Soft Clustering on Graphs

Kai Yu*, Shipeng Yu* ‡, Volker Tresp*

* Siemens AG, Corporate Technology, and ‡ University of Munich

{kai.yu, volker.tresp}@siemens.com, spyu@dbs.ifi.lmu.de

Ludwig-Maximilians-Universität München

Introduction

A novel and simple clustering framework is proposed by factorizing pairwise data relations, which assigns data to clusters probabilistically, and naturally leads to a hierarchical clustering algorithm that exposes clustering structure in different "resolutions".

0.1 Relation-based Clustering vs. Distribution-based Clustering

- **Distance-based**: relying on a similarity (or distance) measure between data points, e.g., normalized cut [4]
  - No assumptions on density distribution of data
  - Useful when only pairwise relations are given, often encoded as a graph
  - Data assigned to clusters exclusively
- **Distribution-based**: density estimation via mixture models, e.g., Gaussian mixture models (GMMs)
  - Each component explains a cluster
  - Proper probabilistic semantics, i.e. data softly assigned to clusters
  - Sometimes restrictive assumptions (e.g. clusters must be Gaussian-like)

0.2 Hierarchical Clustering vs. Flat Clustering

- In contrast to flat clustering, hierarchical clustering organizes data into a tree of clusters, e.g. single linkage
  - Intuitive, information-rich, and indeed useful!
  - Lacking of insights beyond intuitions: any theoretical motivation?
  - Few principled solutions exist, often based on GMMs [1, 2]

0.3 Contributions of This Paper

- **Novel clustering explores data similarities encoded as a graph**
  - Relation factorization by symmetric nonnegative matrix factorization
  - Data are assigned to clusters probabilistically
  - Extended to a natural bottom-up hierarchical clustering
- **Furthermore, insights are gained from Markov random walks**
  - Graph weights are interpreted as empirical frequency of transitions
  - Optimization to maximize the likelihood of transitions
  - Hierarchical clustering employs different "lenses" to expose the clustering structure
    - Let the 1st level be the bottom one, then the l-th level clustering considers 2^{l-1} steps of transitions
    - Long-term transitions amount to a smoother and more global similarity measure
    - Somewhat like the wavelet method!

1 Clustering by Relation Factorization

Assumption: (1) The pairwise relations can be formulated as an undirected graph; (2) Data relations in one cluster are caused by a common factor.

Let $G(\mathbf{V}, \mathbf{E})$ be a weighted undirected graph with vertices $\mathbf{V} = \{v_i\}_{i=1}^n$ and edges $\mathbf{E} \subseteq \{(v_i, v_j)\}$. Let $\mathbf{W} = \{w_{ij}\}$ be the adjacency matrix, where $w_{ij} = w_{ji}$, $w_{ij} > 0$ if $(v_i, v_j) \in \mathbf{E}$ and $w_{ij} = 0$ otherwise. In practice w_{ij} indicates the similarities between data i and j.

1.1 Latent Bipartite Graph

Let $K(\mathbf{V}, \mathbf{U}, \mathbf{F})$ be a bipartite graph with adjacency $\mathbf{B}$ for edges $\mathbf{F}$ connecting vertex sets $\mathbf{U} = \{u_i\}_{i=1}^m$ and $\mathbf{V} = \{v_i\}_{i=1}^n$. Then an equivalent adjacency within $\mathbf{V}$ is [5]

$$w_{ij} = \sum_{p=1}^{m} \frac{b_{ip} b_{jp}}{\lambda_p} = \left(\mathbf{B}\mathbf{\Lambda}^{-1}\mathbf{B}^\top\right)_{ij}, \quad \lambda_p = \sum_{i=1}^{n} b_{ip}, \quad \mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_m) \tag{1}$$

A random walk explanation underlies this connection, where adjacency weights are proportional to the probability of transition between vertices. Then $\mathbf{W}$ comes from a two-step transitions $\mathbf{V} \to \mathbf{U} \to \mathbf{V}$ on the bipartite graph

$$p(v_i, v_j) = p(v_i)p(v_j|v_i) = p(v_i) \sum_p p(u_p|v_i)p(v_j|u_p) = \sum_p \frac{p(v_i, u_p)p(u_p, v_j)}{\lambda_p},$$

We assume that the given graph is induced by a "latent bipartite graph" where $n \gg m$, and $\mathbf{U}$ are clusters accounting for intra-relations of $\mathbf{V}$. Let $d_i = \sum_{p=1}^m b_{ip}$ be the degree of v_i, then $p(u_p|v_i) = b_{ip}/d_i$ tells the conditional probability of random walks from v_i to u_p, giving rise to a soft data-to-cluster assessment.

1.2 Relation Factorization by Bipartite Graph Construction

We explicitly construct an optimal bipartite graph to explain the given graph via

$$\min_{\mathbf{H}, \mathbf{\Lambda}} \ell\left(\mathbf{W}, \mathbf{H}\mathbf{\Lambda}\mathbf{H}^\top\right), \qquad \text{s. t. } \sum_{i=1}^{n} h_{ip} = 1, \mathbf{H} \in \mathbb{R}_+^{n \times m}, \mathbf{\Lambda} \in \mathbb{D}_+^{m \times m}, \tag{2}$$

where the coupling between $\mathbf{B}$ and $\mathbf{\Lambda}$ is removed by $\mathbf{H} = \mathbf{B}\mathbf{\Lambda}^{-1}$, $\ell(\cdot, \cdot)$ is a distance function, $\mathbb{D}_+^{m \times m}$ denotes the set of $m \times m$ diagonal matrices with positive diagonal entries. The minimization is done by a symmetric variant of nonnegative matrix factorization (NMF) [3].

Theorem 1.1 *For divergence distance $\ell(\mathbf{X}, \mathbf{Y}) = \sum_{ij}(x_{ij} \log \frac{x_{ij}}{y_{ij}} - x_{ij} + y_{ij})$, the cost function in (2) is non-increasing under the update rule ($\tilde{}$ denote updated quantities)*

$$\tilde{h}_{ip} \propto h_{ip} \sum_j \frac{w_{ij}}{(\mathbf{H}\mathbf{\Lambda}\mathbf{H}^\top)_{ij}} \lambda_p h_{jp}, \quad \textit{normalize} \;\; \textit{s.t.} \sum_i \tilde{h}_{ip} = 1; \tag{3}$$

$$\tilde{\lambda}_p \propto \lambda_p \sum_{ij} \frac{w_{ij}}{(\mathbf{H}\mathbf{\Lambda}\mathbf{H}^\top)_{ij}} h_{ip} h_{jp}, \quad \textit{normalize} \;\; \textit{s.t.} \sum_p \tilde{\lambda}_p = \sum_{ij} w_{ij}. \tag{4}$$

2 Hierarchical Clustering by Relation Factorization

Considering transitions in another way $\mathbf{U} \to \mathbf{V} \to \mathbf{U}$ leads to cluster similarities,

$$p(u_p, u_q) = \sum_{i=1}^{n} \frac{b_{ip} b_{iq}}{d_i} = \left(\mathbf{B}^\top \mathbf{D}^{-1} \mathbf{B}\right)_{pq}, \quad \mathbf{D} = \mathrm{diag}(d_1, \ldots, d_n) \tag{5}$$

Note that the similarity between clusters p and q takes into account weighted average of contributions from *all* the data.

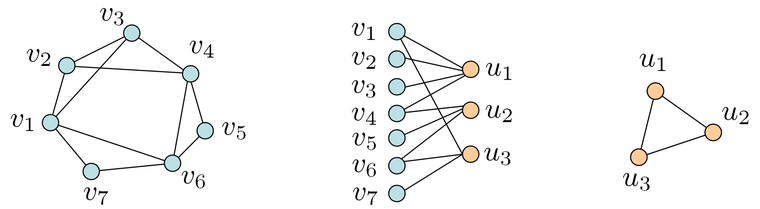


Figure. Left: the original graph representing data relations; Middle: constructed bipartite graph representing data-to-cluster relations; Right: induced graph representing cluster similarities.

We can then factorize this cluster relations to construct a higher level bipartite graph, achieving clustering of clusters. Iteratively building bipartite graphs in this way results in a hierarchical clustering algorithm. Starting at the original graph with vertices $\mathbf{V}_0$, the following algorithm builds hierarchical clustering structures $\mathbf{V}_1, \mathbf{V}_2, \ldots$ from low to higher levels.

At level l, we can derive the class memberships of data with respect to the clusters $\mathbf{V}_l$, which are propagated up from the bottom-level clusters. One can interpret this by deriving an equivalent bipartite graph $\tilde{K}_l(\mathbf{V}_0, \mathbf{V}_l, \tilde{\mathbf{F}}_l)$, which directly induces links from $\mathbf{V}_0$ to the lth level clusters $\mathbf{V}_l$. Based on the chain rule of Markov random walks, the soft assignment of $v_i \in \mathbf{V}_0$ to cluster $v_p^{(l)} \in \mathbf{V}_l$ is given by

$$p\left(v_p^{(l)}|v_i\right) = \sum_{v^{(l-1)} \in \mathbf{V}_{l-1}} \cdots \sum_{v^{(1)} \in \mathbf{V}_1} p\left(v_p^{(l)}|v^{(l-1)}\right) \cdots p\left(v^{(1)}|v_i\right) = \left(\mathbf{D}_1^{-1}\tilde{\mathbf{B}}_l\right)_{ip}, \tag{6}$$

where $\tilde{\mathbf{B}}_l = \mathbf{B}_1\mathbf{D}_2^{-1}\mathbf{B}_2\mathbf{D}_3^{-1}\mathbf{B}_3 \ldots \mathbf{D}_l^{-1}\mathbf{B}_l$ can be viewed as the equivalent adjacency matrix between data $\mathbf{V}_0$ and clusters $\mathbf{V}_l$.

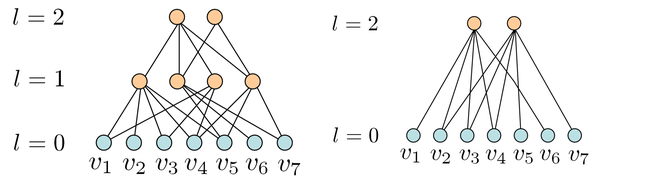


Figure. Left: an illustration to a two-level clustering; Right: induced data assignments to clusters $\mathbf{V}_2$.

3 Interpretations

3.1 Flat Clustering—Modeling One-step Random Transitions

- Suppose that from a stationary stage of a random walk on $G(\mathbf{V}, \mathbf{E})$, one observes π_{ij} one-step transitions between v_i and v_j in a unitary time frame
- As an intuition of graph view to similarities or relations, if two data points are similar or related, the transitions between them are likely to happen.
- Thus we connect observed relations with empirical frequency of transitions via $w_{ij} \propto \pi_{ij}$
- If observed transitions are i.i.d. sampled from a true distribution $p(v_i, v_j) = (\mathbf{H}\mathbf{\Lambda}\mathbf{H}^\top)_{ij}$ where a bipartite graph is behind, then the log likelihood of observed relations is

$$\mathcal{L}(\mathbf{H}, \mathbf{\Lambda}) = \log \prod_{ij} p(v_i, v_j)^{\pi_{ij}} \propto \sum_{ij} w_{ij} \log(\mathbf{H}\mathbf{\Lambda}\mathbf{H}^\top)_{ij}. \quad (7)$$

We can prove that the described one-level relation-factorization clustering maximizes the log-likelihood of observed relations.

3.2 Hierarchical Clustering—Modeling Multi-step Random Transitions

- Even two faraway vertices on $G(\mathbf{V}, \mathbf{E})$ can be connected by multi-step transitions in a longer time frame.
- Multi-step transitions induces a slower decaying and smoother similarity function on the graph, which explores more global relations of data.
- For a period t, the equivalent adjacency matrix is

$$\tilde{\mathbf{W}}^{(t)} = \mathbf{W}_0(\mathbf{D}_0^{-1}\mathbf{W}_0)^{(t-1)} = \tilde{\mathbf{W}}^{(t-1)}\mathbf{D}_0^{-1}\mathbf{W}_0.$$

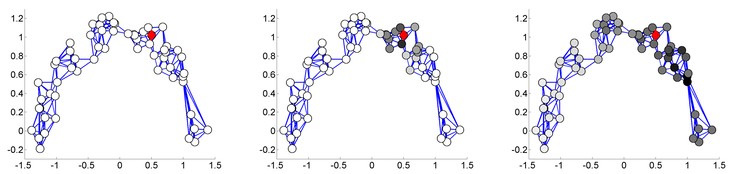


Figure. The similarities of vertices to a fixed vertex (marked in the left panel), respectively induced by 2-step (middle panel) and 64-step (right panel) transitions. The darker color indicates a higher similarity.

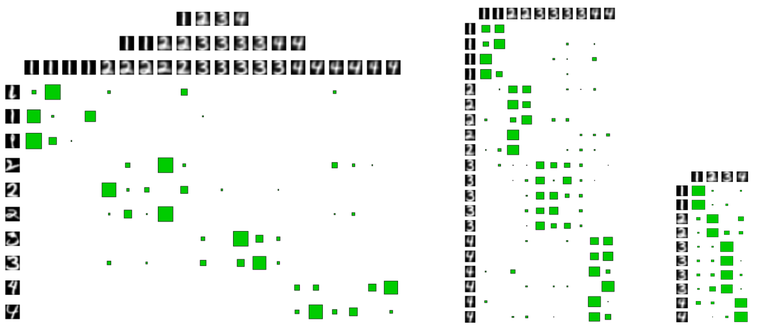
- Generally speaking, a slowly decaying similarity function captures the global structure of clusters, while a rapidly decaying similarity function only tells the local clustering structure.
- The following proposition states that in the suggested hierarchical clustering, higher-level clustering actually uses the slowly decaying similarity measures that correspond to Markov chains in a longer time period.

Proposition 3.1 *For a given hierarchical clustering structure that starts from a bottom graph $G_0(\mathbf{V}_0, \mathbf{E}_0)$ to a higher level $G_k(\mathbf{V}_k, \mathbf{E}_k)$, the obtained clusters $\mathbf{V}_l$ at level $l \leq k$ induces an equivalent adjacency matrix $\tilde{\mathbf{W}}_l$ for data points $\mathbf{V}_0$, which corresponds to the adjacency matrix $\tilde{\mathbf{W}}^{(t)}$ for a period $t = 2^l$.*

4 Empirical Studies

4.1 Visualization

We visualize hierarchical clustering results on the USPS handwritten digits 1, 2, 3 and 4. We build 10-nearest neighbor graph on the 3874 digits, and fix the sequence of cluster numbers in the hierarchy to be 100, 20, 10, 4. We can see clear clustering structure.



4.2 Clustering Comparison

The proposed method is compared against single link, complete link, k-means and spectral clustering. We used two data sets:

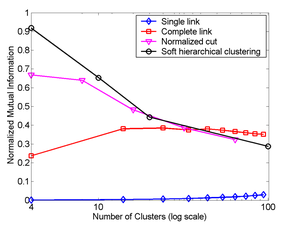
- USPS Handwritten Digits: Digits 1, 2, 3 and 4 with respectively 1269, 929, 824 and 852 images per class. Each image has 256 (16×16) features.
- 20-Newsgroup: Four groups autos, motorcycles, baseball and hockey with respectively 988, 993, 992 and 997 documents in a 8014-dimensional space.

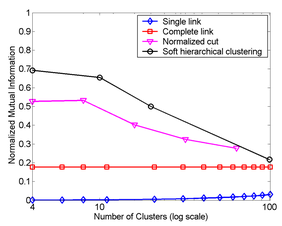
We show the confusion matrices of these methods at 4-cluster level (upper tables), and performance comparison with normalized mutual information (lower figures). This is defined to be the mutual information of the two clustering structures normalized by the maximal self entropy. Some observations:

- The proposed method obtains very good confusion matrix and achieves the best performance.
- Simple linkage methods perform badly if we care about small number of clusters.

	Single link			Complete link				Normalized cut				Soft HC				K-means			
C1	1269	0	0	0	1266	0	3	0	635	630	1	3	1254	3	8	4	1265	1	0	3
C2	927	1	0	1	584	344	1	0	2	4	744	179	1	886	33	9	17	720	95	97
C3	823	0	1	0	440	381	3	0	2	1	817	4	1	4	816	3	10	9	796	9
C4	852	0	0	0	440	5	300	107	10	6	1	835	4	8	2	838	58	20	0	774

	Single link				Complete link				Normalized cut				Soft HC				K-means			
C1	987	1	0	0	N/A	N/A	N/A	N/A	858	98	30	2	772	182	13	21	978	9	1	0
C2	992	0	1	0	N/A	N/A	N/A	N/A	79	893	16	5	42	934	5	12	992	0	1	0
C3	991	0	0	1	N/A	N/A	N/A	N/A	44	33	875	40	15	33	843	101	992	0	0	0
C4	997	0	0	0	N/A	N/A	N/A	N/A	11	8	893	85	7	21	11	958	963	0	24	10

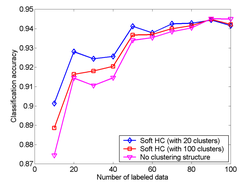


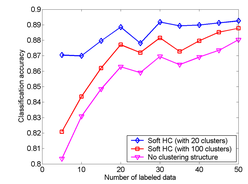


4.3 Clustering for Semi-Supervised Learning

Our last experiment evaluates the multi-step induced adjacency matrices for the underlying data. We use the same data sets and train a semi-supervised learner for classification. Some observations:

- Multi-step transition matrices always lead to better classification performance.
- Semi-supervised learning using clustering structure is much faster than that using individual data points.





5 Discussion and Future Work

Implementation issues: The time complexity of the algorithm is $\mathcal{O}(m^2 L)$ with L the number of non-zero entries in $\mathbf{W}$. This has the advantage to be very efficient if $\mathbf{W}$ is sparse. For instance for k-nearest neighbor graph, the complexity $\mathcal{O}(m^2 nk)$ scales linearly with sample size n.

Loss functions: We can also consider other loss functions in this framework, e.g., Frobenius norm for matrices. Similar iterative algorithms exist, but we lose the nice property that the performance scales linearly w.r.t. n for k-nearest neighbor graph. One interesting future work is to consider the more general Bregman divergence within this framework.

Choosing cluster numbers: In current work we fix the number of clusters at each level. This can be solved by introducing some regularization terms to the loss function and minimizing

$$\ell_{\text{reg}}\left(\mathbf{W}, \mathbf{H}\mathbf{\Lambda}\mathbf{H}^\top\right) = \sum_{i,j} w_{ij} \log \frac{w_{ij}}{\left(\mathbf{H}\mathbf{\Lambda}\mathbf{H}^\top\right)_{ij}} + \sum_p \sum_i (1-\alpha_i) \log h_{ip} + \sum_p (1-\beta_p) \log \lambda_p$$

where $\alpha_i \geq 0$ and $\beta_p \geq 0$ are free parameters to control the sparsity of clustering results. In our recent experiments, $\alpha_i = 1$, $\beta_p = 0.7$ works fine for automatically detecting the number of clusters.

References

[1] J. Goldberger and S. Roweis. Hierarchical clustering of a mixture model. In *Neural Information Processing Systems 17*, 2005.

[2] K.A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 05)*, 2005.

[3] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, pages 556–562, 2000.

[4] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. In *IEEE Conf. Computer Vision and Pattern Recognition(CVPR)*, 1997.

[5] D. Zhou, B. Schlkopf, and T. Hofmann. Semi-supervised learning on directed graphs. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17 (NIPS*04)*, 2005.