Pattern Recognition 48 (2015) 2751–2760

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A convergence theorem for graph shift-type algorithms

Xuhui Fan *, Longbing Cao

Advanced Analytics Institute, University of Technology, Sydney, Australia

ARTICLE INFO

Article history:
Received 13 January 2014
Received in revised form
19 January 2015
Accepted 18 February 2015
Available online 27 February 2015

Keywords:
Convergence proof
The Robust Graph mode seeking by Graph Shift (RGGS) algorithm
Dominant Sets
Pairwise Clustering
Zangwill's theory

ABSTRACT

The Robust Graph mode seeking by Graph Shift (Liu and Yan, 2010) (RGGS) algorithm represents a recent promising approach for discovering dense subgraphs in noisy data. However, there are no theoretical foundations for proving the convergence of the RGGS algorithm, leaving the question as to whether an algorithm works for solid reasons. In this paper, we propose a generic theoretical framework consisting of three key Graph Shift (GS) components: the simplex of a generated sequence set, the monotonic and continuous objective function and closed mapping. We prove that the GS-type algorithms built on such components can be transformed to fit Zangwill's theory, and the sequence set generated by the GS procedures always terminates at a local maximum, or at worst, contains a subsequence which converges to a local maximum of the similarity measure function. The framework is verified by theoretical analysis and experimental results of several typical GS-type algorithms.

1. Introduction

The Robust Graph mode seeking by Graph Shift (RGGS) algorithm [1–7] is a newly proposed algorithm in seeking dense subgraphs (also known as graph mode) and has received much attention in the machine learning and data mining areas. Due to its tremendous advantages in removing the noisy data in learning dense subgraphs, it is popularly used for image processing such as common pattern matching [7,2], computer vision such as object tracking [3–6], and cluster analysis [1]. In addition, its low computational and memory complexity requirements make it feasible for handling real applications, especially for large-scale data. However, little work has been done to build the theoretical foundation for the RGGS algorithm. This leaves important questions about why such algorithms work and whether there is any underpinning foundation to ensure the empirical demonstrations work well for any situations.

The RGGS algorithm originated from the Dominant Sets and Pairwise Clustering (DSPC) algorithm [8,9], which treats the dense subgraphs discovery problem as a constrained optimization problem and gives a clear definition of the so-called “dominant set”, i.e., dense subgraphs. Further, by modifying the existing DSPC (known as Replicator Dynamics in the RGGS algorithm) procedure, the RGGS algorithm adds a “Neighborhood Expansion” procedure to reinforce the learning results. By iteratively employing the DSPC procedure and the newly added Neighborhood Expansion procedure, the RGGS algorithm claims to find a local maximum of the constraint objective function after a finite number of iterations and further, empirically demonstrates the claim.

However, to the best of our knowledge, none of the existing arguments were built on a solid theoretical foundation with a proper justification of the objective functions' behavior during the procedures and the stopping criteria. All of the above issues are closely related to one thing: the convergence property of the RGGS algorithm. That is to say, we need to ensure that the generated sequence set is convergent or at least contains a convergent subsequence set. It is certainly crucial to have a theoretical analysis of the convergence behavior of the RGGS algorithm before we can confidently utilize it.

Building a proper convergence theorem for the respective learning algorithms is a very important theoretical issue in building solid learning theories. While this is often very challenging, several of existing learning theories include such a component, including the convergence theorem for algorithms with an iterative sequence set. The convergence proof for the fuzzy *c*-means algorithm (FCM) was provided by Bedzek [10,11], who employed Zangwill's theory [12,13] to establish the sequence's convergence property. Hoppner and Klawonn [14] proved the convergence of the axis-parallel variant of Gustafson–Kessel's algorithm [15] by applying Banach's classical contraction principle [16], which is the general case of FCM. Groll and Jakel [17] used the equivalence between the original and reduced FCM criteria, and conducted a new and more direct derivation of the convergence properties of FCM algorithms. In addition, Selim and Ismail [18] treated the k-means clustering problem as a nonconvex mathematical program and provided a rigorous proof of the finite convergence of the K-means-type algorithms.

* Corresponding author.
E-mail addresses: xhfan.ml@gmail.com (X. Fan), longbing.cao@uts.edu.au (L. Cao).

http://dx.doi.org/10.1016/j.patcog.2015.02.013
0031-3203/

The FCM's convergence theory seems complete and this inspires the possibility of it being applied to other purposes, e.g. the RGGS algorithm with an iterative set. Unfortunately, it is not easy to capture the RGGS algorithm's complex characteristics in the convergence proof. To address this problem, we provide a theoretical analysis of the RGGS algorithm. We start with an understanding of the principal characteristics of the RGGS algorithm by breaking it down into three key components, the generated sequence set, the objective function and mapping, and then propose a framework to map such components to the conditions required in Zangwill's theory. We find that the mapped RGGS algorithm can then perfectly match with the key requirements in Zangwill's theory. The convergence theorem for the RGGS algorithm is then developed.

Further, a definition of the so-called "GS-type algorithms" is then given to provide us with a general framework to fit algorithms with similar properties. More importantly, we build up a systematic learning process for them by analyzing the objective functions' behaviors and observing how they arrive at the final results. We illustrate the proposed convergence theorem in terms of proving two typical GS-type algorithms: the RGGS algorithm in [1] and the DSPC algorithm in [9]. The theoretical analysis is then verified against the experimental results.

In summary, this work makes the following substantial contributions:

- A theoretical framework is proposed to analyze the convergence behaviors of the RGGS algorithm. This enables the intrinsic key characteristics embedded in the RGGS algorithm to be effectively captured.
- Taking the two typical GS-type algorithms as examples, we prove the RGGS algorithm and the DSPC algorithm either terminate at a local maximum value or at least contain a subsequence which converges to a local maximum.
- A convergence proof framework is built to generalize the proposed theoretical framework to other GS-type algorithms, namely the RGGS algorithm with similar properties.

The paper is organized as follows. Section 2 introduces the principle of the RGGS algorithm. In Section 3, we first introduce Zangwill's theory, and then propose a framework for extracting three key components in the RGGS algorithm, which are then mapped to the properties of Zangwill's theory. Section 4 discusses the convergence and features of the RGGS algorithm. We extend the convergence proof to other GS-type algorithms and build up a generic convergence proof framework in Section 5. Experiments are conducted in Section 6 to verify the convergence theorem and behaviors. Conclusions and future work can be found in Section 7.

2. Preliminaries

2.1. Rationale of the RGGS algorithm

The basic principle of the RGGS algorithm is set forth in [1]. From the perspective of graph mining, the RGGS algorithm searches each vertex's dense "nearer" subgraphs with strong internal closeness. Two procedures: Replicator Dynamics and Neighborhood Expansion are recursively employed on each vertex sequentially to reach the goal. The former largely shrinks the identified subgraphs and the latter expands the existing subgraphs, both shifting towards a local graph mode.

In [1,9], a probabilistic coordinate on Graph G is defined as a mapping: $V \to \Delta^n$, where $\Delta^n = \{\boldsymbol{x} \in R^n : \boldsymbol{x}_i \geq 0, i \in \{1,\ldots,n\}$ and $|\boldsymbol{x}|_1 = 1\}$. The support of $\boldsymbol{x} \in \Delta^n$ is the indices of all non-zero components, denoted as $\delta(\boldsymbol{x}) = \{i | \boldsymbol{x}_i \neq 0\}$, corresponding to subgraphs $G_{\delta(\boldsymbol{x})}$, and $\boldsymbol{x}_i$ denotes vertex i's occurrence in the subgraphs $G_{\delta(\boldsymbol{x})}$ to some extent.

The algorithm operates on an affinity matrix $A = (a_{ij})^{n\times n}$, in which a_{ij} measures the similarity between vertices i and j. Then, the internal similarity of subgraphs $G_{\delta(\boldsymbol{x})}$ is measured as

$$g(\boldsymbol{x}) := a(\boldsymbol{x},\boldsymbol{x}) = \sum_{i,j=1}^{n} a_{ij}\boldsymbol{x}_i\boldsymbol{x}_j = \boldsymbol{x}^T A\boldsymbol{x}. \tag{1}$$

Accordingly, a local maximum solver of $g(\boldsymbol{x})$ can be taken to represent the desired dense subgraphs. The identification of such local maximum regions is equivalent to solving the following quadratic optimization problem:

$$\begin{cases} \text{maximize} & g(\boldsymbol{x}) = \boldsymbol{x}^T A\boldsymbol{x} \\ \text{subject to} & \boldsymbol{x} \in \Delta^n \end{cases} \tag{2}$$

2.2. Mapping definitions

We state here the problem of the RGGS algorithm and propose the corresponding mapping to form a conceptual understanding of the RGGS algorithm.

In general, the RGGS algorithm defines a mapping $T_m : \Delta^n \to \Delta^n$ to obtain the iterative sequence set as

$$\boldsymbol{x}^{(k)} = T_m(\boldsymbol{x}^{(k-1)}) = \cdots = (T_m)^{(k)}(\boldsymbol{x}^{(0)}); \quad k = 1,2,\cdots. \tag{3}$$

where $\boldsymbol{x}^{(0)}$ is an initial starting point, and superscripts in parentheses correspond to the iteration number. Subsequently, the convergence of the RGGS algorithm is to test whether the iterative sequence set $\{\boldsymbol{x}^{(k)}\}_{k=1}^{\infty}$ generated by T_m converges to a local maximum solver of the problem in Eq. (2).

The mapping function T_m plays a crucial role in the RGGS algorithm, combining the Replicator Dynamics procedure (B^{m_k}) and the Neighborhood Expansion procedure (C). Therefore, in order to understand and analyze the convergence problem, it is essential to have a deep understanding of the mapping function. Accordingly, T_m is broken down as

$$T_m := B^{m_k} \circ C. \tag{4}$$

In Eq. (4), B^{m_k} represents the kth ($k \leq m$) Replicator Dynamics procedure; m_k corresponds to the number of transformation B in the kth Replicator Dynamics procedure when a subgraphs's mode is reached in this procedure (this result is actually a special case of Theorem 3). B is the transformation expressed as

$$\begin{aligned} B : \Delta^n \to \Delta^n, \boldsymbol{x}(l_k) \to \boldsymbol{x}(l_k+1) \\ = \left(\frac{\omega_1(l_k)\boldsymbol{x}_1(l_k)}{\sum_{i=1}^n \omega_i(l_k)\boldsymbol{x}_i(l_k)}, \ldots, \frac{\omega_n(l_k)\boldsymbol{x}_n(l_k)}{\sum_{i=1}^n \omega_i(l_k)\boldsymbol{x}_i(l_k)}\right). \end{aligned} \tag{5}$$

In Eq. (5), $\omega_i(l_k) = (A\boldsymbol{x}(l_k))_i = \sum_{j=1}^n a_{ij}\boldsymbol{x}_j(l_k)$, $i \in \{1,\ldots,n\}$, $l_k \in \{1,\ldots,m_k-1\}$.

C is the Neighborhood Expansion procedure, denoted as

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \Delta\boldsymbol{x} = \boldsymbol{x}^{(k)} + t^*\boldsymbol{b}. \tag{6}$$

Details of t^* and $\boldsymbol{b}$ are explained in Appendix A.

2.3. Detailed procedure and stopping criteria

The RGGS algorithm in [1] is an iterative process for seeking dense subgraphs which starts from each vertex in the graph. The pseudo-codes below illustrate the whole process.

Require $A^{n\times n}$, the affinity matrix of the whole data set with the diagonal value 0;
$\{\boldsymbol{x} = \{\boldsymbol{x}_i\}_{i=1^n}\}$, initial starting points, usually taken as $\{\boldsymbol{e} = \{\boldsymbol{e}_i\}_{i=1^n}\}$
1: **for** i=1,…n **do**
2: do Replicator Dynamics (Eq. (5)) of $\boldsymbol{x}_i$