Adversarial Non-Negative Matrix Factorization for Single Channel Source Separation

Martin Ludvigsen

Department of Mathematical Sciences, NTNU.

Prelude

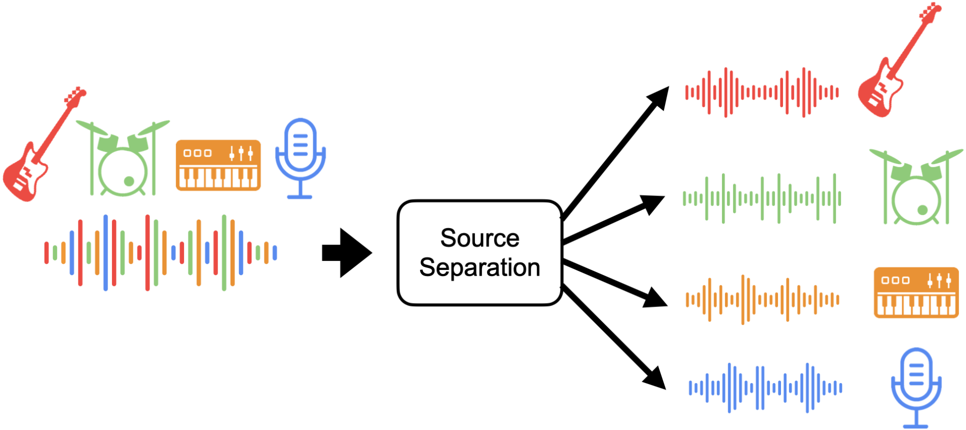


Figure: "De-mixing" music. Measure single channel. Source: https://source-separation.github.io/tutorial/landing.html

Single Channel Source Separation (SCSS)

Problem formulation

$$v = \sum_{i=1}^{S} u_i = Au,$$

$$A = \begin{bmatrix} I & \cdots & I \end{bmatrix}, \quad u = \begin{bmatrix} u_1^T & \cdots & u_S^T \end{bmatrix}^T.$$

Given measured mixed signal $v \in \mathbb{R}^m$, want to recover up to S individual source signals $u_i \in \mathbb{R}^m$, $i = 1, \ldots, S$.

Single Channel Source Separation (SCSS)

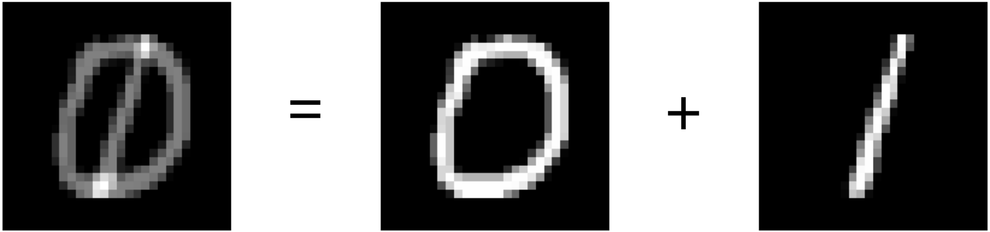
Problem formulation

$$v = \sum_{i=1}^{S} u_i = Au,$$

$$A = \begin{bmatrix} I & \cdots & I \end{bmatrix}, \quad u = \begin{bmatrix} u_1^T & \cdots & u_S^T \end{bmatrix}^T.$$

Given measured mixed signal $v \in \mathbb{R}^m$, want to recover up to S individual source signals $u_i \in \mathbb{R}^m$, $i = 1, \dots, S$.

- Linear inverse problem.
- Underdetermined $\rightarrow$ need prior information about the sources.
- Data-driven approaches are most reasonable.



v $\quad$ u_0 $\quad$ u_1

Given a mixed image $v = u_0 + u_1$, can we recover the individual images u_0 and u_1?

Non-Negative Matrix Factorization (NMF)

- Assume non-negative $M \times N$ matrix $V \approx WH$.
- W is non-negative $M \times d$ matrix.
- H is non-negative $d \times N$ matrix.
- $d \ll N, M$ is the rank of the decomposition chosen a priori.

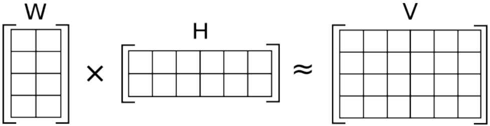


Figure: Source:
`https://en.wikipedia.org/wiki/Non-negative_matrix_factorization`

- Sparse (non-negative) dictionary learning.

$$\min_{W \geq 0} \|V - WH(V,W)\|_F^2 + \mu_W |W|_1$$

$$H(V,W) = \arg\min_{H \geq 0} \|V - WH\|_F^2 + \mu_H |H|_1 .$$

- Bi-level problem. Non-convex. Non-uniqueness.

NMF for source separation

- Assume that we have data from each individual source, stored columnwise in matrices U_i.
- During training, fit NMF for each matrix $U_i \to$ learn S non-negative bases W_i.
- During testing, want to separate v:

$$\min_{\substack{h_i \geq 0 \\ i=1,\dots,S}} \|v - \sum_{i=1}^{S} W_i h_i\|^2 + \mu_H \sum_{i=1}^{S} \|h_i\|_1.$$

- Define $W = [W_1 \cdots W_S]$, $h = [h_1^T \cdots h_S^T]^T$, write testing problem as

$$\min_{h \geq 0} \|v - Wh\|^2 + \mu_H \|h\|_1.$$

- Post-process with Wiener filter to ensure that the sources sum to v:

$$\text{Separated signals } u_i = v \odot \frac{W_i h_i}{\sum_{j=1}^{S} W_j h_j}.$$

NTNU

Example NMF separation

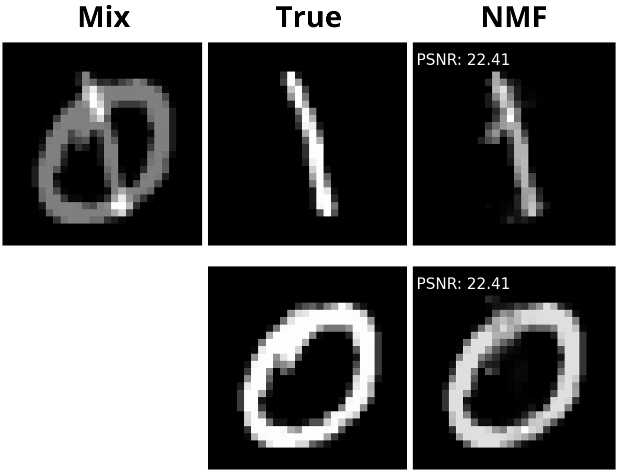


Figure: Example test separation with $N = 5000$ training data. Qualitatively decent, but feature of one source "bleeds" into other source!

Can we do better than fitting the bases individually? Can we use more of the data available?

Data setting

- Distribution of individual sources, $u_i \sim \mathbb{P}_{U_i}$.
- Distribution of measured mixed signals $v \sim \mathbb{P}_V$.
- Joint distribution $(v, u_1, \ldots, u_S) \sim \mathbb{P}_{V \times U_1 \times \ldots \times U_S} = \mathbb{P}_{V \times U}$.

Data setting

- Distribution of individual sources, $u_i \sim \mathbb{P}_{U_i}$.
- Distribution of measured mixed signals $v \sim \mathbb{P}_V$.
- Joint distribution $(v, u_1, ..., u_S) \sim \mathbb{P}_{V \times U_1 \times ... \times U_S} = \mathbb{P}_{V \times U}$.
- **Strong Supervised**: Have access to $\mathbb{P}_U, \mathbb{P}_V$ and the joint $\mathbb{P}_{V \times U}$.
- **Weak Supervised**: Have access to individual sources $\mathbb{P}_{U_i}$, mixed signals $\mathbb{P}_V$, but not joint $\mathbb{P}_{V \times U}$.
- We are interested in the weak supervised case.

Strong supervised data setting

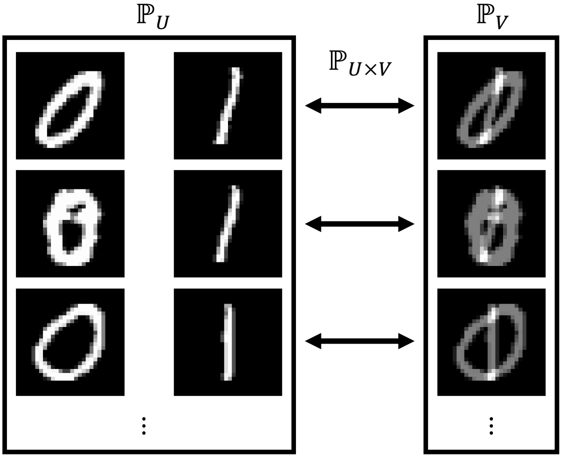


Figure: Have access to all labeled data and joint between them.

Weak supervised data setting

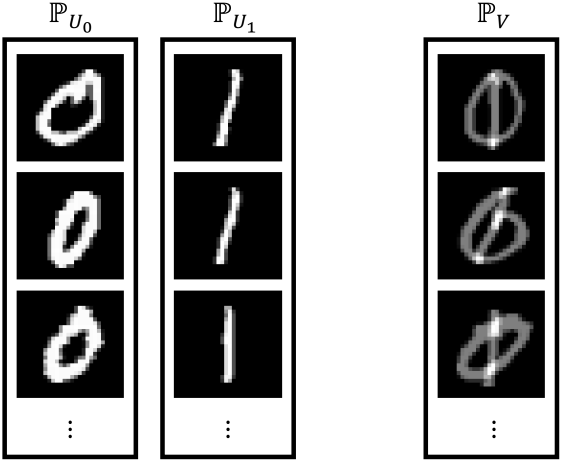


Figure: Only have access to individual labeled data, but no "links" between them. We also do not have $\mathbb{P}_U = \mathbb{P}_{U_0 \times U_1}$.

Adverserial regularization functions for source separation

Based on work by S. Lunz, O. Öktem and C. Schönlieb (*Adversarial Regularizers in Inverse Problems*, 2018), goal is to learn regularization function R.

- Define true data for source i, $u_i \sim \mathbb{P}_{U_i}$.
- Define the adverserial data for source i, $\mathbb{P}_{Z_i}$, which we choose as data from other sources and mixed data.
- R_i should be small for true data $\mathbb{P}_{U_i}$.
- R_i should be large for adversarial data to $\mathbb{P}_{Z_i}$.

Fit regularizations function by solving

$$\min_{R_i \in \Theta : \|R_i\|_L \leq 1} \mathbb{E}_{u \sim \mathbb{P}_{U_i}}[R_i(u)] - \mathbb{E}_{u \sim \mathbb{P}_{Z_i}}[R_i(u)] \tag{1}$$

where Θ is a parameterized space of functions, like neural networks, and $\|.\|_L$ denotes the Lipschitz constant.

Adverserial NMF (ANMF)

Idea: Parameterize regularization function $R(u) = D_C(u)$, where D_C is the distance to a convex set C, specifically a convex cone.

$$\min_{W_i \geq 0} \mathbb{E}_{u \sim \mathbb{P}_{U_i}}[D_{C(W_i)}(u)^2] - \mathbb{E}_{u \sim \mathbb{P}_{Z_i}}[D_{C(W)}(u)^2]$$
$$\approx \min_{W_i \geq 0} \frac{1}{N_i} \|U_i - WH(U_i, W)\|_F^2 - \frac{1}{\hat{N}_i} \|\hat{U}_i - WH(\hat{U}_i, W))\|_F^2$$
where $H(U, W) = \arg\min_{H \geq 0} \|U - WH\|$.

where true data $U_i \approx W_i H_i$ and adversarial data $\hat{U} \neq W_i \hat{H}_i$.

- If we ignore the second term, this is just standard NMF.
- With ANMF we want to both fit the true data $\mathbb{P}_{U_i}$ well and the adverserial data $\mathbb{P}_{Z_i}$ poorly.

Weighted ANMF

Problem: Tradeoff between fitting true data well and adversarial data poorly. NMF is already low complexity $\rightarrow$ NMF is bad at reconstructing data that is not in true data.

NTNU

Weighted ANMF

Problem: Tradeoff between fitting true data well and adversarial data poorly. NMF is already low complexity $\rightarrow$ NMF is bad at reconstructing data that is not in true data.

Solution: Fit a mix between NMF and ANMF

$$\min_{W_i \geq 0} (1 - \tau_A) \underbrace{\mathbb{E}_{u \sim \mathbb{P}_{U_i}}[D_{C(W_i)}(u)^2]}_{\text{NMF}} + \tau_A \underbrace{(\mathbb{E}_{u \sim \mathbb{P}_{U_i}}[D_{C(W_i)}(u)^2] - \mathbb{E}_{u \sim \mathbb{P}_{Z_i}}[D_{C(W)}](u))^2}_{\text{ANMF}}$$

$$= \min_{W_i \geq 0} \mathbb{E}_{u \sim \mathbb{P}_U}[D_{C(W)}(u)^2] - \tau_A \mathbb{E}_{u \sim \mathbb{P}_Z}[D_{C(W)}(u)^2],$$

where $0 \leq \tau_A$ is a tuning parameter chosen a priori or with hyperparameter tuning.

Low τ_A values $\rightarrow$ fit real data well.

High τ_A values $\rightarrow$ fit adverserial data poorly.

NTNU

Numerical algorithm for ANMF

Multiplicative update to fit $U \approx WH$ and adversarially to $\hat{U} \neq W\hat{H}$:

$$W \leftarrow W \odot \frac{UH^T + W\hat{H}\hat{H}^T}{WHH^T + \hat{U}\hat{H}^T + \mu_W}$$

$$H \leftarrow H \odot \frac{W^T U}{W^T WH + \mu_H}$$

$$\hat{H} \leftarrow \hat{H} \odot \frac{W^T \hat{U}}{W^T W\hat{H} + \mu_H}$$

Blue corresponds to true term (NMF), Orange corresponds to Adversarial term (ANMF).

Complexity scales with the total amount of data $\mathcal{O}(dM(N + \hat{N}))$.

Convergence

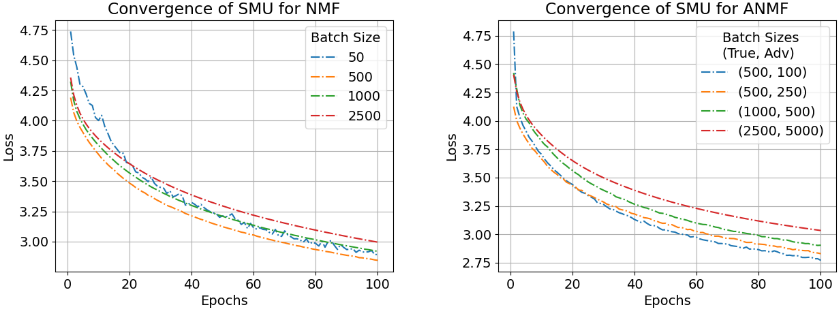


Figure: Convergence with $N = 2500$ data, $d = 32$ and $\tau_A = 0.1$. We here do a Stochastic Multiplicative Update (SMU), i.e we do the updates batchwise for W, and update all H at once. Initialize with ENMF.

Numerical experiments

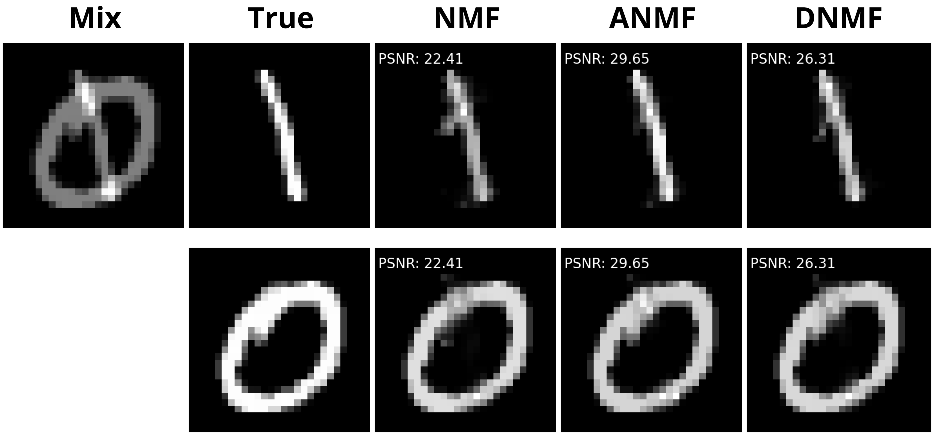
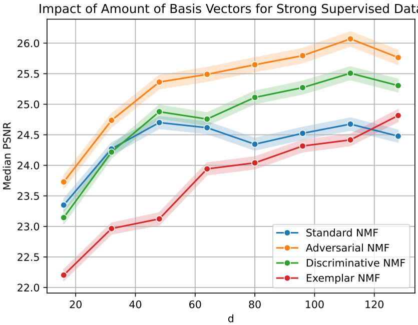


Figure: Test with $N = 5000$ strong supervised data. ANMF outperforms other methods, and is better at knowing what features belong to what source.

Strong supervision experiment



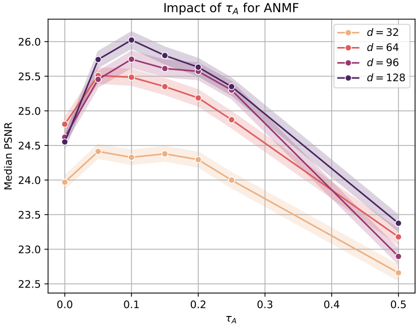


Figure: Test with $N = 5000$ strong supervised data and $N_{\text{test}} = 1000$ test data with mixes of "zero"-digits and "one"-digits. We use 100 training epochs with batch sizes $(\texttt{true}, \texttt{adv}, \texttt{sup}) = (500, 100, 500)$. For ANMF in the left plot we use $\tau_A = 0.1$.

Surprisingly ANMF outperforms DNMF, and performance increase over NMF increases with number of basis vectors d.

Data poor tuning experiment

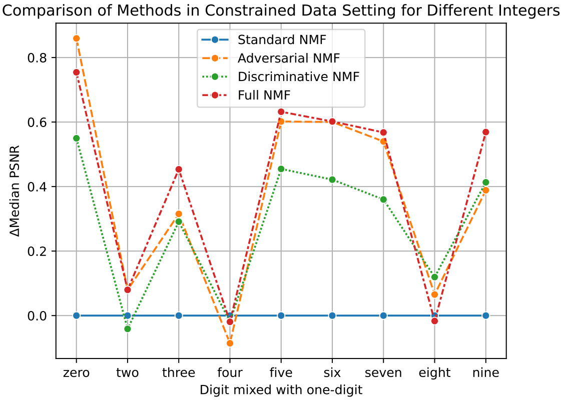


Figure: Experiment with $N_{\text{weak}} = 500$ weak supervision data and $N_{\text{strong}} = 250$ strong supervision data and $N_{\text{test}} = 1000$ test data. Each digit is mixed with a "one"-digit. For each experiment, we do parameter tuning using Random search with 10 experiments and we use $d = 64$.

Further work

- Use ANMF on speech denoising applications. Preliminary results show that ANMF can potentially ourperform existing NMF-based methods.
- ... but is outperformed by more complex deep learning methods.
- $\rightarrow$ Apply adversarial generative regularization with more complicated generative functions.
- Other applications? Tensors? SPD-valued images?

Data-driven methods roadmap

Learned distributions

Learned mappings between v and u. End-to-end NN

Learned spaces, sets, manifolds

Learned regularization functions

Hyperparameter tuning generative methods

Generative regularization SVD, NMF, AE, GANs

Learned prox/grad operators

Generative ⟷ **Discriminative**

Weak supervision ⟷ **Strong supervision**

Generative Regularization

One of the simplest ways of doing generative regularization for inverse problems is to solve

$$\min_{h \in \mathcal{H}} \|Ag(h) - v\|^2 + \mu_H \|h\|_1. \qquad (2)$$

Usually want some regularization on the latent variable to ensure uniqueness and well-posedness.

For the specific case of NMF for source separation, this is the familiar

$$\min_{h \geq 0} \|Wh - v\|^2 0 \mu_H \|h\|_1, \qquad (3)$$

where W is the concatenation of the bases W_i for all sources $i = 1, ...S$.

Generative regularization as a framework for data-driven methods

$$D_C(u) = \min_{h \in \mathcal{H}} \|u - g(h)\| = \|u - P_C(u)\| \tag{4}$$

- C learned set that contains relevant data. For NMF this is a convex cone.
- $D_C : X \to \mathbb{R}$ distance to set, regularization function.
- $P_C : X \to C$ projection onto set. For NMF this is the testing problem/lower level training problem.
- Proximal operator if C is convex.
- Generative functions are fitted to generate a distribution.
- Main problem is that generative functions tend to learn features that are too general