An Introduction to Deep Clustering

Summarized by Tongyu Lu
Dec 2020

Introduction: Toy Example

Suppose we have a series of vectors $V = [v_n], n = 1,2,\dots,N$, where $v_n \in \mathbb{R}^K$. We want to cluster together similar vectors, what will we do?

To make things easy, let us consider $V = [v_1, v_2, v_3, v_4]$, where v_1, v_3 should be clustered together and v_2, v_4 should be clustered together. If we want to find that v_1 and v_3 have affinity, we had better calculate the Gram matrix:

$$VV^T = \begin{bmatrix} v_1^T v_1 & v_1^T v_2 & v_1^T v_3 & v_1^T v_4 \\ v_2^T v_1 & v_2^T v_2 & v_2^T v_3 & v_2^T v_4 \\ v_3^T v_1 & v_3^T v_2 & v_3^T v_3 & v_3^T v_4 \\ v_4^T v_1 & v_4^T v_2 & v_4^T v_3 & v_4^T v_4 \end{bmatrix}$$

If v_1, v_3 look similar, $v_1^T v_3$ should be big.
If we assume that each vector v_n in V are normalized, i.e. $|v_n|^2 = 1$, then the Gram matrix should look like:

$$VV^T = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Another example, if v_1, v_2, v_3 should be clustered together, the Gram matrix should look like:

$$VV^T = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Another example, if v_1, v_2, v_4 should be clustered together, the Gram matrix should look like:

$$VV^T = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

In fact, if we previously know that v_1, v_3 should be clustered together and v_2, v_4 should be clustered together, we can find another series of vectors $Y = [y_1, y_2, y_3, y_4]$, s.t. YY^T is near VV^T.
How to find Y? It is straightforward:

$$y_1 = y_3 = [1,0], \qquad y_2 = y_4 = [0,1]$$

$$YY^T = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Therefore, if there are C clusters, we can define each $y_n \in \mathrm{OneHot}(C)$
Then, if one alleges that "v_1, v_2, v_3 should be clustered together, you can assign $y_1 = y_2 = y_3 = [1,0]$, $y_4 = [0,1]$ and then compare YY^T with VV^T to check if this allege is correct.

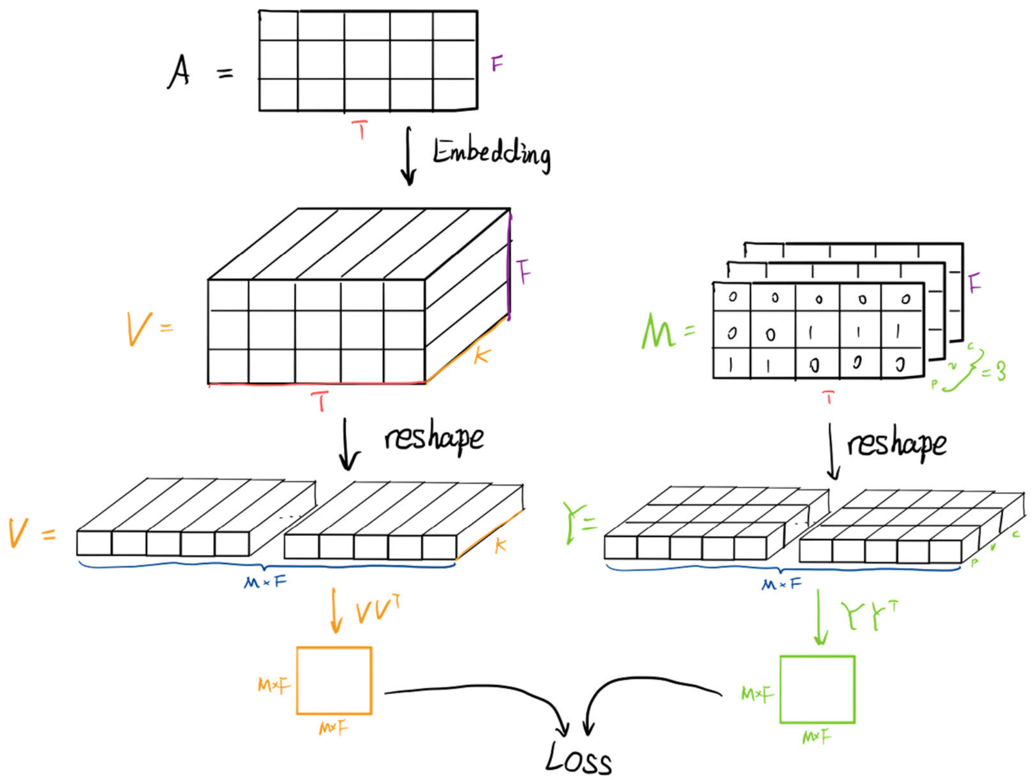
Real Example: Deep Clustering for Music Source Separation [1,2]

Now, let us change our point of view.
Consider a problem: suppose we have a music audio a, or its STFT $A_{T\times F}$. This music audio is a trio of three instruments:

piano (p), violin (v) and cello (c). Now, we want to train a model which could separate A into three instrumental parts A_p, A_v, A_c. What can we do?

First, send A into an embedding network to get embedding for each time-frequency bin. The embeddings are written as $V_{T\times F\times K} = \text{Enocder}(A_{T\times F}|\theta)$. Reshape V as $V_{(T\times F)\times K}$. This matrix looks like the vector series $V = [v_1, v_2, v_3, v_4]$ in the previous section.
Then, we have training data A_p, A_v, A_c. We can convert them into binary masks M_p, M_v, M_c which log the time-frequency activation map of each instrument. Stack them together to get the mask tensor $M_{T\times F\times 3} = [M_p, M_v, M_c]$. Reshape M as $Y = M_{(T\times F)\times 3}$. This matrix looks like the vector series $Y = [y_1, y_2, y_3, y_4]$ in the previous section.
Then you know what to do!

If we allege that at (t_0, f_0) (time t_0 and frequency bin f_0) and (t_1, f_1), there are frequency activations for piano, we should expect that $v_{t_0\times f_0}^T v_{t_1\times f_1} = 1$.
This is the same that $y_{t_0\times f_0}^T y_{t_1\times f_1} = 1 + 0 + 0 = 1$
To reach our expectations, we should make VV^T as near as YY^T. Therefore, define loss function $L = |VV^T - YY^T|_F^2$ and minimize it, we will finish learning our encoder and get the best parameter θ^*.
The whole procedure is depicted in the figure below:



Once finishing learning parameter θ^*, we can do testing for instrumental separation!
First, calculate $V = \text{Enocder}(A|\theta^*)$; then, do clustering on the $T \times F$ vectors of V and get the corresponding masks. In this way, we can handle arbitrarily-many instrumental separations.

References

[1] J. R. Hershey, Z. Chen, J. Le Roux and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, 2016, pp. 31-35, doi: 10.1109/ICASSP.2016.7471631.
[2] Keitaro Tanaka, Takayuki Nakatsuka, Ryo Nishikimi, Kazuyoshi Yoshii, Shigeo Morishima; Multi-instrument Music Transcription Based on Deep Spherical Clustering of Spectrograms and Pitchgrams; *2020 ISMIR*.