论文标题

DEEP NMF FOR SPEECH SEPARATION

要点

给出混合音频的能量谱或者幅度谱：

$$M = [m_1, \ldots, m_T] \in R^{F \times T}$$

假设混合音频中一共有 L 个音频源，每个源 l：

$$l \in \{1, \ldots, L\}$$

每个音频源S^l可以分解为如下两个矩阵：

$$W^l = \begin{bmatrix} | & | & | & \cdots & | \\ w_1^l & w_2^l & w_3^l & \ldots & w_{R_l}^l \\ | & | & | & \cdots & | \end{bmatrix} \in R^{F \times R_l}$$

$$H^l = \begin{bmatrix} | & | & | & \cdots & | \\ h_1^l & h_2^l & h_3^l & \ldots & h_T^l \\ | & | & | & \cdots & | \end{bmatrix} \in R^{R_l \times T}$$

其中，各自的矩阵元素分别为${w_{f,r}}^l$，${h_{r,t}}^l$。注意 W 矩阵中每个基本列向量都是非负的，H 矩阵的第 r 行包好了基本的w_r^l在每个时间步上的激活值。如下给出了基本假设：

$$\mathbf{M} \approx \sum_l \mathbf{S}^l \approx \sum_l \widetilde{\mathbf{W}}^l \mathbf{H}^l = \widetilde{\mathbf{W}}\mathbf{H}.$$

其中W~表示对矩阵 W 做了规范化。并定义代价函数：

$$\hat{\mathbf{H}} = \arg\min_{\mathbf{H}} D_\beta(\mathbf{M} \mid \widetilde{\mathbf{W}}\mathbf{H}) + \mu|\mathbf{H}|_1.$$

β = 1：KL 散度；β = 2：平方损失；μ为正则化系数，并对参数（激活）矩阵使用 L1 正则化，这样一来参数矩阵会变得十分稀疏。更新参数矩阵的方法如下：

$\mathbf{H}^0$ is initialized randomly.

for iteration $k \in \{1, \ldots, K\}$

$$\mathbf{H}^k = \mathbf{H}^{k-1} \circ \frac{\widetilde{\mathbf{W}}^T(\mathbf{M} \circ (\widetilde{\mathbf{W}}\mathbf{H}^{k-1})^{\beta-2})}{\widetilde{\mathbf{W}}^T(\widetilde{\mathbf{W}}\mathbf{H}^{k-1})^{\beta-1} + \mu}$$

在完成了 K 次迭代之后，使用如下方法从混合音频中重构出原来的 l 个音频源，其中公式的前一项可以看成一个滤波器（理解成掩码矩阵，有点像 IRM？）：

$$\widetilde{\mathbf{S}}^{l,K} = \frac{\widetilde{\mathbf{W}}^l \mathbf{H}^{l,K}}{\sum_{l'} \widetilde{\mathbf{W}}^{l'} \mathbf{H}^{l',K}} \circ \mathbf{M}.$$

上述模型称为**基本模型**，对于这种算法，一种常见的做法是在每个音频源上独立进行训练，然后再汇总它们。然而实际应用的时候效果并不好，于是有了下面的一种判别模型 **DNMF**：

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \sum_l \gamma_l D_{\beta_2}\left(\mathbf{S}^l \mid \widetilde{\mathbf{W}}^l \hat{\mathbf{H}}^l(\mathbf{M}, \mathbf{W})\right),$$

$$\hat{\mathbf{H}}(\mathbf{M}, \mathbf{W}) = \arg\min_{\mathbf{H}} D_{\beta_1}(\mathbf{M} \mid \widetilde{\mathbf{W}}\mathbf{H}) + \mu|\mathbf{H}|_1,$$

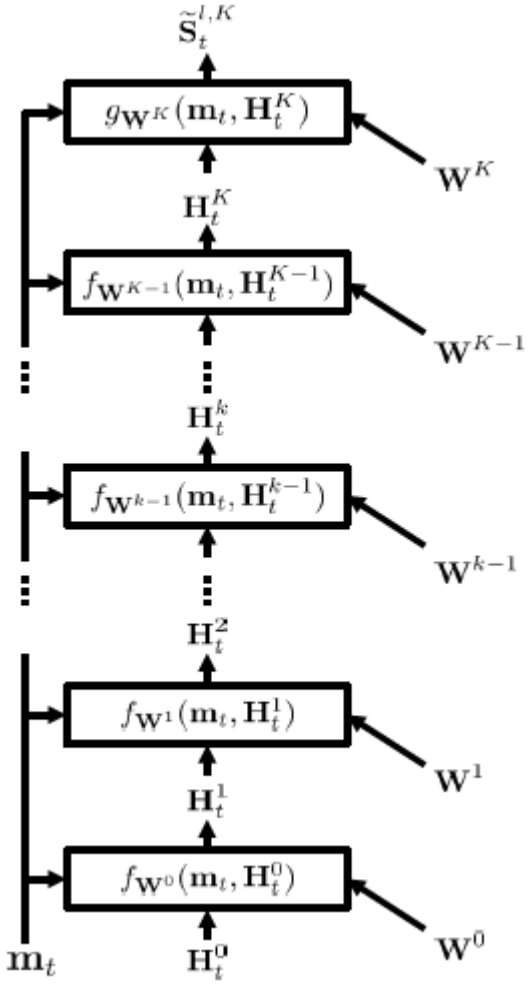
其中下式称为低级别分析目标（凸函数），上式称为高级别重构目标。下式通过减小真实的混合音频 M 和预测的混合音频W~H 之间的差距来获得最优的激活矩阵 H；上式则通过减小真实的音频源和预测的音频源之间的误差来获得最优的 W，这一误差也叫重构误差。但是这一模型的缺点在于这是一个 bi-level 优化问题，因为它有两个优化目标，这种问题处理相对比较困难。为了解决这一问题，又有了如下一种直接对重构误差进行优化的方法：

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \sum_l \gamma_l D_{\beta_2}\left(\mathbf{S}^l \mid \widetilde{\mathbf{S}}^{l,K}(\mathbf{M}, \mathbf{W})\right)$$

对于最早的**基本模型**，有一个有趣的处理：把 K 次迭代解开（unfolding）为一个深度非负的 K 层神经网络，每一层的参数$W^K, k = 0 \ldots . K$，于是该模型变成了：

$$\begin{aligned}
\mathbf{H}_t^k &= f_{\mathbf{W}^{k-1}}(\mathbf{m}_t, \mathbf{H}_t^{k-1}), \\
&= \mathbf{H}_t^{k-1} \circ \frac{(\widetilde{\mathbf{W}}^{k-1})^T(\mathbf{m}_t \circ (\widetilde{\mathbf{W}}^{k-1}\mathbf{H}_t^{k-1})^{\beta-2})}{(\widetilde{\mathbf{W}}^{k-1})^T(\widetilde{\mathbf{W}}^{k-1}\mathbf{H}_t^{k-1})^{\beta-1} + \mu}, \\
\widetilde{\mathbf{S}}_t^{l,K} &= g_{\mathbf{W}^K}(\mathbf{m}_t, \mathbf{H}_t^K) = \frac{\widetilde{\mathbf{W}}^{l,K}\mathbf{H}_t^{l,K}}{\sum_{l'} \widetilde{\mathbf{W}}^{l',K}\mathbf{H}_t^{l',K}} \circ \mathbf{m}_t.
\end{aligned}$$

对应的神经网络结构图如下：



这个新的模型称为 **DeepNMF**。下面是使用梯度下降法对网络进行更新，其中-表示梯度的负数部分，+表示梯度的正数部分：

$$\mathbf{W}^k \Leftarrow \mathbf{W}^k \circ \frac{[\nabla_{\mathbf{W}^k}\mathcal{E}]_-}{[\nabla_{\mathbf{W}^k}\mathcal{E}]_+}$$

而求梯度时具体的反向传播计算如下：

$$\left[\frac{\partial \mathcal{E}}{\partial h_{r,t}^k}\right]_+ = \sum_{r'}\left(\left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_+\left[\frac{\partial h_{r',t}^{k+1}}{\partial h_{r,t}^k}\right]_+ + \left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_-\left[\frac{\partial h_{r',t}^{k+1}}{\partial h_{r,t}^k}\right]_-\right)$$

$$\left[\frac{\partial \mathcal{E}}{\partial h_{r,t}^k}\right]_- = \sum_{r'}\left(\left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_+\left[\frac{\partial h_{r',t}^{k+1}}{\partial h_{r,t}^k}\right]_- + \left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_-\left[\frac{\partial h_{r',t}^{k+1}}{\partial h_{r,t}^k}\right]_+\right)$$

$$\left[\frac{\partial \mathcal{E}}{\partial w_{f,r}^k}\right]_+ = \sum_{t,r'}\left(\left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_+\left[\frac{\partial h_{r',t}^{k+1}}{\partial w_{f,r}^k}\right]_+ + \left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_-\left[\frac{\partial h_{r',t}^{k+1}}{\partial w_{f,r}^k}\right]_-\right)$$

$$\left[\frac{\partial \mathcal{E}}{\partial w_{f,r}^k}\right]_- = \sum_{t,r'}\left(\left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_+\left[\frac{\partial h_{r',t}^{k+1}}{\partial w_{f,r}^k}\right]_- + \left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_-\left[\frac{\partial h_{r',t}^{k+1}}{\partial w_{f,r}^k}\right]_+\right)$$

实验数据集

训练，验证和测试均使用：**Wall Street Journal (WSJ-0) corpus** of read speech and the noise recordings.（混合音频和无噪声的音频产生自不想交的部分）

训练集：7138 段发音，信噪比覆盖范围-6dB-9dB，步长为 3dB。

验证集和测试集：在上述的每个信噪比下各自有 410/330 段发音，总共 2 460 / 1 980 段发音

整个任务视为一个“speaker-independent”的任务，因为训练时和测试时的说话人是一样的。此外，来自训练集和来自验证/测试集的噪声是不一样的。

评估最后的音频分离性能的方法是：source-to-distortion ratio (SDR)

特征提取

每个特征向量将连续的 T=9 帧拼接到一起形成一个目标帧（使用了 short-time Fourier spectral **magnitudes**），而一般的 DNN 不这样拼接，直接传入 logarithmic magnitude **spectra**，使用 hann 窗的平方根，窗大小为 25ms，窗移位 10ms，形成的频带维度的大小为 200。对于基本的 NMF 模型，如果分离语音（l=1）和噪声（l=2），一般假设$R_1 = R_2$，并考虑两种情形进行实验：$R_l = 100$或者$R_l = 1000$，实验在 K=4 时改进空间还很大，K=25 则比较好。

实验分析 Baseline 1：DNN

输入(mixture spectra): x_t

输出(clean speech spectra)：

$$\mathbf{y}_t = (y_{1,t}, \ldots, y_{F,t})^T \in [0,1]^F$$

前向传播：

$$\mathbf{y}_t = \sigma(\mathbf{W}^K \tanh(\mathbf{W}^{K-1} \cdots \tanh(\mathbf{W}^1 \mathbf{x}_t)) \cdots)$$

实验结果：

Table 1. DNN source separation performance on the CHiME development set for various topologies.

SDR [dB]	Input SNR [dB]							
Topology	-6	-3	0	3	6	9	Avg.	# params
3x256	3.71	5.78	7.71	9.08	10.80	12.75	8.31	644 K
1x1024	5.10	7.12	8.84	10.13	11.80	13.58	9.43	2.0 M
2x1024	5.14	7.18	8.87	10.20	11.85	13.66	9.48	3.1 M
3x1024	4.75	6.74	8.47	9.81	11.53	13.38	9.11	4.1 M
2x1536	5.42	7.26	8.95	10.21	11.88	13.67	9.57	5.5 M

可以看出，2*1536 的网络结构的平均 SDR 比较高，9.57dB，实验使用的 DNN 是基于映射型的 DNN，使用基于掩码型的 DNN 效果会更好，这会带来 1.5dB 左右的 SDR 上的提升。

实验分析 Baseline 2：sparse NMF

$$\overline{\mathbf{W}}^l, \overline{\mathbf{H}}^l = \underset{\mathbf{W}^l, \mathbf{H}^l}{\arg\min}\, D_\beta(\mathbf{S}^l \mid \widetilde{\mathbf{W}}^l \mathbf{H}^l) + \mu|\mathbf{H}^l|_1$$

Table 2. Deep NMF source separation performance on CHiME Challenge (WSJ-0) development set.

SDR [dB]	Input SNR [dB]								
$R^l = 100$	-6	-3	0	3	6	9	Avg.	P_D	P
$K = 4,\ C = 0$ (SNMF)	2.03	4.66	7.08	8.76	10.67	12.74	7.66	-	360 K
$K = 4,\ C = 1$ (DNMF)	2.91	5.43	7.57	9.12	10.97	13.02	8.17	40 K	400 K
$K = 4,\ C = 2$	3.19	5.68	7.78	9.28	11.09	13.07	8.35	80 K	440 K
$K = 4,\ C = 3$	3.22	5.69	7.79	9.28	11.09	13.05	8.35	120 K	480 K
$K = 4,\ C = 4$	3.32	5.76	7.84	9.31	11.11	13.05	8.40	160 K	520 K
$K = 25, C = 0$ (SNMF)	4.16	6.46	8.51	9.90	11.61	13.40	9.01	-	360 K
$K = 25, C = 1$ (DNMF)	4.92	7.09	8.90	10.24	12.02	13.83	9.50	40 K	400 K
$K = 25, C = 2$	5.16	7.28	9.05	10.36	12.12	13.89	9.64	80 K	440 K
$K = 25, C = 3$	5.30	7.38	9.14	10.43	12.18	13.93	9.73	120 K	480 K
$K = 25, C = 4$	5.39	7.44	9.19	10.48	12.22	13.95	9.78	160 K	520 K
$R^l = 1000$	-6	-3	0	3	6	9	Avg.	P_D	P
$K = 4,\ C = 0$ (SNMF)	1.79	4.45	6.94	8.66	10.61	12.76	7.54	-	3.6 M
$K = 4,\ C = 1$ (DNMF)	2.94	5.45	7.60	9.15	11.00	13.06	8.20	400 K	4 M
$K = 4,\ C = 2$	3.14	5.62	7.74	9.26	11.10	13.12	8.33	800 K	4.4 M
$K = 4,\ C = 3$	3.36	5.80	7.89	9.37	11.19	13.18	8.47	1.2 M	4.8 M
$K = 4,\ C = 4$	3.55	5.95	8.01	9.48	11.28	13.23	8.58	1.6 M	5.2 M
$K = 25, C = 0$ (SNMF)	4.39	6.60	8.67	10.06	11.82	13.67	9.20	-	3.6 M
$K = 25, C = 1$ (DNMF)	5.74	7.75	9.55	10.82	12.55	14.35	10.13	400 K	4 M
$K = 25, C = 2$	5.80	7.80	9.59	10.86	12.59	14.39	10.17	800 K	4.4 M
$K = 25, C = 3$	5.84	7.82	9.62	10.89	12.61	14.40	10.20	1.2 M	4.8 M

C 表示网络中最后 C 层用于作为判别模型训练。因此有 K-C 层是作为非判别模型来训练，它使用完整的 W（包含了多个帧），而后面 C 层训练时则严格限制为一帧，这样的好处是迅速缩少参数的数量。P_D代表判别模型部分的参数数量，P 是总的参数数量。C=0 时代表 SNMF, C=1 时代表 DNMF，C 比 1 大时就是一般的

Deep NMF 了。可以看到最优的模型给出了 SDR=10.20dB，
比前文提到的 DNN 中的 SDR=9.57dB 高，而且参数数量 4.8M 比 DNN 中的 5.5M 也要少。
DNN 在改善特征和训练过程的情况下可以将 SDR 提升到 10.46dB，不过相同的任务在 RNN 上可以提升到 12.23dB，因此未来 Deep NMF 的改进版本可能会在 RNN 上着手。