Deep Co-Clustering

Dongkuan Xu†, Wei Cheng‡, Bo Zong‡, Jingchao Ni‡, Dongjin Song‡,
Wenchao Yu‡, Yuncong Chen‡, Haifeng Chen‡ and Xiang Zhang†

†The Pennsylvania State University ‡NEC Laboratories America, Inc.

Motivation

A large volume of data matrices contain co-cluster structure between instances and features, such as customer-product matrices [3], term-document matrices [4] and gene-condition matrices [1]. Co-clustering aims to cluster instances and features simultaneously. It has shown advantages over the traditional one-sided clustering. Although the deep clustering research has shown the promising results, the research on leveraging deep representation learning for co-clustering is limited.

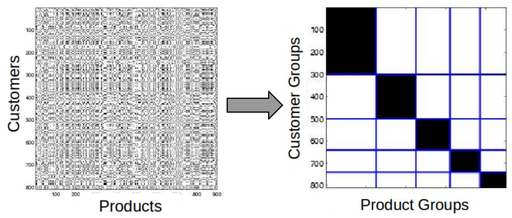


Fig. 1: Co-cluster structure in the customer-product matrix.

Problem Definition

Given instances and features represented by $\{\mathbf{x}_i\}_{i=1}^n = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ and $\{\mathbf{y}_j\}_{j=1}^d = \{\mathbf{y}_1, \cdots, \mathbf{y}_d\}$ respectively, co-clustering aims to group instances into g clusters and features into m clusters, i.e., to find maps C_r and C_c.

$$C_r : \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \rightarrow \{\hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_g\}$$
$$C_c : \{\mathbf{y}_1, \cdots, \mathbf{y}_d\} \rightarrow \{\hat{\mathbf{y}}_1, \cdots, \hat{\mathbf{y}}_m\}$$

where r and c indicate instances and features. We can reorder the instances/features such that the instances/features grouped into the same cluster are arranged to be adjacent. The resulting new data structure consists of blocks called co-clusters.

Model Architecture

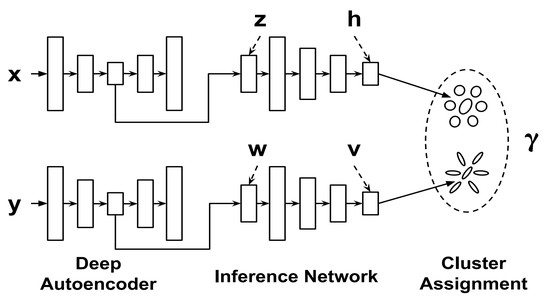


Fig. 2: DeepCC is in an end-to-end training fashion. $\mathbf{x}$ and $\mathbf{y}$ are the instances and features of the matrix data. $\mathbf{z}$ and $\mathbf{w}$ are low-dimensional representations. $\mathbf{h}$ and $\mathbf{v}$ are the outputs of inference network and utilized by a variant of GMM to produce the co-cluster assignment $\boldsymbol{\gamma}$.

Objective Function

Given instances and features denoted by $\{\mathbf{x}_i\}_{i=1}^n$ and $\{\mathbf{y}_j\}_{j=1}^d$, the objective function is

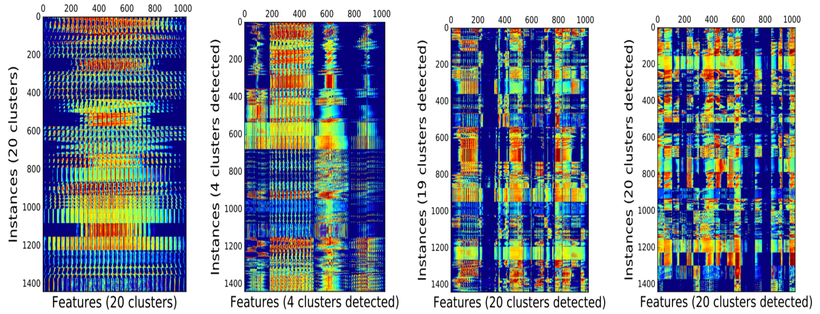
$$\min_{\theta_r, \theta_c, \eta_r, \eta_c} J = J_1 + J_2 + J_3 \tag{1}$$

$$J_1 = \frac{\lambda_1}{n} \sum_{i=1}^{n} l(\mathbf{x}_i, g_r(\mathbf{z}_i)) + \lambda_2 P_{ae}(\theta_r) + \lambda_3(-\mathcal{L}_r) + P_{inf}(\Sigma_r)$$
$$J_2 = \frac{\lambda_1}{d} \sum_{j=1} l(\mathbf{y}_j, g_c(\mathbf{w}_j)) + \lambda_2 P_{ae}(\theta_c) + \lambda_3(-\mathcal{L}_c) + P_{inf}(\Sigma_c)$$
$$J_3 = \lambda_4(1 - \frac{I(\hat{X}; \hat{Y})}{I(X; X)})$$

where J_1, J_2 are the loss for instances and features, J_3 is the cross loss. $l(\cdot)$ is the reconstruction error of autoencoder. $P_{ae}(\cdot)$ is the penalty for autoencoder to avoid overfitting. $-\mathcal{L}$ is the negative of variational lower bound of log-likehood in GMM. $P_{inf}(\cdot)$ is the sum of inverse of the diagonal entries in covariance matrices to avoid trivial solutions. $I(\cdot)$ is mutual information [2].

Results

We visualize the co-clustering results on Coil20 (Fig. 3). The data matrix is rearranged according to instance/feature cluster assignments. Result is better if co-clusters are more significant.



(a) The Coil20 dataset. (b) CCMod. (c) SCC. (d) DeepCC.

Fig. 3: Visualization of the co-clustering results on the Coil20 dataset.

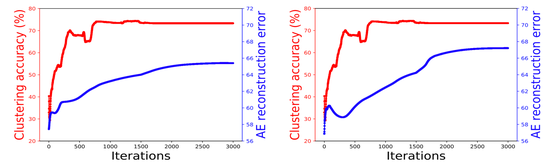
The comparison of clustering performance is shown in the following table.

Table 2: Clustering accuracy (%) comparison. The best performance on each dataset is in bold.

Dataset	k-means	SCC	SBC	CCMod	DRCC	CCInfo	SCMK	DeepCC
Coil20	58.6±2.3	51.7±0.5	66.8±1.1	21.0±2.0	53.2±2.4	60.6±3.4	65.9±0.8	**73.3±1.9**
Yale	41.8±0.7	33.7±0.3	40.0±1.3	21.4±1.4	13.6±0.4	41.8±2.0	46.6±0.5	**53.3±1.4**
Fashion-MNIST-test	54.6±0.4	44.5±0.5	45.8±0.0	28.8±0.0	44.1±1.8	51.8±2.4	-	**62.7±1.6**
Sign-MNIST-test	30.6±0.6	31.8±0.7	18.0±0.0	12.6±0.3	21.3±2.5	33.2±1.4	-	**37.0±1.3**
Citeseer	37.4±0.0	37.4±0.0	40.8±0.1	44.7±5.2	29.5±1.8	43.0±5.3	50.2±0.7	**59.3±2.1**
WebKB4	60.6±0.1	60.6±0.1	47.5±0.1	68.8±3.1	43.6±0.4	68.8±2.5	52.1±0.2	**71.8±2.8**
WebKB_cornell	55.1±2.1	58.9±0.2	54.4±0.6	55.5±2.6	42.6±0.0	56.6±2.7	49.6±0.2	**68.7±1.4**
WebKB_texas	63.9±2.6	59.4±0.2	59.0±0.3	64.5±3.0	55.1±0.0	64.1±3.6	62.0±0.6	**73.8±1.2**
WebKB_washington	65.6±2.7	60.8±0.0	51.7±1.0	68.0±2.7	46.5±0.0	67.7±2.9	65.4±0.4	**75.7±1.9**
WebKB_wisconsin	71.7±3.1	70.2±0.5	72.8±1.4	72.1±3.9	46.1±0.0	72.9±3.1	73.2±0.9	**77.4±1.4**
IMDb_movies_keywords	19.3±0.8	25.2±0.4	24.0±0.2	24.7±2.1	12.6±1.7	23.0±2.0	23.3±1.1	**30.8±1.7**
IMDb_movies_actors	15.4±0.7	20.5±0.4	20.0±0.4	20.0±1.2	14.1±2.8	15.6±0.7	15.8±1.3	**23.8±0.4**

Discussion

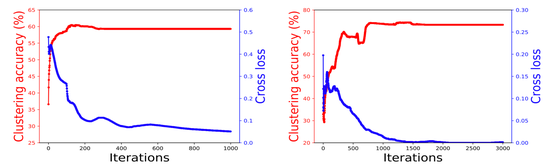
The advantage of end-to-end training is demonstrated in the following two figures.



(a) On the instances of Coil20. (b) On the features of Coil20.

Fig. 5: Reconstruction error of deep autoencoder v.s. clustering accuracy on Coil20

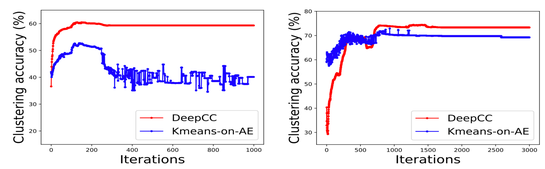
The effectiveness of cross loss is demonstrated in the following two figures.



(a) The Citeseer dataset. (b) The Coil20 dataset.

Fig. 6: Cross loss v.s. clustering accuracy on Citeseer and Coil20

The effectiveness of inference network and GMM parts is verified.



(a) The Citeseer dataset. (b) The Coil20 dataset.

Fig. 7: Clustering performance comparison between DeepCC (on the original input data) and k-means (on the output of the deep autoencoder) on Citeseer and Coil20

Acknowledgements

This work was partially supported by the National Science Foundation grant IIS-1707548.

References

[1] Xiaojun Chen et al. "Subspace weighting co-clustering of gene expression data". In: *IEEE/ACM transactions on computational biology and bioinformatics* (2017).

[2] Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. 2012.

[3] Reinhard Heckel et al. "Scalable and interpretable product recommendations via overlapping co-clustering". In: *Proceedings of ICDE*. 2017, pp. 1033–1044.

[4] Aghiles Salah, Melissa Ailem, and Mohamed Nadif. "Word Co-Occurrence Regularized Non-Negative Matrix Tri-Factorization for Text Data Co-Clustering". In: *Proceedings of AAAI*. 2018.