cube2net: Efficient Query-Specific Network Construction with Data Cube Organization

[Extended Abstract]

Carl Yang, Jiawei Han
UIUC, Urbana, Illinois 61801, USA
{jiyang3, hanj}@illinois.edu

I. Introduction

Networks provide a natural and generic way for modeling the interactions of objects, upon which various tasks can be performed, such as node profiling, community detection and link prediction [1], [2]. However, as real-world networks are becoming more massive and complex each day, various network mining algorithms need to be frequently developed or improved to scale up, but such innovations are often non-trivial, if not impossible. Moreover, the quality of networks taken by these algorithms is often questionable: Do the networks include all necessary information, and is every piece of information in the networks useful?

While existing network mining algorithms mostly focus on more complex models for better capturing of the given network proximities and structures, in this work, for the first time, we draw attention to the fact that network mining tasks are often specified on *particular sets of objects of interest*, which we call *queries*, and advocate for *query-specific network construction*, where the goal is to construct networks that are most *relevant* to the queries.

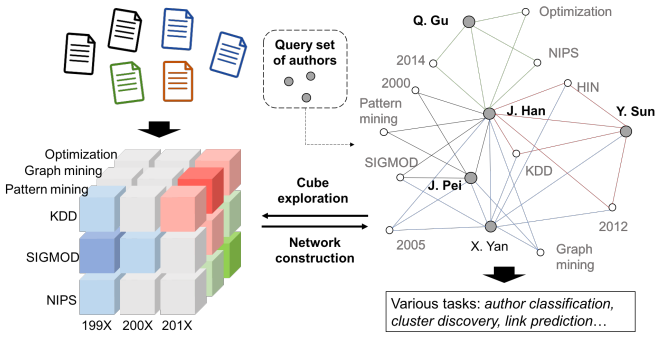

Fig. 1. *cube2net*: A running example on DBLP.

Under the philosophy of the well-developed technology of *data cube* for large-scale multi-dimensional data management, massive complex real-world networks can be decomposed into small substructures residing in fine-grained multi-dimensional multi-granular *cube cells w.r.t.* their essential properties [3], [4]. Assuming proper data cubes can be efficiently constructed for particular networks automatically, we formulate the problem of of this work as follows.

Definition I.1. *Cube-Based Query-Specific Network Construction. Given a massive network with objects organized in a data cube and a query set of objects, find a set of cells, so that objects in the cells are the most relevant to the query.*

Figure 1 gives a toy example of query-specific network construction. Consider the massive author network of DBLP[1]. The task is to find pairs of close collaborators within a particular research group. Only retaining the co-author links within the group and ignoring all outside collaborations clearly lead to significant information loss, while incorporating all co-author links in the whole network is too costly and brings in lots of useless data and noise. Based on the fact that the whole network can be partitioned into fine-grained multi-dimensional cube cells like <200X, KDD, Graph mining>, <201X, ACL, Text mining>, we can look for a few subnetworks that are the most relevant to the considered group (*e.g.*, by looking at their overlap with the group), and leverage the union of them to serve as the query-specific network.

II. cube2net

Data cube has been widely used to organize large-scale multi-dimensional data. With well-designed cube structures, it can largely boost various downstream data analytics, mining and summarization tasks [3], [4]. In the data cube, each object is assigned into a multi-dimensional *cell* which characterizes its properties from multiple aspects.

In this work, we propose and design *cube2net*, a simple and effective reinforcement learning algorithm over the data cube structures, to efficiently find a near optimal solution for the combinatorial problem of cube-based query-specific network construction. In the algorithm, the state is represented by our novelly designed distributed cell embeddings which capture the semantic proximities among cells in multiple dimensions, whereas the reward is designed to optimize the overall relevance between the set of selected subnetworks and the query. In this way, *cube2net* efficiently improves the utility estimation of various related cells regarding relevance to the query by exploring single cells, thus approaching the optimal combination of relevant cells while avoiding the enumeration of all possible combinations. Details of the framework will be described in a full version of this work.

[1] http://dblp.uni-trier.de/

Net. Con. Algorithm	Effectiveness (F1)			Efficiency	
	DeepWalk	*LINE*	*node2vec*	time	size
NoCube	0.7034	0.5620	0.6195	2s	116
NoCube+	0.6559	0.5263	0.5812	5s	3,853
NoCube++	0.6794	0.5247	0.5874	62s	55,724
CubeRandom	0.5839	0.5087	0.5748	4s	2,512
CubeGreedy	0.7445	0.5988	0.6432	3,082s	1,464
cube2net	**0.7628**	**0.6295**	**0.6913**	296s	525
NoCube	0.4336	0.3044	0.3708	3s	4,236
NoCube+	0.5207	0.3212	0.5113	84s	74,459
NoCube++	0.5515	0.3261	0.4984	1,128s	434,941
CubeRandom	0.4018	0.2972	0.3416	4s	21,126
CubeGreedy	0.6125	0.3509	0.5768	4,194s	10,046
cube2net	**0.6447**	**0.3718**	**0.6214**	314s	6,842

TABLE I
PERFORMANCE OF COMPARED ALGORITHMS ON TWO SETS OF QUERIES (SMALLER SET ON TOP AND LARGER SET ON BOTTOM).

III. Experimental Evaluation

In this section, we provide primitive results towards the effectiveness and efficiency of *cube2net* on author clustering with the DBLP network dataset. It contains semi-structured scientific publications, with their corresponding authors, years, venues and textual contents. We construct a simple data cube with dimensions *year*, *venue* and *topic*, where each cell holds the corresponding *authors* and their *coauthor* links.

For evaluations, we use two sets of authors with ground-truth labels published by [5]. The smaller set includes 116 authors from 4 research groups, and the larger set includes 4,236 authors from 4 research areas. To provide a comprehensive evaluation, we use three popular algorithms, *i.e.*, *DeepWalk* [6], *LINE* [7] and *node2vec* [8] to compute network embedding before the standard K-means clustering and compute the *F1 similarity* against the ground-truth.

We compare with five baselines described as follows

- **NoCube**: Without a data cube, this method adds no additional object to the query network.
- **NoCube+**: Without a data cube, this method adds all objects that are directly linked with the query network.
- **NoCube++**: Without a data cube, this method adds all objects that are within two steps away from the query network.
- **CubeRandom**: With a proper data cube, this method adds objects in m random cells to the network.
- **CubeGreedy**: With the same data cube, this method searches through all cells for m times, and greedily add the objects in one cell at each time to optimize the same quality function as cube2net.

1) Performance Comparison with Baselines: Table I shows the performance of compared algorithms. As for effectiveness, the network constructed by *cube2net* is able to best facilitate network mining around the query. As can be observed, (1) blindly adding neighbors into the network without a cube organization or randomly adding cells can hurt the task performance; (2) with a proper cube structure, greedily adding cells *w.r.t.* our quality function can significantly boost the task performance; however, (3) the performance of *CubeGreedy* is still inferior to *cube2net*, which confirms our arguments that the task of network construction is essentially a combinatorial problem, which requires a globally optimal solution that can be efficiently achieved only by reinforcement learning.

As for efficiency, (1) without cube organization, the network can easily get too large, which requires significant network construction time, and leads to long runtimes of network mining algorithms; (2) the sizes of constructed networks are much more controllable with a proper cube organization, because we can easily set the number of cells to add; (3) even with a proper cube, greedily searching the cube at each step to select the proper cells is extremely time-consuming– on the contrary, *cube2net* efficiently explores the cube structures with reinforcement learning and is able to find the particular set of cells to construct the most relevant subnetwork, which also makes the downstream network mining more efficient.

Comparing the results on the two sets of query objects of different sizes, we further find that, (1) as the query set of authors becomes larger, blindly bringing in neighbors leads to much larger networks, which can make subsequently network mining algorithms slower. Such low efficiency is exactly what we want to avoid by aiming at query-specific network construction in this work; (2) when the query set becomes much larger, the runtimes of the cube-based algorithms only increase a little, since they still work on the same well organized cube structure by evaluating the utility of cells rather than individual nodes, indicating the power of the data cube organization.

IV. Conclusions

We demonstrate the power of *cube2net* as a universal framework for improving network mining via query-specific network construction. Since the current designs of both data cube and reinforcement learning are primitive, many improvements and applications can be explored in future works.

Acknowledgements

Research was sponsored in part by U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA under Agreement No. W911NF-17-C-0099, National Science Foundation IIS 16-18481, IIS 17-04532, and IIS-17-41317, DTRA HDTRA11810026, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov).

References

[1] C. Yang, L. Zhong, L.-J. Li, and L. Jie, "Bi-directional joint inference for user links and attributes on large social graphs," in *WWW*, 2017, pp. 564–573.

[2] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, "Bridging collaborative filtering and semi-supervised learning: A neural approach for poi recommendation," in *KDD*, 2017, pp. 1245–1254.

[3] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[4] C. Yang, D. Teng, S. Liu, S. Basu, J. Zhang, J. Shen, C. Zhang, J. Shang, L. Kaplan, T. Harratty, and J. Han, "Cubenet: Multi-facet hierarchical heterogeneous network construction, analysis, and mining," in *KDD*, 2019.

[5] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *VLDB*, vol. 4, no. 11, pp. 992–1003, 2011.

[6] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*. ACM, 2014, pp. 701–710.

[7] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW*, 2015, pp. 1067–1077.

[8] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *KDD*, 2016, pp. 855–864.