Maurice Frank

Unsupervised music source separation with deep generative priors

Contents

Proposal

Abstract

Research Question

Source separation is the task of finding a set of latent sources $\boldsymbol{s} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n]$ to an observed mix of those sources $\boldsymbol{m}$. The induced model proposes a mixing function $\boldsymbol{m} = f(\boldsymbol{s})$ which might just be a liner mixing $\boldsymbol{m} = \boldsymbol{A} \cdot \boldsymbol{s}$. The task is to find the inverse model $f^{-1}(\cdot)$ which retrieves $\boldsymbol{s}$.

> Can we learn an sound source separation model in an unsupervised manner. Unsupervised relating to missing pairing of sources to mixes.

$$p(s'|m) \cdot p(s) \tag{1}$$

Related works

In this chapter we discuss previous research in supervised and semi-supervised source separation.

Deep Latent-Variable Models

For our process we have observations from the data space $\boldsymbol{x} \in \mathcal{D}$ for which there exists an unknown data probability distribution $p^*(\mathcal{D})$. We collect a data set $\{\boldsymbol{x}_1 \ldots \boldsymbol{x}_N\}$ with N samples. We introduce an approximate model with density[1] $p_{\boldsymbol{\theta}}(\mathcal{D})$ and model parameters $\boldsymbol{\theta}$. Learning or modelling means finding the values for $\boldsymbol{\theta}$ which will give the closest approximation of the true underlying process:

[1] We write density and distribution interchangeably to denote a probability function.

$$p_{\boldsymbol{\theta}}(\mathcal{D}) \approx p^*(\mathcal{D}) \tag{2}$$

The model $p_{\boldsymbol{\theta}}$ has to be complex enough to be able to fit the data density while little enough parameters to be learnable. Every choice for the form of the model will *induce* biases[2] about what density we

[2] called *inductive biases*

can model, even before we maximize a learning objective using the parameters $\boldsymbol{\theta}$.

In the following described models we assume the sampled data points $\boldsymbol{x}$ to be drawn from $\mathcal{D}$ *independent and identically distributed*[3]. Therefore we can write the data log-likelihood as:

[3] meaning the sample of one datum does not depend on the other data points

$$p_{\boldsymbol{\theta}}(\mathcal{D}) = \prod_{\boldsymbol{x}\in\mathcal{D}} p_{\boldsymbol{\theta}}(\boldsymbol{x}) \tag{3}$$

$$\log p_{\boldsymbol{\theta}}(\mathcal{D}) = \sum_{\boldsymbol{x}\in\mathcal{D}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \tag{4}$$

The maximum likelihood estimation of our model parameters maximizes this objective.

To form a latent-variable model we introduce a *latent variable*[4]. The data likelihood now is the marginal density of the joint latent density:

[4] Latent variables are part of the directed graphical model but not observed.

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} \tag{5}$$

Typically we introduce a factorization of the joint. Most commonly and simplest:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z} \tag{6}$$

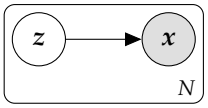

Figure 1: The graphical model with a introduced latent variable $\boldsymbol{z}$. Observed variables are shaded.

This corresponds to the graphical model in which $\boldsymbol{z}$ is generative parent node of the observed $\boldsymbol{x}$, see Figure 1. The density $p(\boldsymbol{z})$ is called the *prior distribution*.

If the latent is small, discrete, it might be possible to directly marginalize over it. If for example $\boldsymbol{z}$ is a discrete random variable and the conditional $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ is a Gaussian distribution than the data model density $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ becomes a mixture-of-Gaussians, which we can directly estimate by maximum likelihood estimation of the data likelihood.

For more complicated models the data likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ as well as the model posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ are intractable because of the integration over the latent $\boldsymbol{z}$ in Equation (6).

To formalize the search for an intractable posterior into a tractable optimization problem we follow the *variational principle*[5] which introduces an approximate posterior distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, also called the *inference model*. Again the choice of model here carries inductive biases as such that even in asymptotic expectation we can not obtain the true posterior.

[5] Michael I. Jordan et al. "An Introduction to Variational Methods for Graphical Models". In: *Machine Learning* 37.2 (1999), pp. 183–233.

Following the derivation in[6] we introduce the inference model into the data likelihood:

[6] Diederik P. Kingma and Max Welling. "An Introduction to Variational Autoencoders". In: (2019). arXiv: 1906.02691 [cs, stat], p. 20.

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x})\right] \tag{7}$$

$$= \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\right] \tag{8}$$

$$= \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\right] \tag{9}$$

$$= \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\right] \tag{10}$$

$$= \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathbb{D}_{\mathrm{KL}}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\|p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})] \tag{11}$$

Note that we separated the likelihood into two parts. The second part is the (positive) Kullback-Leibler divergence of the approximate posterior from the true intractable posterior. This unknown divergence states the 'correctness' of our approximation [7].

The first term is the *variational free energy* [8] or *evidence lower bound* (ELBO):

$$\mathrm{ELBO}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z}) = \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] \tag{12}$$

We can introduce the same factorization as in Equation (6):

$$\mathrm{ELBO}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z}) = \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] \tag{13}$$

$$= \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] \tag{14}$$

$$= -\mathbb{D}_{\mathrm{KL}}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})] + \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] \tag{15}$$

$$\tag{16}$$

Under this factorization we separated the lower bound into two parts. First the divergence of the approximate posterior from the latent prior distribution and second the data posterior likelihood from the latent [9].

The optimization of the $\mathrm{ELBO}_{\boldsymbol{\theta},\boldsymbol{\phi}}$ allows us to jointly optimize the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. Using data samples we can compute the gradient of the ELBO with respect to $\boldsymbol{\theta}$ [10] allowing for an unbiased Monte Carlo estimate of the gradient. We can *not* though do the same for the variational parameters $\boldsymbol{\phi}$, as the expectation of the ELBO is over the approximate posterior which depends on $\boldsymbol{\phi}$.

Through a change of variable of the latent random variable we can make the gradient with respect to the variational parameters tractable. With this reparameterization trick.[11]

[7] More specifically the divergence marries two errors of our approximate model. First it gives the error of our posterior estimation from the true posterior, by defintion of divergence. Second it specifies the error of our complete model likelihood from the marginal likelihood. This is called the *tightness* of the bound.

[8] TODO

[9] this will later be the reconstruction error. How well can we return to the data density from latent space

[10] $\nabla_{\boldsymbol{\theta}}\,\mathrm{ELBO}_{\boldsymbol{\theta},\boldsymbol{\phi}} \approxeq \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})$

[11] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: (2014). arXiv: 1312.6114 [cs, stat].

The VAE framework

VAE[12,13]

β-VAE[14] - intoduces βas controlling hyperparameter in the VAE objective - constraint that controls the capacityof the latent space - gives trade off between reconstruction quality and representation simplicity - similar to information bottleneck[15]

VQ-VAE[16]

[12] Kingma and Welling, "Auto-Encoding Variational Bayes".

[13] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: (2014). arXiv: 1401.4082 [cs, stat].

[14] Irina Higgins et al. "Beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: (2016).

[15] Christopher P. Burgess et al. "Understanding Disentangling in Beta-VAE". In: (2018). arXiv: 1804.03599 [cs, stat].

[16] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. "Neural Discrete Representation Learning". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 6306–6315.

Flow based models

Another class of common deep latent models are based on *normalizing* flows.[17] They use a flow for the approximate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$. A normalizing flow is a function $f(\boldsymbol{x})$ that maps the input density to a fixed, prescribed density $p(\epsilon) = p(f(\boldsymbol{x}))$, in that normalizing the density [18]. Again this is commonly set to be a factorized Gaussian distribution.

For a finite normalizing flow we consider a chain of invertible, smooth mappings.

NICE[19] - volume preserving transformations - coupling layer - triangular shape

Normalizing Flow[20]

RealNVP[21] - non-volume preserving

Glow[22] - invertible 1x1 convs - ActNorm - zero init

[23] introduced WaveNet an autoregressive generative model for raw (*time-domain*) audio. WaveNet closely similar to the earlier PixelCNN[24] but adapted for the audio domain. Unomoidified Cnns are unsuitable to the application to raw audio because of the form of data. as digital audio is ampled at a extremely high sample rate commonly 16kHz up to 44kHz the features of interest lie at scale of stringly different magnitudes. On the one hand recognizing phase, frequency of a wave might require features at those ms scales on the other hand the modelling of speech or music audio happens at the scale of seconds or minutes. As such a generative model for this domain has to cpature those different time sclaes. The wvaenet accomplishes this by using dilated convolutions a common tool in signal processing.[25] A dilated convolutions uses a kernel with an inner stride. Using a stack of dialted convolutions increases the recpetive field of the features without increasing the comutional complexity.

- gated convs -pixelcnn -lstm[26] - dilated convs - global conditioning - μ-law encoding[27] - slow cause autoreg (better with[28]) -

PixelCNN++[29]

[17] Esteban Tabak and Cristina V. Turner. "A family of nonparametric density estimation algorithms". In: *Communications on Pure and Applied Mathematics* 66.2 (2013), pp. 145–164.

[18] The extreme of this idea is, of course, an infinitesimal, continuos-time flow with a velocity field.

[19] Laurent Dinh, David Krueger, and Yoshua Bengio. "NICE: Non-Linear Independent Components Estimation". In: (2015). arXiv: 1410.8516 [cs].

[20] Danilo Jimenez Rezende and Shakir Mohamed. "Variational Inference with Normalizing Flows". In: (2016). arXiv: 1505.05770 [cs, stat].

[21] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. "Density Estimation Using Real NVP". In: (2017). arXiv: 1605.08803 [cs, stat].

[22] Diederik P. Kingma and Prafulla Dhariwal. "Glow: Generative Flow with Invertible 1x1 Convolutions". In: (2018). arXiv: 1807.03039 [cs, stat].

[23] Aäron van den Oord et al. "WaveNet: A Generative Model for Raw Audio". In: (2016). arXiv: 1609.03499 [cs].

[24] Aäron van den Oord et al. "Conditional Image Generation with PixelCNN Decoders". In: (2016). arXiv: 1606.05328 [cs].

[25] P. Dutilleux. "An Implementation of the Algorithme à Trous to Compute the Wavelet Transform". In: *Wavelets*. Ed. by Jean-Michel Combes, Alexander Grossmann, and Philippe Tchamitchian. Inverse Problems and Theoretical Imaging. Berlin, Heidelberg: Springer, 1990, pp. 298–304.

[26] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–

Sound

NSynth[30]
- In[31]
- FloWaveNet[32]

Source separation

WaveNet for Speech denoising[33]
- WaveNet-VAE unsupervised speech rep learning[34]
- Wave-U-Net[35]
- DeMucs[36]
- Source Sep in Time Domain[37]

Methodology

Datasets

ToyData

MusDB

Planning

[30] Nal Kalchbrenner et al. "Efficient Neural Audio Synthesis". In: (2018). arXiv: 1802.08435 [cs, eess].

[31] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. "WaveGlow: A Flow-Based Generative Network for Speech Synthesis". In: (2018). arXiv: 1811.00002 [cs, eess, stat].

[32] Sungwon Kim et al. "FloWaveNet : A Generative Flow for Raw Audio". In: (2019). arXiv: 1811.02155 [cs, eess].

[33] Dario Rethage, Jordi Pons, and Xavier Serra. "A Wavenet for Speech Denoising". In: (2018). arXiv: 1706.07162 [cs].

[34] Jan Chorowski et al. "Unsupervised Speech Representation Learning Using WaveNet Autoencoders". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.12 (2019), pp. 2041–2053. arXiv: 1901.08810.

[35] Daniel Stoller, Sebastian Ewert, and Simon Dixon. "Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation" (Paris, France). 2018.

[36] Alexandre Défossez et al. "Demucs: Deep Extractor for Music Sources with Extra Unlabeled Data Remixed". In: (2019). arXiv: 1909.01174 [cs, eess, stat].

[37] Francesc Lluís, Jordi Pons, and Xavier Serra. "End-to-End Music Source Separation: Is It Possible in the Waveform Domain?" In: (2019). arXiv: 1810.12187 [cs, eess].

Bibliography

Burgess, Christopher P. et al. “Understanding Disentangling in Beta-VAE”. In: (2018). arXiv: 1804.03599 [cs, stat] (cit. on p. 8).

Chorowski, Jan et al. “Unsupervised Speech Representation Learning Using WaveNet Autoencoders”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.12 (2019), pp. 2041–2053. arXiv: 1901.08810 (cit. on p. 9).

Défossez, Alexandre et al. “Demucs: Deep Extractor for Music Sources with Extra Unlabeled Data Remixed”. In: (2019). arXiv: 1909.01174 [cs, eess, stat] (cit. on p. 9).

Dinh, Laurent, David Krueger, and Yoshua Bengio. “NICE: Non-Linear Independent Components Estimation”. In: (2015). arXiv: 1410.8516 [cs] (cit. on p. 8).

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio. “Density Estimation Using Real NVP”. In: (2017). arXiv: 1605.08803 [cs, stat] (cit. on p. 8).

Dutilleux, P. “An Implementation of the Algorithme à Trous to Compute the Wavelet Transform”. In: *Wavelets*. Ed. by Jean-Michel Combes, Alexander Grossmann, and Philippe Tchamitchian. Inverse Problems and Theoretical Imaging. Berlin, Heidelberg: Springer, 1990, pp. 298–304 (cit. on p. 8).

Higgins, Irina et al. “Beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”. In: (2016) (cit. on p. 8).

Hochreiter, Sepp and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780 (cit. on p. 8).

Jordan, Michael I. et al. “An Introduction to Variational Methods for Graphical Models”. In: *Machine Learning* 37.2 (1999), pp. 183–233 (cit. on p. 6).

Kalchbrenner, Nal et al. “Efficient Neural Audio Synthesis”. In: (2018). arXiv: 1802.08435 [cs, eess] (cit. on p. 8).

Kim, Sungwon et al. “FloWaveNet : A Generative Flow for Raw Audio”. In: (2019). arXiv: 1811.02155 [cs, eess] (cit. on p. 9).

Kingma, Diederik P. and Prafulla Dhariwal. “Glow: Generative Flow with Invertible 1x1 Convolutions”. In: (2018). arXiv: 1807.03039 [cs, stat] (cit. on p. 8).

Kingma, Diederik P. and Max Welling. “Auto-Encoding Variational Bayes”. In: (2014). arXiv: 1312.6114 [cs, stat] (cit. on pp. 7, 8).

Kingma, Diederik P. and Max Welling. “An Introduction to Variational Autoencoders”. In: (2019). arXiv: 1906.02691 [cs, stat] (cit. on p. 6).

Lluís, Francesc, Jordi Pons, and Xavier Serra. “End-to-End Music Source Separation: Is It Possible in the Waveform Domain?” In: (2019). arXiv: 1810.12187 [cs, eess] (cit. on p. 9).

Paine, Tom Le et al. “Fast Wavenet Generation Algorithm”. In: (2016). arXiv: 1611.09482 [cs] (cit. on p. 8).

Prenger, Ryan, Rafael Valle, and Bryan Catanzaro. “WaveGlow: A Flow-Based Generative Network for Speech Synthesis”. In: (2018). arXiv: 1811.00002 [cs, eess, stat] (cit. on p. 9).

Recommendation G. 711. Pulse Code Modulation (PCM) of Voice Frequencies. 1988 (cit. on p. 8).

Rethage, Dario, Jordi Pons, and Xavier Serra. “A Wavenet for Speech Denoising”. In: (2018). arXiv: 1706.07162 [cs] (cit. on p. 9).

Rezende, Danilo Jimenez and Shakir Mohamed. “Variational Inference with Normalizing Flows”. In: (2016). arXiv: 1505.05770 [cs, stat] (cit. on p. 8).

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra. “Stochastic Backpropagation and Approximate Inference in Deep Generative Models”. In: (2014). arXiv: 1401.4082 [cs, stat] (cit. on p. 8).

Salimans, Tim et al. “PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications”. In: (2017). arXiv: 1701.05517 [cs, stat] (cit. on p. 8).

Stoller, Daniel, Sebastian Ewert, and Simon Dixon. “Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation” (Paris, France). 2018 (cit. on p. 9).

Tabak, Esteban and Cristina V. Turner. “A family of nonparametric density estimation algorithms”. In: *Communications on Pure and Applied Mathematics* 66.2 (2013), pp. 145–164 (cit. on p. 8).

Van den Oord, Aäron, Oriol Vinyals, and Koray Kavukcuoglu. “Neural Discrete Representation Learning”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 6306–6315 (cit. on p. 8).

Van den Oord, Aäron et al. “Conditional Image Generation with PixelCNN Decoders”. In: (2016). arXiv: 1606.05328 [cs] (cit. on p. 8).

Van den Oord, Aäron et al. “WaveNet: A Generative Model for Raw Audio”. In: (2016). arXiv: 1609.03499 [cs] (cit. on p. 8).