SemiSync: Semi-supervised Clustering by Synchronization

Zhong Zhang, Didi Kang, Chongming Gao, and Junming Shao(✉)

School of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu 611731, China
{zhongzhang,ddkang,chongming.gao}@std.uestc.edu.cn
junmshao@uestc.edu.cn

Abstract. In this paper, we consider the semi-supervised clustering problem, where the prior knowledge is formalized as the Cannot-Link (CL) and Must-Link (ML) pairwise constraints. We propose an algorithm called SemiSync that tackles this problem from a novel perspective: synchronization. The basic idea is to regard the data points as a set of (constrained) phase oscillators, and simulate their dynamics to form clusters automatically. SemiSync allows dynamically propagating the constraints to unlabelled data points driven by their local data distributions, which effectively boosts the clustering performance even if little prior knowledge is available. We experimentally demonstrate the effectiveness of the proposed method.

Keywords: Semi-supervised clustering · Synchronization

1 Introduction

Clustering with a priori knowledge is referred to as constrained clustering or semi-supervised clustering. Extensive studies have shown that once a priori knowledge (commonly formalized as instance-level Cannot-Link (CL) and Must-Link (ML) pairwise constraints) is incorporated, the clustering performance can be greatly improved. In this paper, we are going to tackle the semi-supervised clustering problem from a different perspective: synchronization.

Sync [4] along with its variants [7–9] are novel clustering models, which are derived from an interesting physical phenomenon synchronization. It basically defines a discrete dynamic system. The idea is to regard each object as a phase oscillator and simulate the *local* interaction behavior with its neighborhood over time under a specified interaction model. As time evolves, similar objects will synchronize together and form distinct clusters. Inspired by Sync, we develop a novel semi-supervised clustering method called SemiSync. It incorporates CL and ML constraints in an intuitive way by introducing an additional *global* interaction paradigm. Thanks to the dynamic property, once two or more objects have synchronized together over time (i.e., they have the same position), they can be

G. Li et al. (Eds.): DASFAA 2019, LNCS 11448, pp. 358–362, 2019.
https://doi.org/10.1007/978-3-030-18590-9_45

merged into a prototype. This provides a natural way to propagate the constraints within the synchronized objects. Therefore SemiSync supports to find high-quality clusterings even if only limited prior knowledge is available.

2 The Proposed Method

We first introduce some notations. $\boldsymbol{x}_i(t)$ denotes the i-th data point at the t-th time stamp. For brevity, we omit the time stamp in the following statement. Let $\boldsymbol{p}_i$ denote the i-th prototype and w_i denote the weight of $\boldsymbol{p}_i$. w_i is the number of data points that $\boldsymbol{p}_i$ represents, we will explain them later. $\mathcal{C} = \{(\boldsymbol{x}_i, \boldsymbol{x}_j)|l_i \neq l_j\}$ and $\mathcal{M} = \{(\boldsymbol{x}_i, \boldsymbol{x}_j)|l_i = l_j\}$ denote the CL and ML constraint sets, respectively, where l_i is the label of the i-th data points. For convenience, we use $\mathcal{C}(\boldsymbol{x}_i) = \{\boldsymbol{x}_j|(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{C}\}$ and $\mathcal{M}(\boldsymbol{x}_i) = \{\boldsymbol{x}_j|(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{M}\}$ to denote a set of data points that cannot link and must link to $\boldsymbol{x}_i$, respectively. $\mathcal{N}_\epsilon^c(\boldsymbol{x}_i) = \{\boldsymbol{x}_j|dist(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq \epsilon, (\boldsymbol{x}_i, \boldsymbol{x}_j) \notin \mathcal{C})\}$ denotes the exclusive ϵ-range neighborhood.

The overall clustering algorithm is a discrete dynamic system, which simply requires a interaction model and a stopping criterion. Different from [4], we define the dynamic system over a set of prototypes rather than original data points. We first present the interaction model and explain it in details.

Definition 1 (Semi-supervised Interaction Model). Given the prototype $\boldsymbol{p}_i \in \mathbb{R}^m$ and its neighbors' weights w_j at the t-th iteration, the constraint sets $\mathcal{C}$ and $\mathcal{M}$. The semi-supervised interaction model is defined as follows.

$$\boldsymbol{p}_i \longleftarrow \boldsymbol{p}_i + \frac{\alpha \sum_{\boldsymbol{p}_j \in \mathcal{N}_\epsilon^c(\boldsymbol{p}_i)} w_j \sin(\boldsymbol{p}_j - \boldsymbol{p}_i)}{\sum_{\boldsymbol{p}_j \in \mathcal{N}_\epsilon^c(\boldsymbol{p}_i)} w_j} + \frac{(1-\alpha) \sum_{\boldsymbol{p}_j \in \widetilde{\mathcal{M}}(\boldsymbol{p}_i)} w_j \sin(\boldsymbol{p}_j - \boldsymbol{p}_i)}{\sum_{\boldsymbol{p}_j \in \widetilde{\mathcal{M}}(\boldsymbol{p}_i)} w_j} - \frac{\sum_{\boldsymbol{p}_j \in \mathcal{C}(\boldsymbol{p}_i)} a_{ij} w_j \sin(\boldsymbol{p}_j - \boldsymbol{p}_i)}{\sum_{\boldsymbol{p}_j \in \mathcal{C}(\boldsymbol{p}_i)} w_j}, \tag{1}$$

Note that on the right side the second term is the original local synchronization interaction [4]. The last two terms are the global synchronization and desynchronization interaction for the ML and CL constraints, respectively. $\widetilde{\mathcal{M}}(\boldsymbol{p}_i) = \mathcal{M}(\boldsymbol{p}_i) \setminus \mathcal{N}_\epsilon^c(\boldsymbol{p}_i)$ makes no duplicate interaction. $a_{ij} = e^{-\frac{||\boldsymbol{p}_i - \boldsymbol{p}_j||^2}{2\epsilon^2}}$ is a decay weight for desynchronization. Since we only need to desynchronize cannot-link points when they are significantly close and prevent side effect from other distant cannot-link points. $\alpha \in [0, 1.0]$ is a parameter that balances the local and global synchronization. We need to restrict the synchronization coupling strength to 1 to ensure convergence.

During the dynamic interaction process some data points may synchronize together, i.e., they have the identical positions at the end of the t-th time stamp. A prototype is a representative of the synchronized data points. The synchronized data points will have the identical interaction behaviors, thus we can replace the original data points with prototypes with proper weights. The advantages are, the number of prototypes is monotonically decreasing as the iteration proceeds, which significantly alleviates the computation and memory burden. More importantly, it provides a natural way to propagate constraints.

We assume the pairwise constraint links are local consistent. Thereby, a prototype can inherit all the constraints from the merging points. However, it must be careful with the conflict. Suppose we have two synchronized points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, they can merge into a prototype $\boldsymbol{p}_i$ only when the following two conditions are satisfied: (1) $\mathcal{M}_c(\boldsymbol{x}_i) \cap \mathcal{C}(\boldsymbol{x}_j) = \emptyset$; (2) $\mathcal{M}_c(\boldsymbol{x}_j) \cap \mathcal{C}(\boldsymbol{x}_i) = \emptyset$. $\mathcal{M}_c(\boldsymbol{x}_i)$ is a must-link closure of $\boldsymbol{x}_i$ since ML constraints are transitive. Note the constraints are propagated in a local way, which prevents the error constraints from spreading.

Finally, we define an adjusted cluster order parameter r_a to indicate convergence. The algorithm stops when r_a reaches to 1.0 or barely changes.

Definition 2 (Adjusted Cluster Order Parameter). The adjusted cluster order parameter characterizes the degree of synchronization defined as follows.

$$r_a = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{N}_\epsilon^c(\boldsymbol{p}_i) \cup \mathcal{M}(\boldsymbol{p}_i)|} \sum_{\boldsymbol{p}_j \in \mathcal{N}_\epsilon^c(\boldsymbol{p}_i) \cup \mathcal{M}(\boldsymbol{p}_i)} e^{-||\boldsymbol{p}_j - \boldsymbol{p}_i||}. \tag{2}$$

3 Experiments

We evaluate the semi-supervised clustering performance of the proposed method on eight real-world data sets from the UCI and UCR repositories. For comparison, we select six typical different type of semi-supervised clustering algorithms. It includes MPCK-means [3], LCVQE [5], CECM [2], C1SP [6], CPSNMF [10] and SKMS [1]. We use two unsupervised clustering algorithms Sync and K-means as the baseline. All data sets are normalized to $[0, \pi]$. For all algorithms, we exhaustively tune their parameters to achieve the best performance.

Figure 1 shows the semi-supervised clustering results on all eight real-world data sets while varying the number of pairwise constraints. Overall, SemiSync achieves the best results on all data sets except for the Car data set, but it still yields the second best result. SemiSync can work well when only a few constraints are incorporated. The rationale is that SemiSync propagates the constraints building upon the local and global interactions, simultaneously.

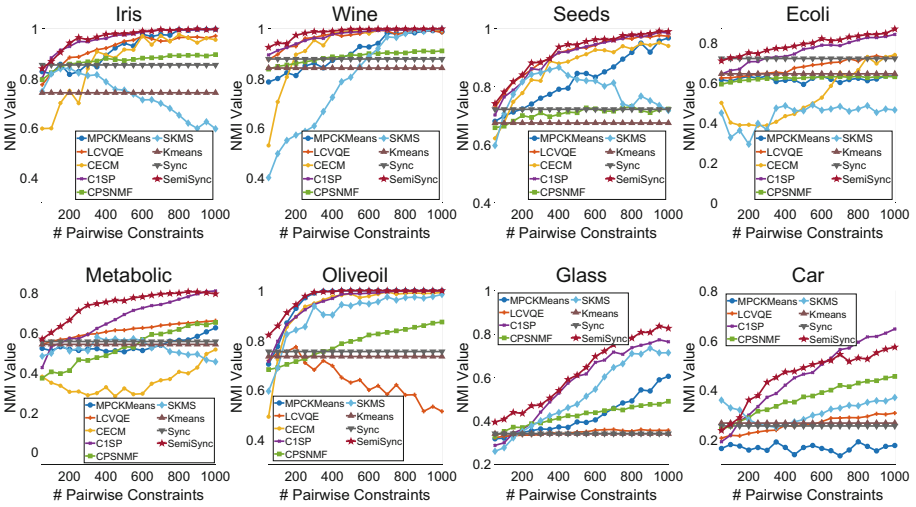

Fig. 1. Semi-supervised clustering results. *Some experiments are aborted due to unsolvable issues: CECM on Plan, Glass, Car.

4 Conclusion

We propose a novel semi-supervised clustering algorithm SemiSync from a different perspective: synchronization. SemiSync utilizes the local and global interaction paradigms to preserve the intrinsic structure of the data set and incorporate the pairwise constraints. Besides, SemiSync supports an intuitive constraint propagation, which helps improve the clustering performance. We experimentally demonstrate the effectiveness of the proposed method.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (61403062, 61433014, 41601025), Science-Technology Foundation for Young Scientist of SiChuan Province (2016JQ0007), Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China (161062) and National key research and development program (2016YFB0502300).

References

1. Anand, S., Mittal, S., Tuzel, O., Meer, P.: Semi-supervised kernel mean shift clustering. IEEE Trans. Pattern Anal. Mach. Intell. **36**(6), 1201–1215 (2014)
2. Antoine, V., Quost, B., Masson, M.H., Denoeux, T.: CECM: constrained evidential C-means algorithm. Comput. Stat. Data Anal. **56**(4), 894–914 (2012)
3. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: ICML, p. 11 (2004)
4. Böhm, C., Plant, C., Shao, J., Yang, Q.: Clustering by synchronization. In: KDD, pp. 583–592 (2010)

5. Pelleg, D., Baras, D.: K-means with large and noisy constraint sets. In: Kok, J.N., Koronacki, J., Mantaras, R.L., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS, vol. 4701, pp. 674–682. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74958-5_67
6. Rangapuram, S.S., Hein, M.: Constrained 1-spectral clustering. In: AISTATS, vol. 30, p. 90 (2012)
7. Shao, J., He, X., Böhm, C., Yang, Q., Plant, C.: Synchronization-inspired partitioning and hierarchical clustering. IEEE Trans. Knowl. Data Eng. **25**(4), 893–905 (2013)
8. Shao, J., Wang, X., Yang, Q., Plant, C., Böhm, C.: Synchronization-based scalable subspace clustering of high-dimensional data. Knowl. Inf. Syst. **52**(1), 83–111 (2017)
9. Shao, J., Yang, Q., Dang, H.V., Schmidt, B., Kramer, S.: Scalable clustering by iterative partitioning and point attractor representation. ACM Trans. Knowl. Discov. Data **11**(1), 5 (2016)
10. Wang, D., Gao, X., Wang, X.: Semi-supervised nonnegative matrix factorization via constraint propagation. IEEE Trans. Cybern. **46**(1), 233–244 (2016)