Parallel Computing 47 (2015) 70–87

Contents lists available at ScienceDirect

Parallel Computing

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

Spectral clustering for divide-and-conquer graph matching

Vince Lyzinski [a,*], Daniel L. Sussman [d], Donniell E. Fishkind [b], Henry Pao [b], Li Chen [b], Joshua T. Vogelstein [c], Youngser Park [b], Carey E. Priebe [b]

[a] Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore, MD 21211, USA
[b] Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD 21218, USA
[c] Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA
[d] Department of Statistics, Harvard University, Cambridge, MA 02138, USA

ARTICLE INFO

Article history:
Available online 12 March 2015

Keywords:
Graph matching
Adjacency spectral embedding
Clustering
Stochastic block model

ABSTRACT

We present a parallelized bijective graph matching algorithm that leverages seeds and is designed to match very large graphs. Our algorithm combines spectral graph embedding with existing state-of-the-art seeded graph matching procedures. We justify our approach by proving that modestly correlated, large stochastic block model random graphs are correctly matched utilizing very few seeds through our divide-and-conquer procedure. We also demonstrate the effectiveness of our approach in matching very large graphs in simulated and real data examples, showing up to a factor of 8 improvement in runtime with minimal sacrifice in accuracy.

1. Introduction

Graph matching is an increasingly important problem in inferential graph statistics, with applications across a broad spectrum of fields including computer vision [38,10], shape matching and object recognition [4,7], and biology and neuroscience [22,34,36], to name a few. The *graph matching problem* (GMP) seeks to find an alignment between the vertex sets of two graphs that best preserves common structure across graphs. Unfortunately, the GMP is inherently combinatorial, and no efficient exact graph matching algorithms are known. Indeed, even the simpler problem of determining if two graphs are isomorphic is famously of unknown complexity [19,30], and if the graphs are allowed to be loopy, weighted and directed, then the simplest version of GMP is equivalent to the NP-hard quadratic assignment problem. Due to its wide applicability, there exist a vast number of approximating algorithms for GMP; see the paper "30 Years of Graph Matching in Pattern Recognition" [11] for an excellent survey of the existing literature.

When matching across graphs, often we have access to a partial matching of the vertices in the form of a *seeding*. In practice, the assumption of seeds is quite natural in many applications. For example, in aligning social networks actors' user names may often allow for a partial alignment to be known a priori. When matching across brain graphs (connectomes), we have geometric information provided by the brain atlas which provides a soft seeding of the vertices. In many time series graphs, it is common to have a group of invariant vertices across time which act as seeds.

In the *seeded graph matching problem* (SGMP), we leverage the information contained in an available partial matching to match the remaining vertices across graphs. Though the literature on seeded graph matching is comparatively small,

* Corresponding author.
E-mail addresses: vlyzins1@jhu.edu (V. Lyzinski), daniellsussman@fas.harvard.edu (D.L. Sussman), def@jhu.edu (D.E. Fishkind), hen.pow@gmail.com (H. Pao), lchen87@jhu.edu (L. Chen), jovo@jhu.edu (J.T. Vogelstein), youngser@jhu.edu (Y. Park), cep@jhu.edu (C.E. Priebe).

http://dx.doi.org/10.1016/j.parco.2015.03.004
0167-8191/

recent results point to significant performance improvements in GM algorithms by incorporating even a modest number of seeds [16,27].

Though a myriad of approximate graph matching algorithms exist, the very large graphs arising in the burgeoning realm of "big data" demand highly scalable algorithms. Roughly speaking, existing state of the art algorithms for approximate graph matching can be divided into two classes: those that seek to bijectively match vertices of graphs of the same order, and those that seek matchings between the vertex sets that are allowed to be many-to-many and many-to-one. The current cutting-edge bijective graph matching algorithms achieve excellent performance in approximately matching graphs with thousands of vertices and with computational complexity $O(n^3)$–n the number of vertices being matched; see for example [34,37,15]. These algorithms often operate directly on the adjacency matrices of the graphs to be matched, utilizing the tools of nonlinear optimization to approximately solve GMP directly. However, owing to their $O(n^3)$ complexity, these algorithms are practically unusable, without significant computation resources, for matching very large graphs ($n \approx 10^5$).

Scalability is often achieved via relaxing the bijection requirement and allowing many-to-many and many-to-one matchings. These graph matching procedures can efficiently match very large graphs, often with n in the tens of thousands; see for example [26,1]. A common approach to these scalable inexact algorithms is that they first match smaller, lower dimensional representative objects (prototype graphs in [1], eigenvectors in [26]) and use these to build the overall matching.

Herein, we propose a new divide-and-conquer approach to *scalable bijective* seeded graph matching. Our algorithm, the Large Seeded Graph Matching algorithm (LSGM, see Algorithm 1), merges the approaches of bijective and non-bijective graph matching and leverages the information in seeded vertices in order to match very large graphs. The algorithm proceeds in two steps: We first spectrally embed the graphs—yielding a low dimensional Euclidean representation of our graph—and then use the information provided by seeded vertices to jointly cluster the vertices of the two embedded graphs. This embedding procedure allows us to employ the powerful theory of adjacency spectral embedding (see for example [31,17]) to prove asymptotically perfect performance in *jointly* clustering stochastic block model random graphs, see Theorem 4.1 for detail.

Once the vertices are jointly clustered, we then match the graphs within the clusters. This matching step is fully parallelizable and flexible in that we can employ any one of a number of matching procedures depending on the properties of the resulting clusters. The flexibility afforded by our procedure in the clustering and matching subroutines can have a dramatic impact on algorithmic scalability. For example, on a 1600 vertex simulated graph our parallelization procedure was able to achieve an factor of 8 improvement in speed at minimal accuracy degradation by increasing the number of clusters and hence the number of cores that were used; see Section 5.2.

Though we are not the first to employ a divide-and-conquer approach to graph matching (see for example [9,38,1]), our focus on the efficient utilization of apriori observed seeded vertices and the theoretical framework for our approach provided by Theorem 4.1 set our algorithm apart from the existing literature.

Note: All graphs considered herein will be simple; in particular there are no multiple edges between two vertices nor are there edges with a single vertex as both endpoints. Modifications for the directed case are quite simple [31,17] but we do not consider them in this manuscript. All vectors considered will be column vectors, and $\vec{1}_m$ is the length-m vector of all 1*s*. When appropriate we drop the subscript and just write $\vec{1}$. Throughout the paper we employ the standard notation $[n] := \{1, 2, \ldots, n\}$ for any $n \in \mathbb{N}$, and to simplify future notation, if $A \in \mathbb{R}^{n\times n}$ and $\tau, \sigma \subset [n]$, then $A(\tau, \sigma)$ will denote the submatrix of A with row indices τ and column indices σ. For a matrix $X, X(:, i)$ will denote the ith column of X and $X(i, :)$ the ith row of X. Also for two matrices X and Y, $[X|Y]$ will denote the column concatenation of X and Y.

Algorithm 1. Divide-and-conquer seeded graph matching; the LSGM algorithm

INPUT: Symmetric, hollow $A, B \in \{0,1\}^{n\times n}, s \in [n]$, seeding $\phi : [s] \to [s]$
OUTPUT: A matching of G_1 and G_2 given by ψ;
Step 1: Embed and jointly cluster the graphs according to Algorithm 2
Step 2: In parallel
for $i = 1$ to k **do**
  Match cluster i across the graphs using, yielding matching $\psi^{(i)}$;
end for
OUTPUT: $\psi = \oplus_{i=1}^{k} \psi^{(i)}$.

2. Background

There are numerous formulations of the graph matching problem, though they all share the same objective heuristic: given two graphs, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, GMP seeks an alignment between the vertex sets V_1 and V_2 that best preserves structure across the graphs. In *bijective* graph matching, we further assume $|V_1| = |V_2| = n$, and the alignment sought by GMP is a bijection between V_1 and V_2. In *non-bijective* graph matching, we allow for $|V_1| \neq |V_2|$ and for alignments that are not one-to-one.