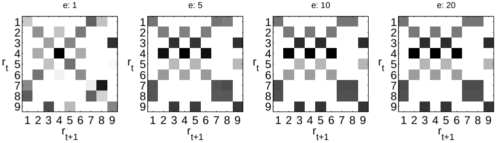


Mixture of HMMs: De-permutation

- $\lim_{e \to \infty} \bar{A}^e = [\bar{v}_1 1_M^\top,\ \bar{v}_2 1_M^\top,\ \ldots,\ \bar{v}_K 1_M^\top]$, where $\bar{v}_k$ is the k'th eigenvector of $\bar{A}$.

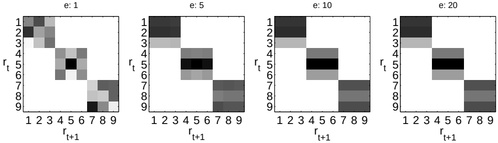


- What happens for $\mathcal{P}(\bar{A})$:

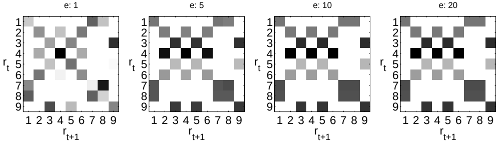


Mixture of HMMs: De-permutation

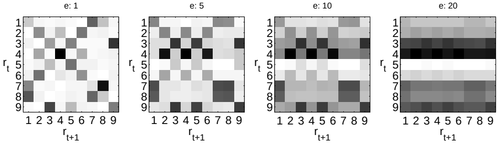
- $\lim_{e \to \infty} \bar{A}^e = [\bar{v}_1 1_M^\top,\ \bar{v}_2 1_M^\top,\ \ldots,\ \bar{v}_K 1_M^\top]$, where $\bar{v}_k$ is the k'th eigenvector of $\bar{A}$.



- What happens for $\mathcal{P}(\bar{A})$:

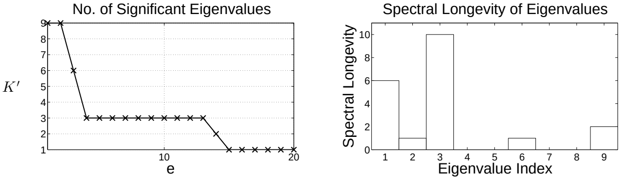


- What happens in practice:



MHMM De-permutation Continued

- But we can estimate the number of HMMs:



- Then form rank-$\widehat{K}$ reconstruction A^r:

$$A^r = V_{1:\widehat{K}} \Lambda_{1:\widehat{K}} V^{-1}$$

- Then Cluster. (A La Spectral Clustering)

Experimental Results on Clustering Handwritten Digits

- **Experiment:** Clustering handwritten digit trajectories by learning MHMMs.
- We form datasets composed of digits 1-2, 1-3, 2-3, and so on.

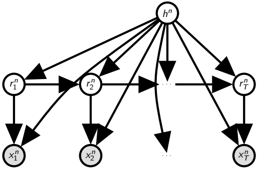
Algorithm	**1v2**	**1v3**	**1v4**	**1v5**	**2v3**	**2v4**	**2v5**
Spectral	100	70	54	55	83	99	99
EM init. at Random	96	99	98	54	83	100	100
EM init. w/ Spectral	**100**	**99**	**100**	**100**	**96**	**100**	**100**

Numbers show percent clustering accuracies.

- Initializing EM with the spectral algorithm boosts the results.

Generalization

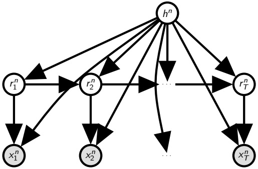
Mixture of HMMs



$$\begin{bmatrix} A_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & A_2 & \dots & \mathbf{0} \\ & & \ddots & \\ \mathbf{0} & \mathbf{0} & \dots & A_K \end{bmatrix}$$

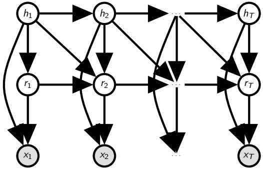
Generalization

Mixture of HMMs



$$\begin{bmatrix} A_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & A_2 & \cdots & \mathbf{0} \\ & & \ddots & \\ \mathbf{0} & \mathbf{0} & \cdots & A_K \end{bmatrix}$$

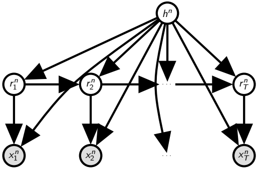
Switching HMM



$$\begin{bmatrix} B_{1,1}\ A_{1,1} & B_{1,2}\ \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^\top & \cdots & B_{1,K}\ \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^\top \\ B_{2,1}\ \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^\top & B_{2,2}\ A_{2,2} & \cdots & B_{2,K}\ \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^\top \\ & & \ddots & \\ B_{K,1}\ \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^\top & B_{K,2}\ \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^\top & \cdots & B_{K,K}\ A_{K,K} \end{bmatrix}$$

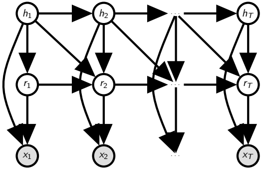
Generalization

Mixture of HMMs



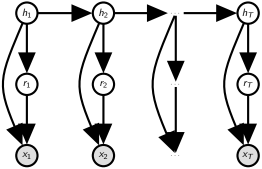
$$\begin{bmatrix} A_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & A_2 & \cdots & \mathbf{0} \\ & & \ddots & \\ \mathbf{0} & \mathbf{0} & \cdots & A_K \end{bmatrix}$$

Switching HMM



$$\begin{bmatrix} B_{1,1}\ A_{1,1} & B_{1,2}\ \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^\top & \cdots & B_{1,K}\ \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^\top \\ B_{2,1}\ \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^\top & B_{2,2}\ A_{2,2} & \cdots & B_{2,K}\ \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^\top \\ & & \ddots & \\ B_{K,1}\ \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^\top & B_{K,2}\ \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^\top & \cdots & B_{K,K}\ A_{K,K} \end{bmatrix}$$

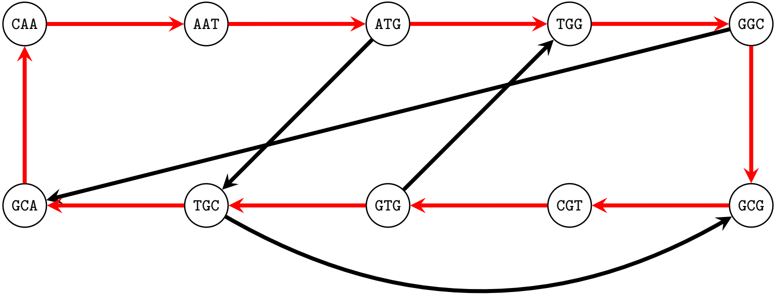
Hidden Markov Model
with Mixture Observations



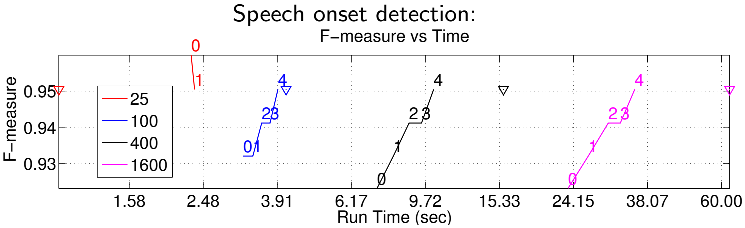
$$\begin{bmatrix} B_{1,1}\ \nu_1\mathbf{1}_M^\top & B_{1,2}\ \nu_1\mathbf{1}_M^\top & \cdots & B_{1,K}\ \nu_1\mathbf{1}_M^\top \\ B_{2,1}\ \nu_2\mathbf{1}_M^\top & B_{2,2}\ \nu_2\mathbf{1}_M^\top & \cdots & B_{2,K}\ \nu_2\mathbf{1}_M^\top \\ & & \ddots & \\ B_{K,1}\ \nu_K\mathbf{1}_M^\top & B_{K,2}\ \nu_K\mathbf{1}_M^\top & \cdots & B_{K,K}\ \nu_K\mathbf{1}_M^\top \end{bmatrix}$$

Bakis-HMM

- Is an HMM that can only move one state at a time.
- $A = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 1 & 1 & \dots & 0 & 0 \\ 0 & \dots & \ddots & \dots & 0 \\ 0 & \dots & 1 & 1 & 0 \\ 0 & \dots & 0 & 1 & 1 \end{bmatrix}$
- Every state is visited exactly once.
- **Depermutation:** Find a maximum weight Hamiltonian circuit on $\widehat{A}$. (Traveling Salesman problem)



Experimental Results on Speech Onset Detection



- Triangles denote randomly initialized EM performance on run-time vs f-measure. (EM is implemented in C)
- Numbers show spectral + number of EM iterations.
- Spectral Algorithm accelerates EM learning.

Contribution and summary

- **Contribution**: A method of moments based framework for HMMs with special transition structure. (learning paradigm)
  - Helps in initializing EM. (optimization)

Contribution and summary

- **Contribution**: A method of moments based framework for HMMs with special transition structure. (learning paradigm)
  - Helps in initializing EM. (optimization)

Thoughts on method of moments:

- Good:
  - **Global**
  - **Initialization:** No need to worry about initialization (**Great initializer for EM (optimization)**).
  - **Scalable:** Computationally cheap: Gather the moments, factorize a small matrix.
  - **Interesting/Theoretical:** Bounds.

Contribution and summary

- **Contribution**: A method of moments based framework for HMMs with special transition structure. (learning paradigm)
  - Helps in initializing EM. (optimization)

Thoughts on method of moments:

- Good:
  - **Global**
  - **Initialization:** No need to worry about initialization (**Great initializer for EM (optimization)**).
  - **Scalable:** Computationally cheap: Gather the moments, factorize a small matrix.
  - **Interesting/Theoretical:** Bounds.
- Bad:
  - **Model Mismatch:** Horrible in regards to model mismatch. (Hard assumption on model Unlike ML, which minimizes $KL(p||q)$.

Contribution and summary

- **Contribution**: A method of moments based framework for HMMs with special transition structure. (learning paradigm)
    - Helps in initializing EM. (optimization)

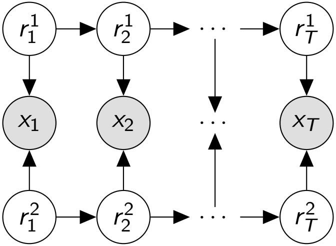
Thoughts on method of moments:

- Good:
    - **Global**
    - **Initialization:** No need to worry about initialization (**Great initializer for EM (optimization)**).
    - **Scalable:** Computationally cheap: Gather the moments, factorize a small matrix.
    - **Interesting/Theoretical:** Bounds.
- Bad:
    - **Model Mismatch:** Horrible in regards to model mismatch. (Hard assumption on model Unlike ML, which minimizes $KL(p||q)$.
- Ugly:
    - You can get complex numbers for parameter estimates/likelihoods.

Plan

Factorial HMM

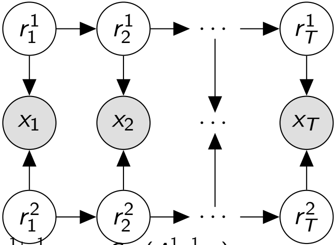
[Ghahramani, Jordan; 97]



$$
\begin{aligned}
r_t^1 | r_{t-1}^1 &\sim Cat(A^1 r_{t-1}^1) \\
&\vdots \\
r_t^K | r_{t-1}^K &\sim Cat(A^K r_{t-1}^K) \\
x_t | r_t^1, \ldots, r_t^K &\sim \mathcal{N}([O^1, \ldots, O^K] \begin{bmatrix} r_t^1 \\ \cdots \\ r_t^K \end{bmatrix}, \sigma^2 I)
\end{aligned}
$$

Factorial HMM

[Ghahramani, Jordan; 97]



$$
\begin{aligned}
r_t^1 | r_{t-1}^1 &\sim Cat(A^1 r_{t-1}^1) \\
&\vdots \\
r_t^K | r_{t-1}^K &\sim Cat(A^K r_{t-1}^K) \\
x_t | r_t^1, \ldots, r_t^K &\sim \mathcal{N}([O^1, \ldots, O^K] \begin{bmatrix} r_t^1 \\ \ldots \\ r_t^K \end{bmatrix}, \sigma^2 I)
\end{aligned}
$$

$$
X = \underbrace{O}_{\textit{The dictionary}} \underbrace{R}_{\textit{Activations}} + \underbrace{\epsilon}_{\textit{noise}}
$$

Some Dictionary Learning Perspective..

- General Dictionary Learning

$$\min_{O,R} \|X - \underbrace{O}_{Dictionary} \underbrace{R}_{Activations} \|_F$$

  - **PCA:** Both O and R are orthogonal.
  - **ICA:** Solvable if R has independent coordinates.
  - **Mixture Model:** R is one sparse. Solvable is O has full column rank.
  - **Sparse Dictionary Learning:** Solvable if O is square and R is sparse Bernouilli-Gaussian. [Spielman et al. 12]

Some Dictionary Learning Perspective..

- General Dictionary Learning

$$\min_{O,R} \|X - \underbrace{O}_{Dictionary} \underbrace{R}_{Activations} \|_F$$

  - **PCA:** Both O and R are orthogonal.
  - **ICA:** Solvable if R has independent coordinates.
  - **Mixture Model:** R is one sparse. Solvable is O has full column rank.
  - **Sparse Dictionary Learning:** Solvable if O is square and R is sparse Bernouilli-Gaussian. [Spielman et al. 12]

- Factorial Models:

$$O = \begin{bmatrix} O^1 & \dots & O^K \end{bmatrix}, \ R = \begin{bmatrix} R^1 \\ \vdots \\ R^K \end{bmatrix}$$

  - No constraint on O, columns of R are block-K sparse.
  - No Unique Solution!!!

FHMM Identifiability

Rank Deficiency

$\mathbf{rank}(R) \leq MK - (K - 1)$

FHMM Identifiability

Rank Deficiency

$\mathbf{rank}(R) \leq MK - (K-1)$

Proof Sketch:

$$\dim(\text{null}(R^\top)) \geq K - 1.$$

Therefore from rank-nullity theorem $\mathbf{rank}(R) \leq MK - (K-1)$.

FHMM is unidentifiable

For a given assignment matrix $R \in \mathbb{R}^{KM \times T}$ There exists $O_1 \neq O_2$ such that $\prod_t \mathcal{N}(x_t | O_1 R, \sigma^2 I) = \prod_t \mathcal{N}(x_t | O_2 R, \sigma^2 I)$.

FHMM Identifiability

Rank Deficiency

$\textbf{rank}(R) \leq MK - (K-1)$

Proof Sketch:

$$\dim(\text{null}(R^\top)) \geq K - 1.$$

Therefore from rank-nullity theorem $\textbf{rank}(R) \leq MK - (K-1)$.

FHMM is unidentifiable

For a given assignment matrix $R \in \mathbb{R}^{KM \times T}$ There exists $O_1 \neq O_2$ such that $\prod_t \mathcal{N}(x_t | O_1 R, \sigma^2 I) = \prod_t \mathcal{N}(x_t | O_2 R, \sigma^2 I)$.

Proof: Since $\dim(\text{null}(R^\top)) \geq K - 1$, $(O_1 - O_2)R = 0$, for $O_1 \neq O_2$.

FHMM Identifiable Alternative 1

Shared Component FM

$$\forall k,\ O^k = \begin{bmatrix} | & | & | & | & | \\ \mu_k^1 & \mu_2^k & \dots & \mu_{M-1}^k & s \\ | & | & | & | & | \end{bmatrix}$$

SC-FM is identifiable

Given an assignment matrix $\widetilde{R}$ which is rank $MK - (K-1)$, the emission matrix of an SC-FM is identifiable.

Proof Sketch:

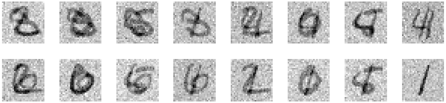
$$\dim(\text{null}(R^\top)) = 0.$$

Therefore $(O_1 - O_2)R \neq 0$, $\forall\ O_1 \neq O_2$.

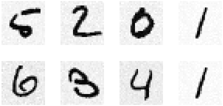
Learning Example for Shared Component Factorial Model

- **Gist:** If the shared component s is incoherent, then we can identify it, and reveal the other components.

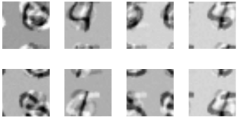
Example Observations



Obtained Components with SC-FM



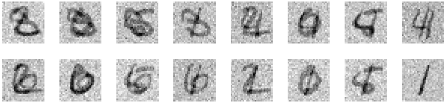
Components with regular model-EM



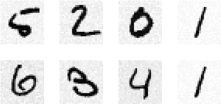
Learning Example for Shared Component Factorial Model

- **Gist:** If the shared component s is incoherent, then we can identify it, and reveal the other components.

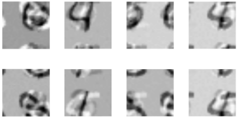
Example Observations



Obtained Components with SC-FM



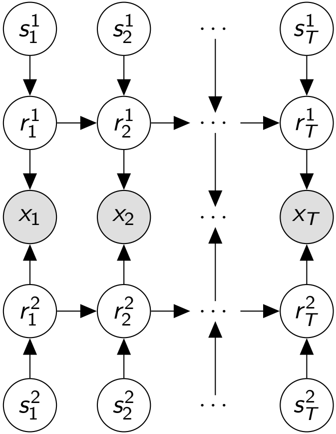
Components with regular model-EM



- The shared component + incoherence assumption a bit too restrictive. Can we think of another model?

FHMM Identifiable Alternative 2

$$
\begin{aligned}
{\color{red} s_t^k} &\;{\color{red} \sim Bernoulli(\pi),\;\; k \in \{1, \ldots, K\}} \\
r_t^1 | r_{t-1}^1 &\sim s_t^1\, Cat(A^1 r_{t-1}^1) \\
&\vdots \\
r_t^K | r_{t-1}^K &\sim s_t^K\, Cat(A^K r_{t-1}^K) \\
x_t | r_t^1, \ldots, r_t^K &\sim \mathcal{N}([O^1, \ldots, O^K] \begin{bmatrix} r_t^1 \\ \ldots \\ r_t^K \end{bmatrix}, \sigma^2 I)
\end{aligned}
$$



- Identifiability follows similarly from the activation matrix R.

Revealing FHMM Practical Algorithm

Practical Algorithm for Revealing FHMM

- Cluster the data matrix $X \in \mathbb{R}^{L \times T}$ into clusters $X^c \in \mathbb{R}^{L \times C}$.
- Solve:

$$\begin{aligned} &\min_H \|X^c - X^c H\|_F^2 + \beta \|H\|_1, \\ &s.t.\ H_{i,i} = 0,\ \text{for } 1 \leq i \leq C, \\ &H \geq 0, \end{aligned}$$

  where $H \in \mathbb{R}^{C \times C}$.
- Construct a bi-partite graph by reading the solution for H.

Revealing FHMM Practical Algorithm

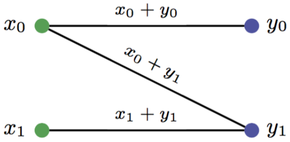
Practical Algorithm for Revealing FHMM

- Cluster the data matrix $X \in \mathbb{R}^{L \times T}$ into clusters $X^c \in \mathbb{R}^{L \times C}$.
- Solve:

$$\begin{aligned} &\min_H \|X^c - X^c H\|_F^2 + \beta \|H\|_1, \\ &s.t.\ H_{i,i} = 0,\ \text{for } 1 \le i \le C, \\ &\quad H \ge 0, \end{aligned}$$

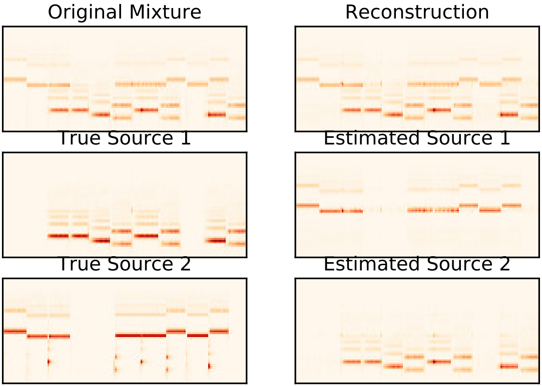
  where $H \in \mathbb{R}^{C \times C}$.
- Construct a bi-partite graph by reading the solution for H.

- **Condition for learnability:** Let $O_1 = [x_0, x_1]$, $O_2 = [y_0, y_1]$. Observed combinations needs to form a connected graph (Connectivity), and we need to observe all nodes and edges (Observability).



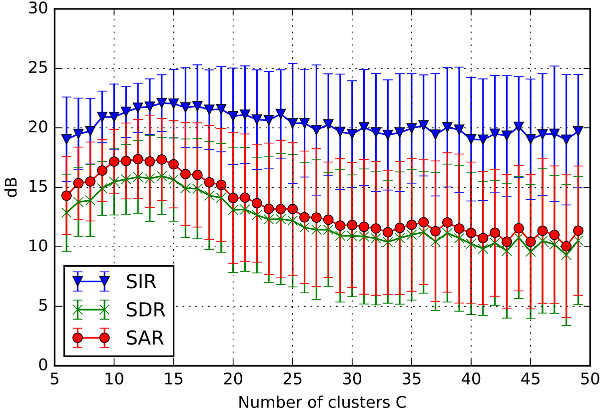
Unsupervised audio source separation example

- We mixed recording of double bass and flute (at 0dB).
- The observed mixtures satisfy the connectivity constraint.



- We obtain almost perfect source separation.

Sensitivity on number of clusters



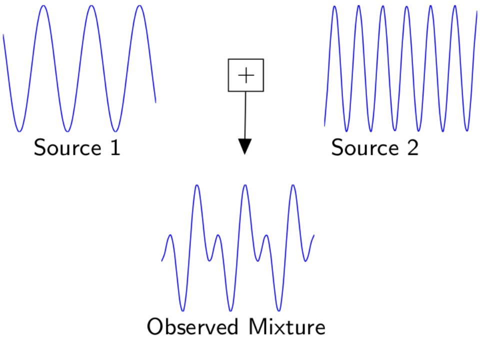
- The algorithm is robust to the choice of number of clusters C.

Contributions and thoughts

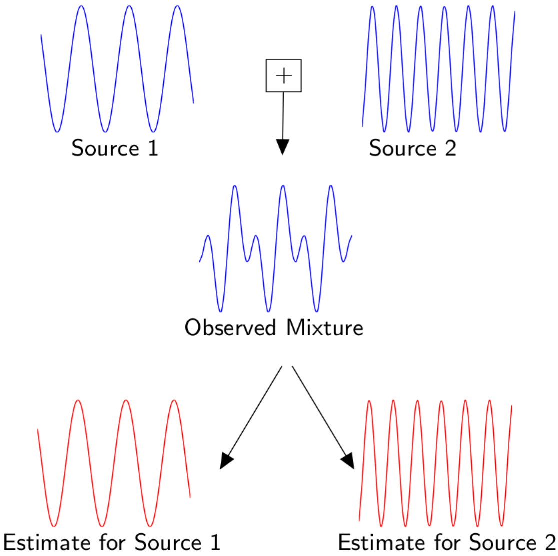
- **Contribution 1:** We have shown that the standard Factorial Model is not statistically identifiable. (modeling)
- **Contribution 2:** We have proposed two identifiable alternatives, along with practical parameter estimation algorithms. (modeling and optimization)
- **Future work:**
  - Can we relax the observability assumption so that we only require to observe less nodes in the connectivity graph?
  - Potential application in semi-supervised source separation.

Plan

Source Separation



Source Separation



Generative Supervised Source Separation

- Assumes the following generative model:

$$\begin{aligned} s_1 &\sim p_{\text{out}}(s_1 | f_{\theta^1}(h_1)) \\ s_2 &\sim p_{\text{out}}(s_2 | f^2_{\theta^2}(h_2)) \\ x &\sim p_{\text{out}}(x | s_1 + s_2) \end{aligned}$$

Generative Supervised Source Separation

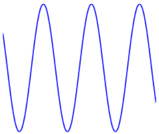
- Assumes the following generative model:

$$\begin{aligned} s_1 &\sim p_{\text{out}}(s_1 | f_{\theta^1}(h_1)) \\ s_2 &\sim p_{\text{out}}(s_2 | f^2_{\theta^2}(h_2)) \\ x &\sim p_{\text{out}}(x | s_1 + s_2) \end{aligned}$$

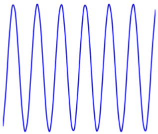
- First train the generative models for each source (with Maximum Likelihood):

$$\max_{\theta^k} \mathbb{E}_{s_k} p(s_k | f_{\theta^k}(h_k)),$$

where $h_k = f^{\text{enc}}_{\theta^k}(s_k)$, is some encoding.



Train $f_{\theta^1}(h_1)$

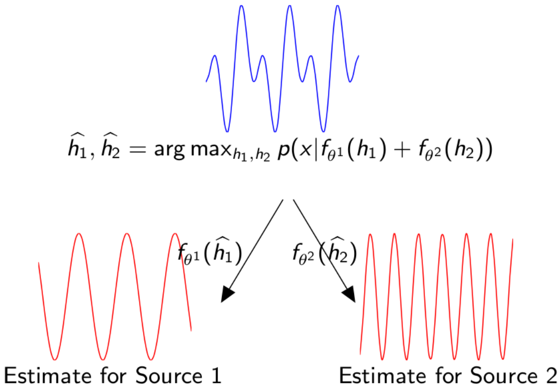


Train $f_{\theta^2}(h_2)$

Generative Supervised Source Separation

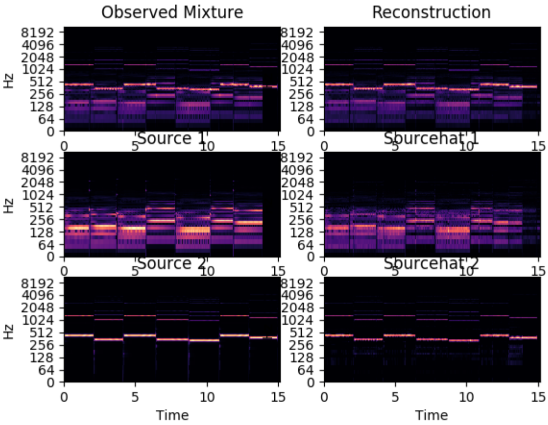
In test time, the source estimates are obtained via:

$$\widehat{h_1}, \widehat{h_2} = \arg\max_{h_1, h_2} p(x | f_{\theta^1}(h_1) + f_{\theta^2}(h_2))$$



Separation Example

- Separation is usually done on spectrograms.



- Because of non-negativity, we usually use $p(X|f_\theta(H)) = \mathcal{PO}(X; f_\theta(H))$

Popular Linear Models for Supervised Source Separation

- Non-Negative Matrix Factorization (NMF) [Smaragdis 2003]

$$f_\theta(H) = WH,\ W \geq 0, H \geq 0$$

Only, the forward model $f_\theta(H)$ is specified, H is obtained with an algorithm.

Popular Linear Models for Supervised Source Separation

- Non-Negative Matrix Factorization (NMF) [Smaragdis 2003]

$$f_\theta(H) = WH,\ W \geq 0, H \geq 0$$

  Only, the forward model $f_\theta(H)$ is specified, H is obtained with an algorithm.

- Convolutive NMF [Smaragdis 2004]

$$f_\theta(H) = \sum_{k=0}^{K} W_k * H_k,\ W \geq 0, H \geq 0$$

Popular Linear Models for Supervised Source Separation

- Non-Negative Matrix Factorization (NMF) [Smaragdis 2003]

$$f_\theta(H) = WH, \; W \geq 0, H \geq 0$$

  Only, the forward model $f_\theta(H)$ is specified, H is obtained with an algorithm.

- Convolutive NMF [Smaragdis 2004]

$$f_\theta(H) = \sum_{k=0}^{K} W_k * H_k, \; W \geq 0, H \geq 0$$

- Linear mappings allow adaptive step-size optimization algorithms such as EM, multiplicative update rules (even globally optimal methods such as method of moments). However representation wise, they are limited.

Popular Linear Models for Supervised Source Separation

- Non-Negative Matrix Factorization (NMF) [Smaragdis 2003]

$$f_\theta(H) = WH, \;\; W \geq 0, H \geq 0$$

  Only, the forward model $f_\theta(H)$ is specified, H is obtained with an algorithm.

- Convolutive NMF [Smaragdis 2004]

$$f_\theta(H) = \sum_{k=0}^{K} W_k * H_k, \;\; W \geq 0, H \geq 0$$

- Linear mappings allow adaptive step-size optimization algorithms such as EM, multiplicative update rules (even globally optimal methods such as method of moments). However representation wise, they are limited.
- Rest of the thesis will utilize generative models which employ more general non-linear mappings. (neural networks)

More general mappings

- Neural Network Alternative for NMF [Smaragdis, Venkataramani, 2016]

$$\begin{aligned} f_\theta(X) =& \sigma(WH(X)) \\ =& \sigma(Wf_\theta^{\text{enc}}(X)) \\ =& \sigma(W\sigma(W^{\text{enc}}X)) \end{aligned}$$

where $f_\theta^{\text{enc}}(X)$ is the encoder, and it is learned.

More general mappings

- Neural Network Alternative for NMF [Smaragdis, Venkataramani, 2016]

$$
\begin{aligned}
f_\theta(X) =& \sigma(WH(X)) \\
=& \sigma(W f_\theta^{\text{enc}}(X)) \\
=& \sigma(W\sigma(W^{\text{enc}} X))
\end{aligned}
$$

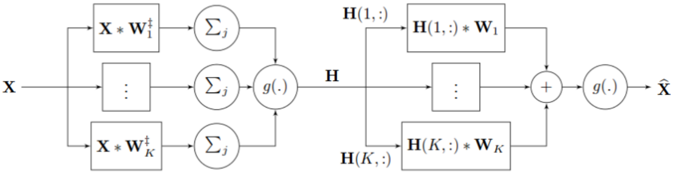
  where $f_\theta^{\text{enc}}(X)$ is the encoder, and it is learned.
- Convolutive neural-net alternative?

Convolutive Neural Network Alternative

- Neural Network Alternatives for Convolutive NMF [Best student paper award, MLSP 2017]

$$f_\theta(H(X)) = \sigma\left(\sum_{k=0}^{K} W_k * H_k(X)\right),$$

where $H_k(X) = \sigma\left(\sum_j (W_k^{inv} * X)_j\right)$

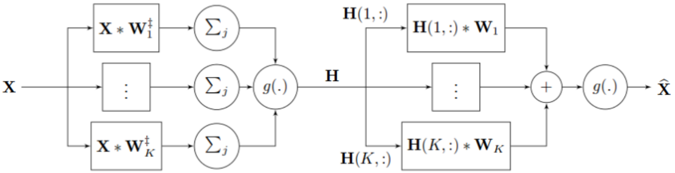


Convolutive Neural Network Alternative

- Neural Network Alternatives for Convolutive NMF [Best student paper award, MLSP 2017]

$$f_\theta(H(X)) = \sigma\left(\sum_{k=0}^{K} W_k * H_k(X)\right),$$

where $H_k(X) = \sigma\left(\sum_j (W_k^{inv} * X)_j\right)$

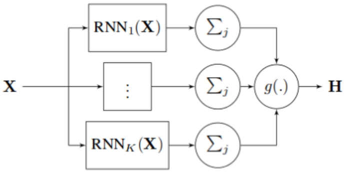


- We can also try RNNs to model arbitrarily long dependencies.

Using RNNs in the Encoder

- RNNs in the encoder:

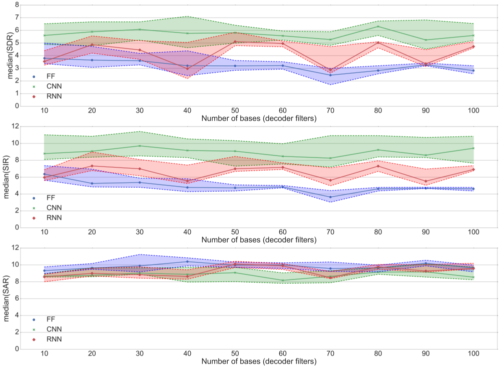
$$H_k(X) = \sigma\left(\sum_j (RNN_k(X))_j\right)$$



Experimental Set-up

- **Dataset:** Male-female speaker mixtures from TIMIT dataset.
  - Training set: 9 utterances for each speaker.
  - Test set: Single sentence mixture at 0dB.
  - Evaluated for 25 pairs of speakers.
- **Evaluation:** BSS eval metrics. (SIR, SAR, SDR)
- We compare Feedforward-Feedforward, Convolutive-Convolutive, Recurrent-Convolutive Autoencoders.

Some results



- Conv-Conv, Conv-RNN, FF-FF autoencoders.
- Variance is over the speaker pairs.
- Significant SIR improvement with Convolutive Models.
- Recursive encoder model is better than the baseline, but not as good as the convolutive model.

- Maximum likelihood training requires specifying $p_{\text{out}}(.)$/loss function.
- Instead of hand picking $p_{\text{out}}(.)$, we can use adversarial training.

- Maximum likelihood training requires specifying $p_{\text{out}}(.)$/loss function.
- Instead of hand picking $p_{\text{out}}(.)$, we can use adversarial training.

ML objective

$$\max_{\theta_k} \mathbb{E}_{s_k} p_{\text{out}}(s_k | f_{\theta^k}(h_k)),$$

Generative Adversarial Source Separation

- Maximum likelihood training requires specifying $p_{\text{out}}(.)$/loss function.
- Instead of hand picking $p_{\text{out}}(.)$, we can use adversarial training.

ML objective

$$\max_{\theta_k} \mathbb{E}_{s_k} p_{\text{out}}(s_k | f_{\theta^k}(h_k)),$$

Adversarial training objective

$$\min_{\xi} \max_{\theta_k} \mathbb{E}_{s_k} \log D_{\xi_k}(s_k) + \mathbb{E}_{h_k} \log(1 - D_{\xi_k}(f_{\theta^k}(h_k)))$$

Generative Adversarial Source Separation

- Maximum likelihood training requires specifying $p_{\text{out}}(.)$/loss function.
- Instead of hand picking $p_{\text{out}}(.)$, we can use adversarial training.

ML objective

$$\max_{\theta_k} \mathbb{E}_{s_k} p_{\text{out}}(s_k | f_{\theta^k}(h_k)),$$

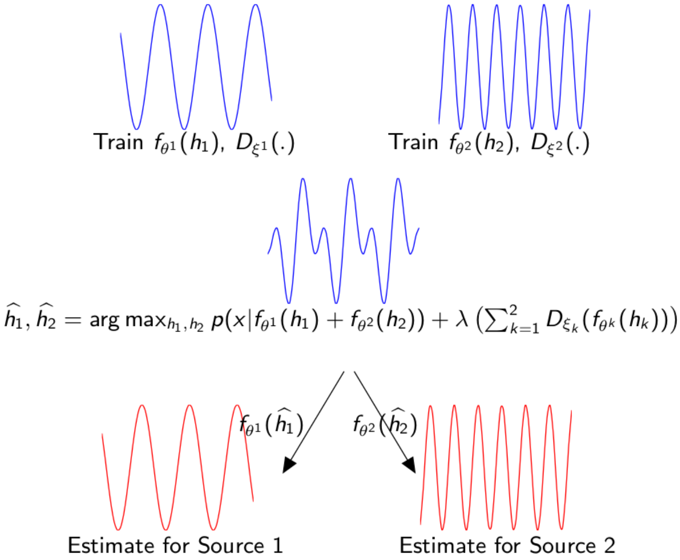
Adversarial training objective

$$\min_{\xi} \max_{\theta_k} \mathbb{E}_{s_k} \log D_{\xi_k}(s_k) + \mathbb{E}_{h_k} \log(1 - D_{\xi_k}(f_{\theta^k}(h_k)))$$

- Define the likelihood via a classifier $D(.)$.
- In testing we can use the classifier:

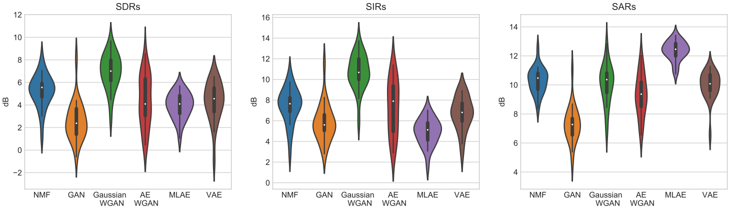
$$\max_{h_1, h_2} p_{\text{out}}(x | f_{\theta^1}(h_1) + f_{\theta^2}(h_2)) + \lambda \left(\sum_{k=1}^{2} D_{\xi_k}(f_{\theta^k}(h_k)) \right)$$

Generative Adversarial Source Separation



Results

- **Dataset:** Male-female speaker mixtures from TIMIT dataset.
  - Training set: 9 utterances for each speaker.
  - Test set: Single sentence mixture at 0dB.
  - Evaluated for 25 pairs of speakers.
- **Evaluation:** BSS eval metrics. (SIR, SAR, SDR)
- We compare NMF, Variational Autoencoders, Denoising Autoencoder, GAN, and Wasserstein GAN, all with a multilayer perceptron architecture.

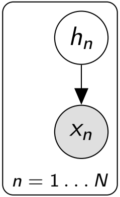


Contributions

- **Contribution 1:** We developed a neural network model which is an analog of convolutive NMF, both with convolutional and recurrent neural network architectures. (representation)
- **Contribution 2:** We showed that GANs worked better than maximum likelihood based methods on a speech source separation task.
  - This is potentially because GANs are more agnostic to output noise. (learning paradigm)

Plan

VAEs and GANs



VAEs:

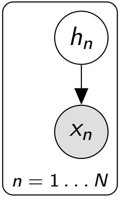
$$\begin{aligned} h &\sim \mathcal{N}(0, I) \\ x|h &\sim \mathcal{N}(f_\theta(h), \sigma^2 I) \end{aligned}$$

GANs:

$$\begin{aligned} h &\sim \mathcal{N}(0, I) \\ x|h &= f_\theta(h) \end{aligned}$$

- VAEs and GANs are very popular methods for generative model learning.

VAEs and GANs



VAEs:

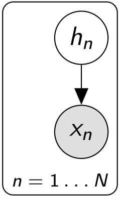
$$
\begin{aligned}
h &\sim \mathcal{N}(0, I) \\
x|h &\sim \mathcal{N}(f_\theta(h), \sigma^2 I)
\end{aligned}
$$

GANs:

$$
\begin{aligned}
h &\sim \mathcal{N}(0, I) \\
x|h &= f_\theta(h)
\end{aligned}
$$

- VAEs and GANs are very popular methods for generative model learning.
- VAE is learned by gradient descent optimizing a lower bound to likelihood.

VAEs and GANs



VAEs:

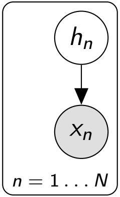
$$\begin{aligned} h &\sim \mathcal{N}(0, I) \\ x|h &\sim \mathcal{N}(f_\theta(h), \sigma^2 I) \end{aligned}$$

GANs:

$$\begin{aligned} h &\sim \mathcal{N}(0, I) \\ x|h &= f_\theta(h) \end{aligned}$$

- VAEs and GANs are very popular methods for generative model learning.
- VAE is learned by gradient descent optimizing a lower bound to likelihood.
- GAN model is an implicit generative model. It is learned by using an auxiliary "discriminator" network. Optimization is very very tricky.

VAEs and GANs



VAEs:

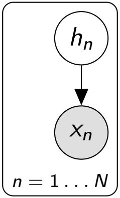
$$
\begin{aligned}
h &\sim \mathcal{N}(0, I) \\
x|h &\sim \mathcal{N}(f_\theta(h), \sigma^2 I)
\end{aligned}
$$

GANs:

$$
\begin{aligned}
h &\sim \mathcal{N}(0, I) \\
x|h &= f_\theta(h)
\end{aligned}
$$

- VAEs and GANs are very popular methods for generative model learning.
- VAE is learned by gradient descent optimizing a lower bound to likelihood.
- GAN model is an implicit generative model. It is learned by using an auxiliary "discriminator" network. Optimization is very very tricky.
- A big problem for both: They try to map a simplistic distribution such as isotropic Gaussian to the whole set of observations.

Get the base

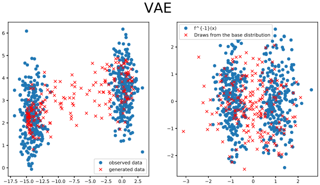


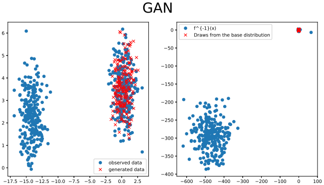
VAEs:

$$h \sim \mathcal{N}(0, I)$$
$$x|h \sim \mathcal{N}(f_\theta(h), \sigma^2 I)$$

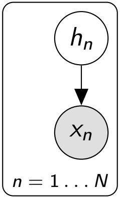
GANs:

$$h \sim \mathcal{N}(0, I)$$
$$x|h = f_\theta(h)$$





Get the base



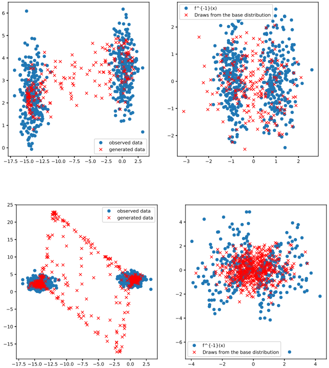
VAEs:

$$h \sim \mathcal{N}(0, I)$$
$$x|h \sim \mathcal{N}(f_\theta(h), \sigma^2 I)$$

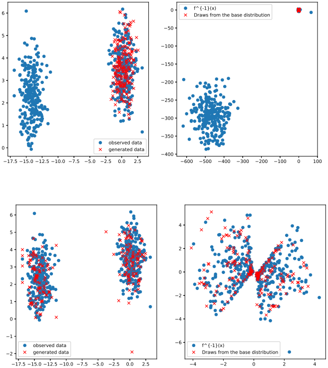
GANs:

$$h \sim \mathcal{N}(0, I)$$
$$x|h = f_\theta(h)$$

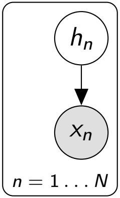
VAE



GAN



Get the base



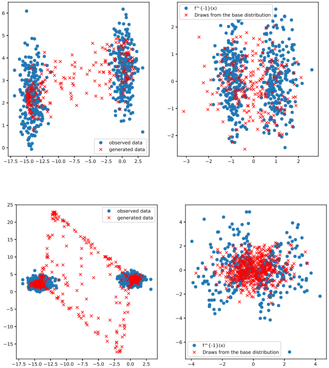
VAEs:

$$h \sim \mathcal{N}(0, I)$$
$$x|h \sim \mathcal{N}(f_\theta(h), \sigma^2 I)$$

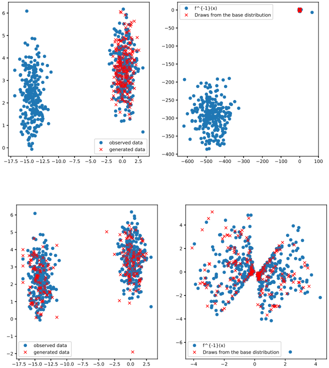
GANs:

$$h \sim \mathcal{N}(0, I)$$
$$x|h = f_\theta(h)$$

VAE



GAN

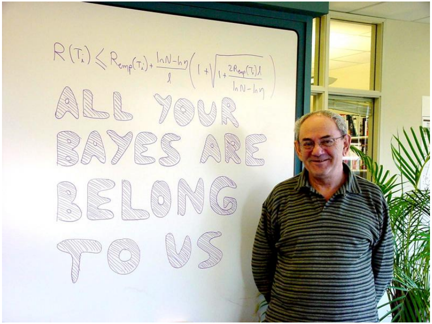


Implicit Maximum Likelihood

- This is what we do. Why? - Allows complicated base distributions.

Implicit Maximum Likelihood

- This is what we do. Why? - Allows complicated base distributions.



Maximum Likelihood for Implicit Generative Model

- Implicit Generative Models:

$$h \sim p^0_\phi(h),\ x = f_\theta(h)$$

  where, $p^0_\phi(h)$ is the base distribution and $f_\theta(h)$ is some forward mapping.
- The likelihood is given by,

$$p_{\text{model}}(x|\theta, \phi) = p^0_\phi(f_\theta^{-1}(x))V_\theta(x)$$

  where $V_\theta(x) := \left|\det \frac{\partial f_\theta^{-1}(x)}{\partial x}\right| = \left|\det \frac{\partial f_\theta(h)}{\partial h}\right|^{-1}$, which measures the volume change due to the transformation.

Maximum Likelihood for Implicit Generative Model

- Implicit Generative Models:

$$h \sim p^0_\phi(h),\ x = f_\theta(h)$$

where, $p^0_\phi(h)$ is the base distribution and $f_\theta(h)$ is some forward mapping.

- The likelihood is given by,

$$p_{\text{model}}(x|\theta, \phi) = p^0_\phi(f_\theta^{-1}(x))V_\theta(x)$$

where $V_\theta(x) := \left|\det \frac{\partial f_\theta^{-1}(x)}{\partial x}\right| = \left|\det \frac{\partial f_\theta(h)}{\partial h}\right|^{-1}$, which measures the volume change due to the transformation.

- Main problem: This requires a square transformation. No good for high dimensional structured data.

Maximum Likelihood for Implicit Generative Model

- Implicit Generative Models:

$$h \sim p_\phi^0(h),\ x = f_\theta(h)$$

  where, $p_\phi^0(h)$ is the base distribution and $f_\theta(h)$ is some forward mapping.
- The likelihood is given by,

$$p_{\text{model}}(x|\theta, \phi) = p_\phi^0(f_\theta^{-1}(x))V_\theta(x)$$

  where $V_\theta(x) := \left|\det \frac{\partial f_\theta^{-1}(x)}{\partial x}\right| = \left|\det \frac{\partial f_\theta(h)}{\partial h}\right|^{-1}$, which measures the volume change due to the transformation.
- Main problem: This requires a square transformation. No good for high dimensional structured data.
- Also joint optimization is difficult. (Joint in θ and ϕ)

IML with general mappings

Consider an autoencoder such that $f_\theta(f_\psi^{\text{enc}}(x)) \approx x$.

-Train the auto-encoder parameters θ, ψ such that:

$$\min_{\theta,\psi} \sum_n \| f_\theta(f_\psi^{\text{enc}}(x_n)) - x_n \|$$

-Fit the base distribution on the latent space such that:

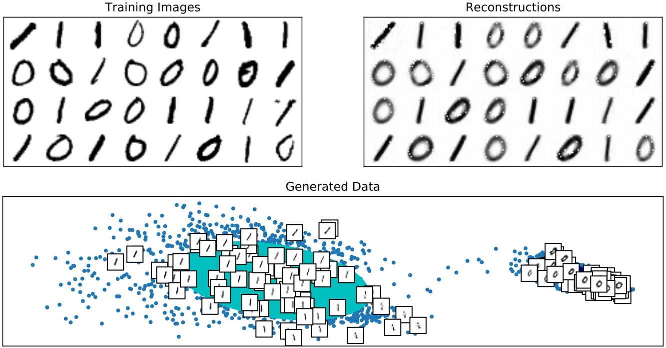
$$\max_\phi \sum_n \log p_\phi^0(f_\psi^{enc}(x_n))$$

This is approximately maximum likelihood:

$$= \max_\phi \sum_n \log p_\phi^0(f_\psi^{enc}(x_n)) + \log V(x_n)$$

Base distribution parameters are independent from the volume term.

Demonstrate the algorithm



Distributions over sequential data

- The likelihood for a sequence is given as:

$$p_{\text{model}}(x_{1:T}|\psi,\phi) = \prod_{t=1}^{T} p_\phi^0(f_\psi^{\text{enc}}(x_t)|f_\psi^{\text{enc}}(x_{1:t-1}))V(x_t),$$

Distributions over sequential data

- The likelihood for a sequence is given as:

$$p_{\text{model}}(x_{1:T}|\psi, \phi) = \prod_{t=1}^{T} p^0_\phi(f^{\text{enc}}_\psi(x_t)|f^{\text{enc}}_\psi(x_{1:t-1}))V(x_t),$$

- The algorithm: Fit an autoencoder. Then fit a sequential base distribution $p^0(.)$, such as an HMM or RNN.

Distributions over sequential data

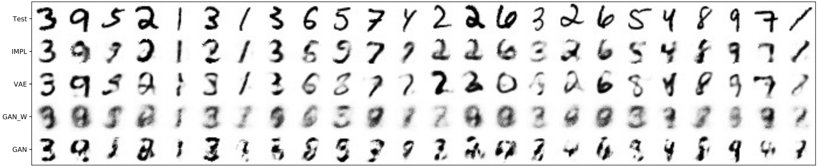
- The likelihood for a sequence is given as:

$$p_{\text{model}}(x_{1:T}|\psi,\phi) = \prod_{t=1}^{T} p^0_{\phi}(f^{\text{enc}}_{\psi}(x_t)|f^{\text{enc}}_{\psi}(x_{1:t-1}))V(x_t),$$

- The algorithm: Fit an autoencoder. Then fit a sequential base distribution $p^0(.)$, such as an HMM or RNN.
- This is a bonus that comes with this method. Not straightforward to do sequence learn with GANs and VAEs.

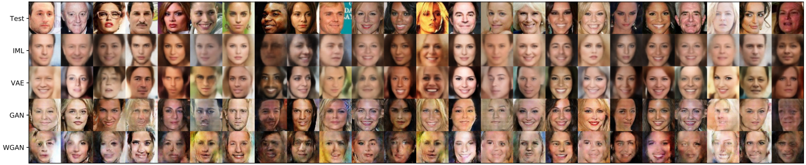
MNIST

(top) Nearest neighbor samples to test instances **(bottom)** Random samples



CELEB-A

(top) Nearest neighbor samples to test instances **(bottom)** Random samples

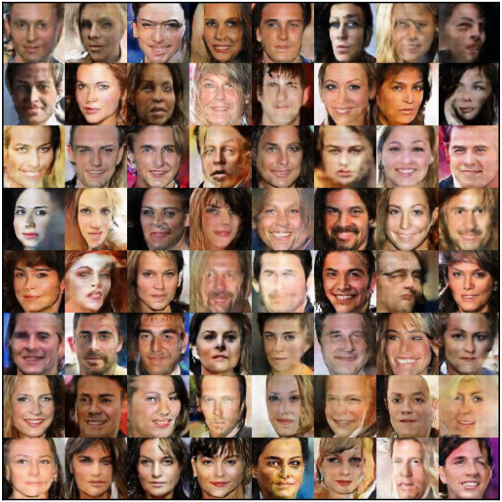


More random faces

VAE

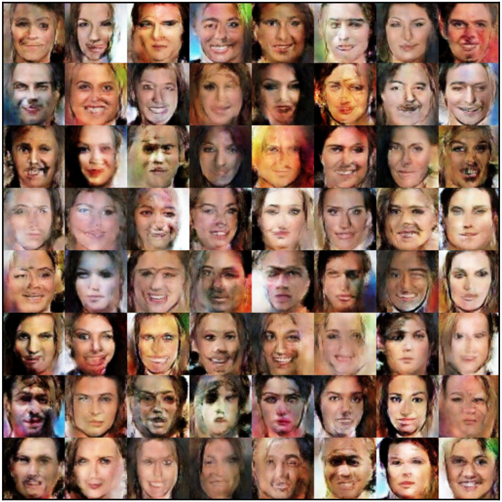
More random faces

GAN



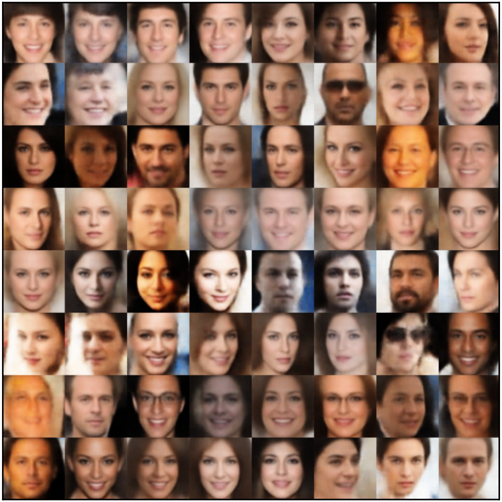
More random faces

Wasserstein GAN



More random faces

IML



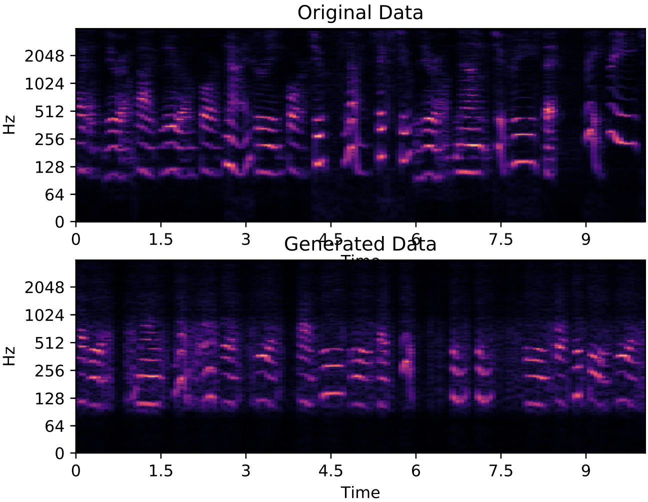
KDE Scores

Algorithm	MNIST	CELEB-A
IML	143	-8318
VAE	132	-11003
GAN	-5	-11970
WGAN	64	-12986

$$\text{KDE score} = \frac{1}{N_{\text{test}} N_{\text{samples}}} \sum_{n=1}^{N_{\text{test}}} \sum_{m=1}^{N_{\text{samples}}} \mathcal{N}(x_n^{\text{test}}; x_m^{\text{sample}}, 0.1I).$$
$$\approx \text{KL}(p_{\text{data}}(x) \| p_{\text{model}}(x|\theta))$$

Audio

We learn a distribution over overlapping windows in the time domain.



Contributions

- **Contribution 1:** We have developed a method which enables using multi-modal latent representations. (representation)

Contributions

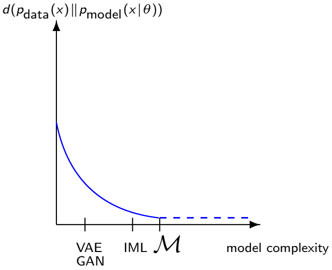
- **Contribution 1:** We have developed a method which enables using multi-modal latent representations. (representation)
- **Contribution 2:** The method does maximum likelihood for an approximate implicit model likelihood. (learning paradigm)

Contributions

- **Contribution 1:** We have developed a method which enables using multi-modal latent representations. (representation)
- **Contribution 2:** The method does maximum likelihood for an approximate implicit model likelihood. (learning paradigm)
- **Contribution 3:** We have proposed an efficient algorithm for two step optimization. The algorithm is much less sensitive to hyper-parameter fine tuning, unlike GANs. (optimization)

Contributions

- **Contribution 1:** We have developed a method which enables using multi-modal latent representations. (representation)
- **Contribution 2:** The method does maximum likelihood for an approximate implicit model likelihood. (learning paradigm)
- **Contribution 3:** We have proposed an efficient algorithm for two step optimization. The algorithm is much less sensitive to hyper-parameter fine tuning, unlike GANs. (optimization)
- Overall, we get closer to $\mathcal{M}$:



Plan

Summary

- The first half my PhD was focused more on optimization. My aim was to develop methods for “global” optimization.

Summary

- The first half my PhD was focused more on optimization. My aim was to develop methods for "global" optimization.
- However, I do admit these methods require simple models. I think for success in real data applications one needs *realistic* models. Neural networks at the moment seem to be good models for this.

Summary

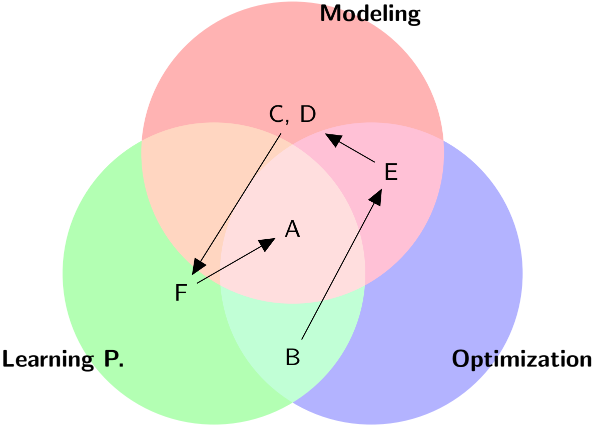
- The first half my PhD was focused more on optimization. My aim was to develop methods for “global” optimization.
- However, I do admit these methods require simple models. I think for success in real data applications one needs *realistic* models. Neural networks at the moment seem to be good models for this.

	Representation	**Learning Paradigm**	**Optimization**
Chapter 2	N.A.	MoM learning framework for HMMs	EM initialization with the MoM framwork
Chapter 3	Identifiable FHMM alternatives	N.A.	Proposed algorithms for FHMM
Chapter 4	Multi modal latent representation with IMLs	Maximum Likelihood Learning for Implicit Models	Two-Step optimization procedure
Chapter 5	Convolutive Architectures for Audio, Diagonal RNNs*	GANs in Audio	N.A.

*Not presented today for the interest of time.

Contributions in this thesis

- **A** - Learning with multi-modal latent representations in implicit generative models (UAI 2018 submission - (New)
- **B** - Method of Moments Framework for HMMs with special structure (NIPS 2014, WASPAA 2015)
- **C** - Convolutional neural nets for source separation (MLSP 2017 best paper award)
- **D** - Diagonal RNNs in symbolic music modeling (WASPAA 2017)
- **E** - Identifiable Factorial HMMs (NIPS 2015, ICASSP 2017 submissions)
- **F** - GANs for source separation (ICASSP 2018) - (New)



Conclusions

- Some models are difficult to learn, and can use help in optimization: e.g. HMMs. Method of Moments is a good initialization scheme.

Conclusions

- Some models are difficult to learn, and can use help in optimization: e.g. HMMs. Method of Moments is a good initialization scheme.
- More agnostic models can help in generalization. (Source separation with GANs)

Conclusions

- Some models are difficult to learn, and can use help in optimization: e.g. HMMs. Method of Moments is a good initialization scheme.
- More agnostic models can help in generalization. (Source separation with GANs)
- Some models are not learnable (identifiable). In cases where we care about inference, this matters. (FHMM)

Conclusions

- Some models are difficult to learn, and can use help in optimization: e.g. HMMs. Method of Moments is a good initialization scheme.
- More agnostic models can help in generalization. (Source separation with GANs)
- Some models are not learnable (identifiable). In cases where we care about inference, this matters. (FHMM)
- **My main belief after all this:**
  - An approximate learning algorithm for an exact model is better than an exact algorithm for an approximate model. (IML, convolutive NMF are good examples for this)