Senior Thesis Notes

Anuththari Gamage

January 21, 2018

1 Theory

1. Clustering with Multiple Graphs [Tang et al., 2009]

    - Linked Matrix Factorization (LMF)

      Given tensor A, each layer being a separate graph $A^{(m)}$, construct a common factor matrix $P \in \mathbf{R}^{n \times k}$ and layer-specific factor matrices $\Lambda^{(m)} \in \mathbf{R}^{k \times k}$ s.t.

      $$[\hat{P}, \hat{\Lambda}^{(1 \ldots m)}] = \arg\min \frac{1}{2} \sum_{m=1}^{M} \left\| A^{(m)} - P \Lambda^{(m)} P^T \right\|_F^2 + \frac{\alpha}{2} (\Sigma_{m=1}^{M} \left\| \Lambda^{(m)} \right\|_F^2 + \|P\|_F^2)$$

      where α is a Frobenius norm regularization term. Here, $\Lambda^{(m)}$ are not necessarily diagonal.

2. Consistency of Community Detection in Multi-layer Networks [Paul and Chen, 2017]

    - Orthogonal Linked Matrix Factorization (OLMF)

      The same approach as LMF with the constraint that P is orthogonal, which avoids the need for the regularization term. $\Lambda^{(m)} \in \mathbf{R}^{k \times k}$ s.t.

      $$[\hat{P}, \hat{\Lambda}^{(1 \ldots m)}] = \arg\min_{P^T P = I} \sum_{m=1}^{M} \left\| A^{(m)} - P \Lambda^{(m)} P^T \right\|_F^2$$

    - Baseline comparison to Spectral Clustering and similar methods
    - Tested on Multi-Layer Stochastic Block Model (MLSBM) graphs.

3. Extensions to try:

    - Replacing the adjacency matrix in LMF/OLMF with the *normalized* graph Laplacian. Does this still give the same results as using the adjacency matrix?
    - Using the NBT operator instead. Proven to capture the community structure better (Spectral Redemption) but is this necessarily the case?

2 Implementation

- MLSBM [Paul and Chen, 2017] to generate the adjacency tensor A. The algorithm is as follows:

  1. Generate community vector for nodes, assigning each node one of k clusters with equal probability
  2. Convert the above vector to a community assignment matrix $Z \in \mathbf{R}^{n \times k}$, which has a 1 in one of the k indicating the cluster for each node, represented by the nodes, and 0 elsewhere.
  3. For each layer m, create $B^{(m)} \in \mathbf{R}^{k \times k}$. If δ is the vector of k diagonal elements of B and ϵ is the vector of $k^2 - k$ off diagonal elements of $B^{(m)}$,

     $$\delta \sim U(pa, pb)$$

     $$\epsilon \sim U(a, b)$$

     where $[a, b]$ signify a short range of values and U is a uniform distribution. Only alf of ϵ is generated so that the elements are symmetric in B. The parameter p signifies signal strength ($p \geq 3$ - strong SNR). Each B is assumed to be full rank.
  4. Construct each layer of the adjacency tensor A s.t.

     $$A^{(m)} = ZB^{(m)}Z^T$$

- LMF [Tang et al., 2009]

  1. Construct adjacency tensor A
  2. Minimize objective using L-BFGS:

     $$[\hat{P}, \hat{\Lambda}^{(1 \ldots m)}] = \text{arg min} \frac{1}{2} \sum_{m=1}^{M} \left\| A^{(m)} - P\Lambda^{(m)}P^T \right\|_F^2 + \frac{\alpha}{2} (\Sigma_{m=1}^{M} \left\| \Lambda^{(m)} \right\|_F^2 + \|P\|_F^2)$$

     This is done using an alternating optimization of P and each of the Λ matrices using the given gradients [Tang et al., 2009]. More information about L-BFGS can be found in the attached article "A MATLAB Implementation of L-BFGS-B" by Brian Granzow.
  3. Cluster the *rows* of P using k-Means clustering
  4. OLMF [Paul and Chen, 2017] OLMF follows the same steps as LMF, with the addition of the orthogonality constraint on P and the removal of the *alpha* regularization. They remark that the performance of OLMF is identical to that of LMF (but the performance comparison is not shown in the paper).

3 Results so far

- No. of layers vs. accuracy for n=300, k=2, m varied 1 to 10, avg. degree = 1Results follow an increasing trend, confirming that accuracy increases with layers
- No. of layers vs. accuracy for n=300, k=6, m varied 1 to 20, avg. degree = 1Replicating result in [Paul and Chen, 2017].

4 Remarks/Issues to resolve

- LMF was implemented as detailed above since I wasn't yet sure about how to enforce orthogonality in OLMF. Does the gradient in OLMF enforce orthogonality? Check and implement
- It isn't clear how they control the average node degree of the MLSBM in [Paul and Chen, 2017], so I went with a random selection of edges in a selected subset of nodes, controlled by a density parameter.
- Unsure how to construct the data set used in [Tang et al., 2009] since it's large and isn't readily available.

References

[Paul and Chen, 2017] Paul, S. and Chen, Y. (2017). Consistency of community detection in multi-layer networks using spectral and matrix factorization methods. 61820.

[Tang et al., 2009] Tang, W., Lu, Z., and Dhillon, I. S. (2009). Clustering with multiple graphs. *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 1016–1021.