Available online at www.sciencedirect.com

SciVerse ScienceDirect

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

Survey

Data mining of social networks represented as graphs

*David F. Nettleton**

Universitat Pompeu Fabra, Barcelona, Spain
IIIA-CSIC, Bellaterra, Spain

ARTICLE INFO

Article history:
Received 18 October 2012
Received in revised form
20 December 2012
Accepted 20 December 2012

Keywords:
Graphs
Online social networks
Graph mining
Data mining
Statistical analysis
Data modelling

ABSTRACT

In this survey we review the literature and concepts of the data mining of social networks, with special emphasis on their representation as a graph structure. The survey is divided into two principal parts: first we conduct a survey of the literature which forms the 'basis' and background for the field; second we define a set of 'hot topics' which are currently in vogue in congresses and the literature. The 'basis' or background part is divided into four major themes: graph theory, social networks, online social networks and graph mining. The graph mining theme is organized into ten subthemes. The second, 'hot topic' part, is divided into five major themes: communities, influence and recommendation, models metrics and dynamics, behaviour and relationships, and information diffusion.

Contents

1. Introduction 2
2. Base survey 2
   2.1. Graphs 2
   2.2. Social networks 4
   2.3. Social networks and computers 5
      2.3.1. On-line social network applications 5
      2.3.2. The analysis of social network datasets 5
      2.3.3. Applications and software for social network analysis 6
   2.4. Graph mining 6
      2.4.1. Preamble 6
      2.4.2. Prediction/supervised learning 6
      2.4.3. Efficiency 6
      2.4.4. Pattern detection 7
      2.4.5. Measurement & metrics 7
      2.4.6. Modelling, evolution and structure 9
      2.4.7. Data processing 13

* *Correspondence to:* Universitat Pompeu Fabra, Barcelona, Spain. Tel.: +34 93 542 25 00.
E-mail address: david.nettleton@upf.edu.

1574-0137/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
doi:10.1016/j.cosrev.2012.12.001

International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-1, Issue-3, July 2011

Survey of Finding Frequent Patterns in Graph Mining: Algorithms and Techniques

Vijender Singh, Deepak Garg

***Abstract*— Graphs become increasingly important in modeling complicated structures, such as circuits, images, chemical compounds, protein structures, biological networks, social networks, the web, workflows, and XML documents. Many graph search algorithms have been developed in chemical informatics, computer vision, video indexing and text retrieval with the increasing demand on the analysis of large amounts of structured data; graph mining has become an active and important theme in data mining.**

Index Termes: **Subgraphs, Graph Mining, gSpan**

I. INTRODUCTION

In mathematics, computer science and related subjects an algorithm is an effective method for solving a problem expressed as a finite sequence of instructions. Algorithms are used for calculation data processing and many other fields.

Meaning1. An algorithm operating on data that represent continuous quantities, even enough this data is represented by discrete approximation such algorithm are studied in numerical analysis.

Meaning2. An algorithm in the form of different equations that operates continuous on the data running an analog computer.

A. Apriori-Based Approach

Apriori based frequent substructure mining algorithm share similar characteristics with Apriori-based frequent item set mining algorithms. The search for frequent groups starts with graphs a small "size" and proceeds in a bottom-up manner by generating candidate having an extra vertex, edge or path. The definition of graph size depends on algorithm used.

B. Pattern-Growth Approach

The Apriori-based approach has to use the breadth-first search (BFS) strategy because of its level-wise candidate generation.

II. SURVEY OF TECHNIQUES AND ALGORITHMS

Various algorithms on graph mining were developed by many researchers. Some of them are reviewed in this section. Ullmann [1] in 1976 developed an algorithm for subgraph isomorphism. Subgraph isomorphism determined by means of a brute-force tree search procedure. This algorithm attains efficiency by inferentially eliminating successor's nodes in the tree search. Agarwal and Srikant [2] in 1994 considered the problem of discovering association rules between items in a large database of sales transaction. They presented two new algorithms for solving this problem that are fundamentally different from the known algorithm. Cook and Holder [14] in 1994 discovered a new version of their SUBDUE substructure discovery system is based on minimum description length principle. Holder, Cook and Djoko [3] in 1994 described the SUBDUE system which the minimum description length (MDL) principle is discovered substructures that compress the database and represent structural concepts in the data. In this paper they described the application of SUBDUE and also discussed the minimum description length principle and background knowledge used by SUBDUE can guide substructure discovery in a variety of domain. Blockeel and Raedt [6] in 1998 introduced a first-order framework for top-down induction of logical decision tree. Top-down induction of decision trees is the best known and most successful machine learning technique. It has been used solve numerous practical problems. It employs a divide-and conquers strategy, and in this it differs from its rule-based competitors which are based on covering strategies. Chakrabarti, Dom and Indyk [7] in 1998 developed a new method for automatically classifying hypertext into a given topic hierarchy, using an iterative relaxation algorithm. After bootstrapping off a text-based classifier, they used both local texts in a document as well as the distribution of the estimated classes of other documents in its neighborhood, to refine the class distribution of document being classified. They discussed three area of research: text and hypertext information retrieval, machine learning in context other text or hypertext, and computer vision and pattern recognition.

Inokuchi,Washio and Motoda [9] in 1998 proposed a novel approach name AGM to efficiently mine the association rule among the frequently appearing substructure in a given graph dataset. A graph is represented by adjacency matrices and the frequent patterns appearing in the matrices are mined through the extended algorithm of the basket analysis. Calders and Wisen [10] in 2001 Presented on monotone data mining layer a simple data-mining logic (DML) that can express common data mining tasks like "find Boolean association rules" or "Find inclusion dependencies". Kramer, Raedt, and Helma [11] in 2001 presented the application of feature mining techniques to the developmental therapeutics program's AIDS antiviral screen database. Kuramochi and Karypis [12] in 2001 presented a computationally efficient algorithm for finding all frequent subgraphs in large graph databases. They evaluated the performance of the algorithm by experiments with synthetic datasets as

Manuscript received June 4, 2011.

Vijender Singh, Department of Computer Science and Engineering, Thapar University, Patiala (Punjab), India, Mobile No. +91-9255074702, (e-mail: vijender_bhar@hotmail.com).

Dr. Deepak Garg, Professor, Department of Computer Science and Engineering, Thapar University, Patiala (Punjab), India (e-mail: deep108@yahoo.com).

IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013
ISSN: 2320 - 8791
www.ijreat.org

Fast Sequential Pattern Mining With UpDown Directed Acyclic Graph

Iswarya. T[1] and Sivakumar. E[2]

[1,2] **Computer Science and Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, Tamil Nadu, India**

Abstract

Sequential Pattern Mining is a technique used to discover frequent patterns in a sequence database. An UpDown Directed Acyclic Graph (UDDAG) is a data structure used for sequential pattern mining which derives patterns based on the bidirectional pattern growth approach. In existing, to derive a pattern, a database projection method was used. This method projects the database according to their prefix and suffix recursively and makes the database much smaller to further levels. UDDAG derives length-(k+1) patterns based on the projected databases of length-k patterns recursively. However this method takes time to build the database and to find the support count. To overcome the problem, an approach to represent the item sets in database as vertical bitmap representation is proposed. It efficiently stores the database as vertical bitmaps, where each bitmap represents an item set in the database. Efficiently counting the support of the item set is one of the main advantages of the vertical bitmap representation of the data.

Keywords: *Sequential Pattern Mining, Database Projection, Vertical Bitmaps, UDDAG.*

1. Introduction

Data Mining is the process which helps to extracting interesting and hidden information or patterns from large information repositories such as relational database, data warehouses, XML repository, etc. Sequential Pattern Mining is an important problem in data mining.

Sequential Pattern Mining is the process of extracting certain sequential patterns from the sequence database whose support exceeds a predefined minimal support threshold. A minimum support is defined by users because the number of sequences can be very large, and users have different interests and requirements to get the most interesting sequential patterns. By using the minimum support we can prune out those sequential patterns, consequently making the mining process more efficient. Sequential pattern can be widely used in different areas.

For example, from a sequence database, we can find the frequent sequential purchasing patterns, for example if a customer buys the computer typically the same customer will buy the pen drive within few weeks. 2. Basic Concepts

Let I = { $x_1,\ldots,x_n$ } be a set of items. A non-empty subset of items is referred an item set. The number of items in an item set is called the length of an item set. A list of item sets, $\alpha = < X_1,\ldots,X_l >$ is called sequence. An item set X_i ($1 \leq i \leq l$) in a sequence is called a transaction. The number of transactions in a sequence is called the length of the sequence. A sequence $\alpha = < X_1,\ldots,X_n >$ is called a subsequence of another sequence $\beta = < Y_1,\ldots,Y_m >$ where ($n \leq m$), and β a super-sequence of α, if there exist integers $1 \leq i_1 <\ldots< i_n \leq m$ such that $X_1 Y_{i1},\ldots, X_n Y_{in}$. A sequence database is a set of 2-tuples (sid, α), where sid is a sequence-id and α is a sequence.

2.1 Sequential Pattern

Sequential Pattern is a sequence of item sets that frequently occurs in a specific order, all items in the same item sets are supposed to have the same transaction time. Consider the database D is sorted, with customer-id as major key and transaction-time as minor key. Usually all the transactions of a customer are together viewed as a sequence, usually called customer-sequence as shown in Table 1.

Table 1: Customer sequence database

Seq.id	***Sequence***
s1	<(1,2)(4)(3,5)(7)>
s2	<(1,2,3)(3)(5,7)>
s3	<(6)(5,6)(4,7)>
s4	<(4,5)(5,6)(1)(3,5)>
s5	<(4)(5)(2,3)(1,6)>

2.2 Support

A customer supports a sequence s if s is contained in their corresponding sequences in the sequence database; the

International Journal of Computer Science & Information Technology (IJCSIT) Vol 4, No 1, Feb 2012

GRAPH BASED NEW APPROACH FOR FREQUENT PATTERN MINING

Anurag Choubey[1] , Dr. Ravindra Patel[2] and Dr. J.L. Rana[3]

[1]Dean Academic, Technocrats Institute of Technology, Bhopal, (M.P.), India
E-mail: anuragphd11@gmail.com
[2]Reader & Head, Department of Computer Application, UIT-RGPV, Bhopal, (M.P.), India
E-mail: ravindra@rgtu.net
[3]Former Professor & Head, Department of Computer Science & Engineering, MANIT, Bhopal, (M.P.), India
E-mail: jl_rana@yahoo.com

ABSTRACT
Association rule mining is a function of data mining research domain and frequent pattern mining is an essential part of it. Most of the previous studies on mining frequent patterns based on an Apriori approach, which required more number of database scans and operations for counting pattern supports in the database. Since the size of each set of transaction may be massive that it makes difficult to perform traditional data mining tasks. This research intends to propose a graph structure that captures only those itemsets that needs to define a sufficiently immense dataset into a submatrix representing important weights and does not give any chance to outliers. We have devised a strategy that covers significant facts of data by drilling down the large data into a succinct form of an Adjacency Matrix at different stages of mining process. The graph structure is so designed that it can be easily maintained and the trade off in compressing the large data values is reduced. Experimental results show the effectiveness of our graph based approach.

KEYWORDS
Data mining, Frequent pattern, Graph structure, Adjacency Matrix

1. INTRODUCTION

Data mining is the non-trivial mining of implicit and structured data, previously unknown prototype that is potentially useful especially decision making systems. It is one such area which continues to pave its way from bioinformatics [1] to web-based mining[2], from image appreciation[3] to wireless remote sensing[4]. Be it any giant organization or any small private run firm, they all need to figure out associative relationships for improving marketing strategies and informed business decisions.

Along with major domains like sequential patterns, classifications, clustering etc, Association rule discovery has been an active area of examination. Frequent pattern mining was first introduced by Agrawal et al. [5] The methods for efficient exploration is based on the candidate generation-and-test approach like Apriori [6,7] and pattern growth strategy during the series of decades [8,9,10,11,12] have already been developed but they have a huge setback that generates n number of candidate sets and takes as many database scans in large datasets for frequent itemsets discovery. Also the performance gets degraded when the size of database is exponential. To improve efficiency of mining process, Han et al [13,14,15] proposed an alternative frame work, namely a tree based frame work. The algorithm F. P. growth, they

DOI : 10.5121/ijcsit.2012.4117


Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 15, no. 6, pp. 753–776 (2011)

FP-GraphMiner – A Fast Frequent Pattern Mining Algorithm for Network Graphs

R. Vijayalakshmi[1] *R. Nadarajan*[1] *John F. Roddick*[2] *M. Thilaga*[1] *P. Nirmala*[1]

[1]Department of Mathematics and Computer Applications,
PSG College of Technology,
Coimbatore 641004, Tamil Nadu, India
[2]School of Computer Science, Engineering and Mathematics,
Flinders University,
PO Box 2100, Adelaide, SA 5001, South Australia.

Abstract

In recent years, graph representations have been used extensively for modelling complicated structural information, such as circuits, images, molecular structures, biological networks, weblogs, XML documents and so on. As a result, frequent subgraph mining has become an important subfield of graph mining. This paper presents a novel Frequent Pattern Graph Mining algorithm, FP-GraphMiner, that compactly represents a set of network graphs as a Frequent Pattern Graph (or FP-Graph). This graph can be used to efficiently mine frequent subgraphs including maximal frequent subgraphs and maximum common subgraphs. The algorithm is space and time efficient requiring just one scan of the graph database for the construction of the FP-Graph, and the search space is significantly reduced by clustering the subgraphs based on their frequency of occurrence. A series of experiments performed on sparse, dense and complete graph data sets and a comparison with *MARGIN*, *gSpan* and *FSMA* using real time network data sets confirm the efficiency of the proposed FP-GraphMiner algorithm.

Keywords: frequent pattern mining, frequent subgraph, graph database, graph mining, maximal frequent subgraph, maximum common subgraph.

Submitted:	Reviewed:	Revised:	Reviewed:
February 2011	May 2011	May 2011	August 2011
Revised:	Accepted:	Final:	Published:
August 2011	October 2011	October 2011	November 2011
Article type:		Communicated by:	
Regular paper		G. Liotta	

E-mail addresses: rv@mca.psgtech.ac.in (R. Vijayalakshmi) rn@mca.psgtech.ac.in (R. Nadarajan) john.roddick@flinders.edu.au (John F. Roddick) mta@mca.psgtech.ac.in (M. Thilaga) pna@mca.psgtech.ac.in (P. Nirmala)

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 3, March 2013

Algorithms for Frequent Subgraph Mining

Sujata J. Suryawanshi[1], Prof. Mrs. S. M. Kamalapur[2]

Computer Engg. Department, KKWECOE, Nashik[1]

Computer Engg. Department, KKWECOE, Nashik[2]

Abstract*:* Due to increasing number of complex objects, Data mining algorithms are facing the challenges. To model such complex object, Graph a natural data structure is used. Graph mining is an important research area within the domain of data mining. A graph is a general model to represent data and has been used in many domains like cheminformatics and bioinformatics. Mining patterns from graph data base is difficult task than mining pattern from data set, sequences or tree, because graph related operations, such as subgraph testing, generally have higher time complexity. This paper gives the comparative study of frequent subgraph mining algorithms. In this paper different issues are discussed like graph representation, searching strategy, and Graph Summarization.

Keywords: Data mining, graph mining, pattern summarization.

I. INTRODUCTION

Data mining is the procedure of extracting knowledge from raw data. In recent years, there has been an increased interest in developing data mining algorithms that operate on graphs. Data can also be represented by various means. Structured data and semi-structured data are naturally suited to graph representations. Graph mining is an important research area within the domain of data mining. Most important concept in Graph mining is to find frequent subgraph from graph database.

Frequent subgraph mining is the processor of extracting all frequent subgraphs from graph dataset who have occurrence count greater than or equal to the specified threshold. Following figure gives example of frequent subgraph mining, where figure 1(a) and 1(b) represents any two graphs and figure 1(c) gives frequent subgraph which is present in both graphs.

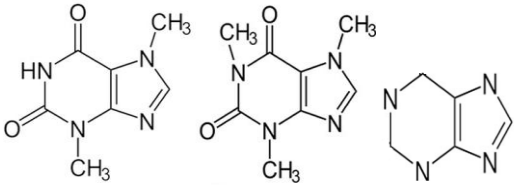

Fig. 1(a) Theobromine, 1(b) Caffeine and 1(c) Frequent subgraph.

The rest of the paper is structured as follows: Section II describes frequent subgraph mining algorithms, advantages and disadvantages of each. Also describes algorithms for summarizing graph patterns. Finally, we draw conclusion in Section III.

II. FREQUENT SUBGRAPH MINING ALGORITHMS

There are several efficient frequent sub graph mining algorithms have been proposed. Efficient frequent sub graph mining algorithm is an algorithm which gives small number of graphs as a result from large graph database. There is an algorithm which gives approximate graph mining based on Spanning tree [15]. The frequent sub graph mining algorithms comes under two different types :

A. Algorithms using BFS Search Strategy

These types of algorithms come under Apriori based approach. In these approaches before mining any of the sub graphs of size k+1, mining of sub graphs with size k needs to be completed, where the size of the graph is defined by the number of its vertices. In this method, since the candidates are generated in level-wise manner, the BFS search strategy must be used. To generate the biggest candidate for frequent sub graphs, frequent sub graphs of size k are merged together to generate a graph of size k+1. Major algorithms with this approach are:

1. *AGM Algorithm:* AGM [2] is the Apriori based graph mining algorithm. This algorithm uses adjacency matrix for graph representation. Searching technique used in this algorithm is BFS Strategy. BFS mines all isomorphic subgraphs, whereas DFS does not do so and therefore, DFS consumes less memory. Largest graph of chemical compound discoved by AGM algorithm have the size of 13 atoms. AGM algorithm can efficiently mine frequently induced subgraphs in given graph dataset. Though it mines frequent subgraphs, still it generates multiple candidates due to use of BFS strategy.

2. *FSG Algorithm:* FSG[3] is also an Apriori based graph mining algorithm. In this algorithm, edges in graph considered as frequent items in traditional itemset. Hence

Physics Reports 486 (2010) 75–174

Contents lists available at ScienceDirect

Physics Reports

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

Community detection in graphs

Santo Fortunato *

Complex Networks and Systems Lagrange Laboratory, ISI Foundation, Viale S. Severo 65, 10133, Torino, I, Italy

ARTICLE INFO

Article history:
Accepted 5 November 2009
Available online 4 December 2009
editor: I. Procaccia

Keywords:
Graphs
Clusters
Statistical physics

ABSTRACT

The modern science of networks has brought significant advances to our understanding of complex systems. One of the most relevant features of graphs representing real systems is community structure, or clustering, i.e. the organization of vertices in clusters, with many edges joining vertices of the same cluster and comparatively few edges joining vertices of different clusters. Such clusters, or communities, can be considered as fairly independent compartments of a graph, playing a similar role like, e.g., the tissues or the organs in the human body. Detecting communities is of great importance in sociology, biology and computer science, disciplines where systems are often represented as graphs. This problem is very hard and not yet satisfactorily solved, despite the huge effort of a large interdisciplinary community of scientists working on it over the past few years. We will attempt a thorough exposition of the topic, from the definition of the main elements of the problem, to the presentation of most methods developed, with a special focus on techniques designed by statistical physicists, from the discussion of crucial issues like the significance of clustering and how methods should be tested and compared against each other, to the description of applications to real networks.

Contents

1. Introduction 76
2. Communities in real-world networks 78
3. Elements of community detection 82
   3.1. Computational complexity 83
   3.2. Communities 83
      3.2.1. Basics 83
      3.2.2. Local definitions 84
      3.2.3. Global definitions 85
      3.2.4. Definitions based on vertex similarity 86
   3.3. Partitions 87
      3.3.1. Basics 87
      3.3.2. Quality functions: Modularity 88
4. Traditional methods 90
   4.1. Graph partitioning 90
   4.2. Hierarchical clustering 93
   4.3. Partitional clustering 93
   4.4. Spectral clustering 94
5. Divisive algorithms 96
   5.1. The algorithm of Girvan and Newman 97
   5.2. Other methods 99

* Tel.: +39 011 6603090; fax: +39 011 6600049.
E-mail address: fortunato@isi.it.

0370-1573/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.physrep.2009.11.002

Available online at www.sciencedirect.com
ScienceDirect
Artificial Intelligence 171 (2007) 939–950
Artificial Intelligence
www.elsevier.com/locate/artint

Introducing possibilistic logic in ILP for dealing with exceptions

Mathieu Serrurier *, Henri Prade

UPS, IRIT 118 route de Narbonne, Toulouse, France

Received 10 August 2005; received in revised form 27 April 2007; accepted 30 April 2007

Available online 22 May 2007

Abstract

In this paper we propose a new formalization of the inductive logic programming (ILP) problem for a better handling of exceptions. It is now encoded in first-order possibilistic logic. This allows us to handle exceptions by means of prioritized rules, thus taking lessons from non-monotonic reasoning. Indeed, in classical first-order logic, the exceptions of the rules that constitute a hypothesis accumulate and classifying an example in two different classes, even if one is the right one, is not correct. The possibilistic formalization provides a sound encoding of non-monotonic reasoning that copes with rules with exceptions and prevents an example to be classified in more than one class. The benefits of our approach with respect to the use of first-order decision lists are pointed out. The possibilistic logic view of ILP problem leads to an optimization problem at the algorithmic level. An algorithm based on simulated annealing that in one turn computes the set of rules together with their priority levels is proposed. The reported experiments show that the algorithm is competitive to standard ILP approaches on benchmark examples.

Keywords: Inductive logic programming; Non-monotonic reasoning; Possibilistic logic

1. Introduction

Inductive Logic Programming (ILP) [1] provides a general framework for learning classical first-order logic rules. Reasonably efficient algorithms have been developed and several applications have been described in data-mining, especially for biochemistry databases, for natural language processing and more generally, in relational databases, since first-order logic is well-adapted for describing them.

The handling of exceptions is a serious bottleneck in ILP when learning from entailment with the classical separate and conquer methods (Progol [2], LogAn-H [3], ...), due to the formalization in first-order logic. In separate and conquer ILP approaches, the rules are learned one by one. When a rule is induced, all the examples that are correctly classified by it are usually removed (however, some algorithms such as ALEPH [4] can take them into account for the computation of the quality measure). Then, when a rule has some exceptions, i.e. some examples are misclassified by it, there is no way to compensate these exceptions by means of another rule. So, a hypothesis, i.e. a set of clauses, accumulates all the exceptions of the rules that appear in it. Moreover, when dealing with more than two classes, if an example is classified in two different classes (even if one is the right one), it is considered as misclassified since there is no way to prefer one class to the other. Note that this problem still remains for binary classification. This is why

* Corresponding author.
E-mail address: serrurie@irit.fr (M. Serrurier).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2007.04.016

Pattern Recognition 46 (2013) 1020–1038

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

A survey of graph theoretical approaches to image segmentation

Bo Peng [a,b], Lei Zhang [b,*], David Zhang [b]

[a] Department of Software Engineering, Southwest Jiaotong University, Chengdu, China
[b] Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China

ARTICLE INFO

Article history:
Received 18 December 2011
Received in revised form
15 September 2012
Accepted 18 September 2012
Available online 5 October 2012

Keywords:
Image segmentation
Graph theoretical methods
Minimal spanning tree
Graph cut

ABSTRACT

Image segmentation is a fundamental problem in computer vision. Despite many years of research, general purpose image segmentation is still a very challenging task because segmentation is inherently ill-posed. Among different segmentation schemes, graph theoretical ones have several good features in practical applications. It explicitly organizes the image elements into mathematically sound structures, and makes the formulation of the problem more flexible and the computation more efficient. In this paper, we conduct a systematic survey of graph theoretical methods for image segmentation, where the problem is modeled in terms of partitioning a graph into several sub-graphs such that each of them represents a meaningful object of interest in the image. These methods are categorized into five classes under a uniform notation: the minimal spanning tree based methods, graph cut based methods with cost functions, graph cut based methods on Markov random field models, the shortest path based methods and the other methods that do not belong to any of these classes. We present motivations and detailed technical descriptions for each category of methods. The quantitative evaluation is carried by using five indices – Probabilistic Rand (PR) index, Normalized Probabilistic Rand (NPR) index, Variation of Information (VI), Global Consistency Error (GCE) and Boundary Displacement Error (BDE) – on some representative automatic and interactive segmentation methods.

1. Introduction

Image segmentation is a classical and fundamental problem in computer vision. It refers to partitioning an image into several disjoint subsets such that each subset corresponds to a meaningful part of the image. As an integral step of many computer vision problems, the quality of segmentation output largely influences the performance of the whole vision system. A rich amount of literature on image segmentation has been published over the past decades. Some of them have achieved an extraordinary success and become popular in a wide range of applications, such as medical image processing [1–3], object tracking [4,5], recognition [6,7], image reconstruction [8,9] and so on.

Since the very beginning, image segmentation has been closely related to perceptual grouping or data clustering. Such a relationship was clearly pointed out by Wertheimer's gestalt theory [10] in 1938. In this theory, a set of grouping laws such as similarity, proximity and good continuation are identified to explain the particular way by which the human perceptual system groups tokens together. The gestalt theory has inspired many approaches to segmentation, and it is hoped that a good segmentation can capture perceptually important clusters which reflect local and/or global properties of the image. Early edge detection methods such as the Robert edge detector, the Sobel edge detector [11] and the Canny edge detector [12,13] are based on the abrupt changes in image intensity or color. Due to the distinguishable features of the objects and the background, a large number of thresholding based methods [14–16] have been proposed to separate the objects from the background. In the partial differential equations (PDE) based methods [17–21], the segmentation of a given image is calculated by evolving parametric curves in the continuous space such that an energy functional is minimized for a desirable segmentation. Region splitting and merging is another popular category of segmentation methods, where the segmentation is performed in an iterative manner until some uniformity criteria [22,23] are satisfied. The reviews of various segmentation techniques can be found for image thresholding methods [24], medical image segmentation [25,26], statistical level set segmentation [27], 3D image segmentation [28], edge detection techniques [29] and so on.

Among the previous image segmentation techniques, many successful ones benefit from mapping the image elements onto a graph. The segmentation problem is then solved in a spatially discrete space by the efficient tools from graph theory. One of the advantages of formulating the segmentation on a graph is that it might require no discretization by virtue of purely combinatorial operators and thus incur no discretization errors. Despite the

* Corresponding author.
E-mail address: cslzhang@comp.polyu.edu.hk (L. Zhang).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.09.015

Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Linear Algebra and its Applications 416 (2006) 745–758
LINEAR ALGEBRA AND ITS APPLICATIONS
www.elsevier.com/locate/laa

Greedy pathlengths and small world graphs

Desmond J. Higham

Department of Mathematics, University of Strathclyde, Glasgow G1 1XH, UK

Received 21 April 2005; accepted 3 January 2006
Available online 8 February 2006
Submitted by M. Neumann

Abstract

We use matrix analysis to study a cycle plus random, uniform shortcuts—the classic small world model. For such graphs, in addition to the usual edge and vertex information there is an underlying metric that determines distance between vertices. The metric induces a natural greedy algorithm for navigating between vertices and we use this to define a pathlength. This pathlength definition, which is implicit in [J. Kleinberg, The small-world phenomenon: an algorithmic perspective, in: Proceedings of the 32nd ACM Symposium on Theory of Computing, 2000] is entirely appropriate in many message passing contexts. Using a Markov chain formulation, we set up a linear system to determine the expected greedy pathlengths and then use techniques from numerical analysis to find a continuum limit. This gives an asymptotically correct expression for the expected greedy pathlength in the limit of large network size: both the leading term and a sharp estimate of the remainder are produced. The results quantify how the greedy pathlength drops as the number of shortcuts is increased. Further, they allow us to measure the amount by which the greedy pathlength, which is based on local information, exceeds the traditional pathlength, which requires knowledge of the whole network. The analysis allows for either O(1) shortcuts per node or O(1) shortcuts per network. In both cases we find that the greedy algorithm fails to exploit fully the existence of short paths.

AMS classification: 68R10; 60J10; 65L05

Keywords: Continuum limit; Differential equation; Finite difference; Greedy algorithm; Mean hitting time; Markov chain; Small world phenomenon

E-mail address: djh@maths.strath.ac.uk

0024-3795/$ - see front matter
doi:10.1016/j.laa.2006.01.001

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Physics Procedia 25 (2012) 1761 – 1768

Physics Procedia

2012 International Conference on Solid State Devices and Materials Science

A Partition Method for Graph Isomorphism

Lijun Tian, Chaoqun Liu and Jianquan Xie

Information Technology Department Hunan University of Finance and Economics No.139, Fenglin 2nd Road, Changsha, 410205, China

Abstract

Complexity of the graph isomorphism algorithms mainly depends on matching time which is directly related to efficiency of their partition methods. This paper proposed a partition method by sorted sequences of length-L path numbers, and divided cells of partition into 3 categories: not similar; completely similar; similar but not completely. The method was tested on several types of graphs with different order, each type with the same order 100 graphs. The results indicate that not similar cells can be refined by adding path length to other types or trivial cells if the vertex is not similar with all other vertices. For almost all asymmetric graphs, the path length is a small value, e.g., for 6-regular graphs with 100~1000 vertices the average path length is 4 to get all cells to trivial ones.

Keywords: Graph isomorphism; Partition method; Similar; Path length; Pattern recognition.

1. Introduction

Graph-based methodologies have been proposed as a powerful tool for pattern recognition and computer vision starting from the late 1970s [1,2]. All of these approaches are related to problems called as graph-matching [3].

Graph-matching problems can be divided into two categories: exact and inexact. The graph isomorphism (GI) is the simplest form of exact graph matching, which is still an open question whether it is a NP-complete problem or not [4], while other problems such as subgraph isomorphism problem are proved to be NP-complete.

GI problems can be solved by a type of brute-force backtrack search, while it yield O(n!) time for an n-vertex graph in the worst case[5]. This naive approach can be approved by classifying the vertices of the graph into more than one class, i.e., a partition of vertex set. Let$\pi=(\pi_1,\ldots\pi_r)$ is such a partition, there need only $\prod_{i=1}^{r}|\pi_i|$ time to match.

1875-3892 © 2012 Published by Elsevier B.V. Selection and/or peer-review under responsibility of Garry Lee
doi:10.1016/j.phpro.2012.03.308

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 18 (2013) 60 – 69

Procedia Computer Science

2013 International Conference on Computational Science

Data analysis with intersection graphs

V. M. Vairinhos [a,*], V. Lobo [a], P. Galindo Villardón [b]

[a]Centro de Investigação Naval, Escola Naval, Almada 2810-001, Portugal
[b]Departamento de Estadística, Universidad de Salamanca, Salamanca 37007, España

Abstract

This paper presents a new framework for multivariate data analysis, based on graph theory, using intersection graphs [1]. We have named this approach DAIG – Data Analysis with Intersection Graphs. This new framework represents data vectors as paths on a graph, which has a number of advantages over the classical table representation of data. To do so, each node represents an atom of information, *i.e.* a pair of a variable and a value, associated with the set of observations for which that pair occurs. An edge exists between a pair of nodes whenever the intersection of their respective sets is not empty. We show that this representation of data as an intersection graph allows an easy and intuitive geometric interpretation of data observations, groups of observations, and results of multivariate data analysis techniques such as biplots, principal components, cluster analysis, or multidimensional scaling. These will appear as paths on the graph, relating variables, values and observations. This approach allows for a compact and memory efficient representation of data that contains many missing values or multi-valued attributes. The basic principles and advantages of this approach are presented with an example of its application to a simple toy problem. The main features of this methodology are illustrated with the aid software specifically developed for this purpose.

Selection and peer review under responsibility of the organizers of the 2013 International Conference on Computational Science

Keywords: Categorical data ; data models ; data structures ; intersection graphs ; multivariate data analysis.

1. Introduction

The widespread use of information technology has led to a dramatic increase in the availability of raw data, which in turn has driven the need to use new and more powerful multivariate data analysis techniques. One of

* Corresponding author . Tel.: +351-918-234-755
E-mail address: valter.vairinhos@sapo.pt

1877-0509
Selection and peer review under responsibility of the organizers of the 2013 International Conference on Computational Science
doi:10.1016/j.procs.2013.05.169

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 15 (2011) 5455 – 5459

Procedia Engineering

www.elsevier.com/locate/procedia

Advanced in Control Engineeringand Information Science

RF_MISMOC: Improvement of MISMOC graph based classification algorithm

Maryam Kohzadi[a] ,Mohammad reza Keyvanpour[b], a*

[a,b]*Department of Computer Engineering, Alzahra University, Tehran, Iran*

Abstract

In recent years, the graph mining has gained much attention in the area of data mining. A novel technique called mining interesting substructures in molecular data for classification (MISMOC) is one of the most efficient algorithms in graph based classification. In this paper, we propose a novel technique called RF_MISMOC (Relative Frequency MISMOC) for computing interestingness of patterns by considering relative frequency of patterns in each class. In addition, we have improved the performance of the base algorithm by selecting equal numbers of interesting indicator patterns of classes and also determining optimum threshold value for selection of indicator patterns. The experimental results demonstrate that, the proposed algorithm has improved efficiency of the base algorithm.

Keywords: graph based classification, frequent sub graph, directed graph, relative frequency.

1. Introduction

The key point in graph based classification is discovering of distinctive patterns which could assign different data to corresponding classes. To find such patterns for graph structured data, graph mining techniques are used. Graph mining's algorithms, discover frequent sub-graphs in each class and then use them for classification [1]. While the discovered frequent sub-graphs could characterize each graph classes, they may not be very useful in discriminating different classes. Therefore, the aim of graph based classification is to discover interesting sub-graphs for each class, these sub-graphs are patterns which appeared more frequently in a certain class than other classes [2, 3]. The MISMOC algorithm, first removes sub-graphs that are not frequent enough for classification. Then by performing a statistical test, it only keeps sub-graphs those are frequent in one classes rather than other classes. Those that remain are interesting sub-graphs, which are not only characterizing a class of graphs, but they also could discriminate different classes.

* Corresponding author.
E-mail address:kohzadi.maryam@gmail.com.

1877-7058 © 2011 Published by Elsevier Ltd.
doi:10.1016/j.proeng.2011.08.1012

J. Vis. Commun. Image R. 24 (2013) 635–646

Contents lists available at SciVerse ScienceDirect

J. Vis. Commun. Image R.

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

Robust common visual pattern discovery using graph matching

Hongtao Xie [a,b], Yongdong Zhang [a,*], Ke Gao [a], Sheng Tang [a], Kefu Xu [b], Li Guo [b], Jintao Li [a]

[a] Advanced Computing Research Laboratory, Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, China
[b] Institute of Information Engineering, Chinese Academy of Sciences, National Engineering Laboratory for Information Security Technologies, China

ARTICLE INFO

Article history:
Received 1 December 2011
Accepted 21 April 2013
Available online 30 April 2013

Keywords:
Common visual pattern
Graph matching
Maximal clique
Quadratic optimization
Feature correspondence
Point set matching
Object recognition
Near-duplicate image retrieval

ABSTRACT

Discovering common visual patterns (CVPs) between two images is a difficult and time-consuming task, due to the photometric and geometric transformations. The state-of-the-art methods for CVPs discovery are either computationally expensive or have complicated constraints. In this paper, we formulate CVPs discovery as a graph matching problem, depending on pairwise geometric compatibility between feature correspondences. To efficiently find all CVPs, we propose a novel framework which consists of three components: Preliminary Initialization Optimization (PIO), Guided Expansion (GE) and Post Agglomerative Combination (PAC). PIO gets the initial CVPs and reduces the search space of CVPs discovery, based on the internal homogeneity of CVPs. Then, GE anchors on the initializations and gradually explores them, to find more and more correct correspondences. Finally, to reduce false and miss detection, PAC refines the discovery result in an agglomerative way. Experiments and applications conducted on benchmark datasets demonstrate the effectiveness and efficiency of our method.

1. Introduction

A common visual pattern (CVP) is the common part of two images, which has coherent spatial layout and similar visual content [1]. Discovering CVPs refers to establishing correct correspondences between two images. CVPs discovery is becoming increasingly important for various applications, such as object retrieval [1], image categorization and recognition [2,3], point set matching [4], near-duplicate image detection [5] and image database browsing [6].

Discovering CVPs is a difficult task, and there are several challenges. Firstly, significant photometric and geometric transformations usually take place between two images, such as occlusions, cropping, adding noise, changes of illumination, scale, viewpoint, and contrast, or even nonrigid deformations. Under the combinations of transformations and deformations, two instances of a CVP may differ not only in visual appearance, but also in 2D layout. Secondly, it lacks a priori knowledge of the CVPs, thus not known in advance the positions, shape, appearances, scales and the total number of CVPs. Finally, it is complex to detect a single CVP between two images, and thus finding all candidate CVPs will inevitably be computationally expensive. Fig. 1 shows such a challenging example with two CVPs between two images.

Recently, several CVPs discovery methods have been proposed [4–10,14–16]. However, existing approaches can only deal with weakly supervised cases with relatively slight occlusions and deformations [1,7], hence limiting the target image categories [4]. Besides, some of them rely on the initialization and do not guarantee the global optimal solution [5,6], or have complicated constraints and high computational cost [8–10,14].

In this paper, we formulate CVPs discovery as a graph matching problem by defining an objective function based on pairwise geometric compatibility between feature correspondences [11]. Firstly, we extract local features [12,13] in each image and get potential feature correspondences between two images. Then, we build a similarity graph, the nodes of which represent potential correspondences, and the edges represent the pairwise geometric consistency between corresponding correspondences. So, the spatially coherent feature correspondences i.e., a CVP constitute a dense subgraph [14], which is a weighted counterpart of maximal clique of un-weighted graph [17]. In this way, the CVPs discovery is formulated as a graph matching problem.

Finding the subgraphs (CVPs discovery) is NP-hard in general, but we solve this problem in an effective and efficient way, which contains three successive steps. Firstly, the Preliminary Initialization Optimization (PIO) algorithm is proposed to reduce the search space of CVPs discovery. PIO takes advantage of the internal homogeneity of CVPs, so it not only has better initializations of candidate CVPs, but also improves time performance. Secondly, we put forward a Guided Expansion (GE) method. GE anchors on the initializations of PIO, and gradually explores their neighboring correspondences, trying to construct more and more complete

* Corresponding author: Fax: +86 010 82546701.
E-mail addresses: xiehongtao@iie.ac.cn (H. Xie), zhyd@ict.ac.cn (Y. Zhang), kegao@ict.ac.cn (K. Gao), ts@ict.ac.cn (S. Tang), xukefu@iie.ac.cn (K. Xu), guoli@iie.ac.cn (L. Guo), jtli@ict.ac.cn (J. Li).

1047-3203/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2013.04.012

A COMPARATIVE STUDY OF FREQUENT SUBGRAPH MINING ALGORITHMS

K.Lakshmi[1] and Dr. T. Meyyappan[2]

[1.] Department of MCA, Sir M.Visvesvaraya Institute of Technology, Bangalore.
lakshmi_kes@rediffmail.com
[2.] Department of Computer Science and Engineering, Alagappa University,Karaikudi.
meyslotus@yahoo.com

ABSTRACT

Data mining algorithms are facing the challenge to deal with an increasing number of complex objects. Graph is a natural data structure used for modeling complex objects. Frequent subgraph mining is another active research topic in data mining . A graph is a general model to represent data and has been used in many domains like cheminformatics and bioinformatics. Mining patterns from graph databases is challenging since graph related operations, such as subgraph testing, generally have higher time complexity than the corresponding operations on itemsets, sequences, and trees. Many frequent subgraph Mining algorithms have been proposed. SPIN, SUBDUE, g_Span, FFSM, GREW are a few to mention. In this paper we present a detailed survey on frequent subgraph mining algorithms, which are used for knowledge discovery in complex objects and also propose a frame work for classification of these algorithms. The purpose is to help user to apply the techniques in a task specific manner in various application domains and to pave wave for further research.

KEYWORDS

Frequent subgraph mining, Isomorphism, Pattern growth, Apriori

1. INTRODUCTION

Knowledge discovery in complex objects involves understanding the relationship between their components. Examples are the Machine learning in domains such as bioinformatics, drug discovery, adverse drug events and web data mining. Graphs are natural data structures to model such relations, with nodes representing objects and edges the relationships between them. In this context, finding similarity between graphs is important. Simple ways of comparing graphs which are based on pair wise comparison of nodes or edges, are possible in quadratic time, yet may neglect information represented by the structure of the graph.

As interaction networks are graphs, where each node represents for example, a protein and each edge represents the presence of an interaction, Conventionally there are two ways of measuring similarity between graphs. One approach is to perform a pair wise comparison of the nodes and/or edges in two networks, and calculate an overall similarity score for the two networks from the similarity of their components. This approach takes time quadratic in the number of nodes and edges, and is thus computationally feasible even for large graphs. However, this strategy is flawed in that it completely neglects the structure of the networks, treating them as sets of nodes and edges instead of graphs. A more principled alternative would be to deem two networks similar if they share many common substructures, or more technically, if they share many common subgraphs. To compute this, however, we would have to solve the so-called

A New proposal for Graph Classification using Frequent Geometric Subgraphs

Andrés Gago-Alonso[a,*], Alfredo Muñoz-Briseño[a], Niusvel Acosta-Mendoza[a]

[a] *Advanced Technologies Application Center, 7a ♯ 21812, Siboney, Playa, CP: 12200, Havana, Cuba*
E-mail: {agago, amunoz, nacosta}@cenatav.co.cu

Abstract

Geometric graph mining has bees identified as a need in many applications. This technique detect patterns with some tolerance under a geometric transformation. To meet this need, some graph miners have been developed for detecting frequent geometric subgraphs. However, there are few works for applying this kind of geometric patterns as feature for classification tasks. In this paper, a new geometric graph miner and a framework for using frequent geometric subgraphs in classification, are proposed. Our solution was tested in two real collections. The experimentation on these collections shows that our proposal gets better results than graph-based image classification using non-geometric graph miners.

Keywords: Mining methods and algorithms, classification, clustering, frequent subgraph mining

1. Introduction

In recent years, several authors have developed techniques and tools for facing tasks related with converting large volumes of data into useful information [19]. Frequent pattern discovery is an example of such techniques [37], when the objects of these datasets are represented as graphs [25]. These techniques has been successfully employed for classification tasks [1, 12, 20], using frequent subgraphs as features for representing objects. As example of this, in literature we can find classification of images [1, 8, 29, 30, 20], texts [21] and chemical compounds [6, 18].

In the graph collections used in these applications, geometric features of vertices can be considered for modelling the objects. For example, atom coordinates are included in some molecular datasets [24], the spatial coordinates of regions of interest are considered in image collections [1], among others [32]. These datasets are commonly affected by some geometric shaped distortions of similar structures in several objects. Therefore, the application of a mechanism for dealing with such distortions can help to improve classification results in geometric graph databases.

*Corresponding author

DOI : 10.1007/978-3-642-20844-7

Two New Graph Kernels and Applications to Chemoinformatics

Benoit Gaüzère†, Luc Brun†, and Didier Villemin‡

†GREYC UMR CNRS 6072, ‡LCMT UMR CNRS 6507,
Caen, France
{benoit.gauzere,didier.villemin}@ensicaen.fr,
luc.brun@greyc.ensicaen.fr

Abstract. Chemoinformatics is a well established research field concerned with the discovery of molecule's properties through informational techniques. Computer science's research fields mainly concerned by the chemoinformatics field are machine learning and graph theory. From this point of view, graph kernels provide a nice framework combining machine learning techniques with graph theory. Such kernels prove their efficiency on several chemoinformatics problems. This paper presents two new graph kernels applied to regression and classification problems within the chemoinformatics field. The first kernel is based on the notion of edit distance while the second is based on sub trees enumeration. Several experiments show the complementary of both approaches.

Keywords: edit-distance, graph kernel, chemoinformatics

1 Introduction

Chemoinformatics aims to predict or analyse molecule's properties through informational techniques. One of the major principle in this research field is the *similarity principle*, which states that two structurally similar molecules should have similar activities and properties. The structure of a molecule is naturally encoded by a labeled graph $G = (V, E, \mu, \nu)$, where the unlabeled graph (V, E) encodes the structure of the molecule while μ maps each vertex to an atom's label and ν characterizes a type of bond between two atoms (single, double, triple or aromatic).

A first family of methods introduced within the Quantitative Structure-Activity Relationship (QSAR) field is based on the correlation between molecule's descriptors such as the number of atoms and molecule's properties (e.g. molecule's boiling point). Vectors of descriptors may be defined from structural information [2], physical properties or biological activities and may be used within any statistical machine learning algorithm to predict molecule's properties. Such a scheme allows to benefit from the large set of tools available within the statistical machine learning framework. However, the definition of a vector from a molecule, ie. a graph, induces a loss of information. Moreover, for each application, the definition of a vectorial description of each molecule remains heuristic.

Mach Learn (2011) 82: 95–121
DOI 10.1007/s10994-010-5215-6

Efficiently mining δ-tolerance closed frequent subgraphs

Ichigaku Takigawa · Hiroshi Mamitsuka

Received: 1 June 2009 / Accepted: 1 May 2010 / Published online: 17 September 2010

Abstract The output of frequent pattern mining is a huge number of frequent patterns, which are very redundant, causing a serious problem in understandability. We focus on mining frequent subgraphs for which well-considered approaches to reduce the redundancy are limited because of the complex nature of graphs. Two known, standard solutions are closed and maximal frequent subgraphs, but closed frequent subgraphs are still redundant and maximal frequent subgraphs are too specific. A more promising solution is δ-tolerance closed frequent subgraphs, which decrease monotonically in δ, being equal to maximal frequent subgraphs and closed frequent subgraphs for $\delta = 0$ and 1, respectively. However, the current algorithm for mining δ-tolerance closed frequent subgraphs is a naive, two-step approach in which frequent subgraphs are all enumerated and then sifted according to δ-tolerance closedness. We propose an efficient algorithm based on the idea of "reverse-search" by which the completeness of enumeration is guaranteed and for which new pruning conditions are incorporated. We empirically demonstrate that our approach significantly reduced the amount of real computation time of two compared algorithms for mining δ-tolerance closed frequent subgraphs, being pronounced more for practical settings.

Keywords Frequent subgraph mining · δ-tolerance closedness · Partial reverse search

Editors: S.V.N. Vishwanathan, Samuel Kaski, Jennifer Neville, and Stefan Wrobel.

I. Takigawa (✉) · H. Mamitsuka
Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji 611-0011, Japan
e-mail: takigawa@kuicr.kyoto-u.ac.jp

H. Mamitsuka
e-mail: mami@kuicr.kyoto-u.ac.jp

I. Takigawa · H. Mamitsuka
Institute for Bioinformatics Research and Development, BIRD, Japan Science and Technology Agency, JST, Tokyo, Japan

available at www.sciencedirect.com

ScienceDirect

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

Survey

Graph clustering

*Satu Elisa Schaeffer**

Laboratory for Theoretical Computer Science, Helsinki University of Technology TKK, P.O. Box 5400, FI-02015 TKK, Finland

ARTICLE INFO

Article history:
Received 29 January 2007
Received in revised form
8 May 2007
Accepted 28 May 2007

ABSTRACT

In this survey we overview the definitions and methods for graph clustering, that is, finding sets of "related" vertices in graphs. We review the many definitions for what is a cluster in a graph and measures of cluster quality. Then we present global algorithms for producing a clustering for the entire vertex set of an input graph, after which we discuss the task of identifying a cluster for a specific seed vertex by local computation. Some ideas on the application areas of graph clustering algorithms are given. We also address the problematics of evaluating clusterings and benchmarking cluster algorithms.

1. Introduction

Any nonuniform data contains underlying structure due to the heterogeneity of the data. The process of identifying this structure in terms of grouping the data elements is called *clustering*, also called *data classification* [152]. The resulting groups are called *clusters*. The grouping is usually based on some *similarity measure* defined for the data elements. Clustering is closely related to *unsupervised learning* in pattern recognition systems [81]. A basic task in unsupervised learning is to classify a data set into two or more classes based on a similarity measure over the data, without resorting to any *a priori* information on how the classification should be done.

Graphs are structures formed by a set of *vertices* (also called *nodes*) and a set of *edges* that are *connections* between pairs of vertices. *Graph clustering* is the task of grouping the vertices of the graph into clusters taking into consideration the edge structure of the graph in such a way that there should be many edges *within* each cluster and relatively few *between* the clusters. Graph clustering in the sense of grouping the vertices of a given input graph into clusters, which is the topic of this survey, should *not* be confused with the clustering of *sets of graphs* based on structural similarity; such clustering of graphs as well as measures of graph similarity is addressed in other literature [38,124,168,169,202,206], although many of the techniques involved are closely related to the task of finding clusters *within* a given graph.

As the field of graph clustering has grown quite popular and the number of published proposals for clustering algorithms as well as reported applications is high, we do not even pretend to be able to give an exhaustive survey of *all* the methods, but rather an explanation of the methodologies commonly applied and pointers to some of the essential publications related to each research branch.

1.1. Outline of the survey

We begin by providing basic definitions in Section 2. In Section 3 we proceed to defining the task of graph clustering by discussing the different definitions of clusterings and clusters. These definitions lead into definitions of similarity measures discussed in Section 4, global clustering algorithms summarized in Section 5, and local methods presented in

* *Corresponding address:* Universidad Autónoma de Nuevo León, Facultad de Ingeniería Mecánica y Eléctrica, Posgrado en Ingeniería de Sistemas (PISIS), AP 126-F, Ciudad Universitaria, San Nicolás de los Garza, NL 66450, Mexico.
E-mail address: elisa.schaeffer@gmail.com.

1574-0137/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cosrev.2007.05.001

Theoretical Computer Science 411 (2010) 2475–2486

Contents lists available at ScienceDirect

Theoretical Computer Science

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

Finding common structured patterns in linear graphs☆

Guillaume Fertin [a], Danny Hermelin [b], Romeo Rizzi [c], Stéphane Vialette [d,*]

[a] LINA, CNRS UMR 6241, Université de Nantes, 2 rue de la Houssinière, 44322 Nantes, France
[b] Department of Computer Science, University of Haifa, Mount Carmel, Haifa 31905, Israel
[c] DIMI, Università degli Studi di Udine, Via delle Scienze, 208 I-33100 Udine (UD), Italy
[d] LIGM, CNRS UMR 8049, Université Paris-Est Marne-la-Vallée, 5 Bd Descartes 77454 Marne-la-Vallée, France

ARTICLE INFO

Article history:
Received 13 November 2007
Received in revised form 12 February 2010
Accepted 15 February 2010
Communicated by A. Apostolico

Keywords:
Linear graphs
Approximation

ABSTRACT

A linear graph is a graph whose vertices are linearly ordered. This linear ordering allows pairs of disjoint edges to be either preceding ($<$), nesting ($\sqsubset$) or crossing ($\between$). Given a family of linear graphs, and a non-empty subset $\mathcal{R} \subseteq \{<, \sqsubset, \between\}$, we are interested in the Maximum Common Structured Pattern (MCSP) problem: find a maximum size edge-disjoint graph, with edge pairs all comparable by one of the relations in $\mathcal{R}$, that occurs as a subgraph in each of the linear graphs of the family. The MCSP problem generalizes many structure-comparison and structure-prediction problems that arise in computational molecular biology.

We give tight hardness results for the MCSP problem for $\{<, \between\}$-structured patterns and $\{\sqsubset, \between\}$-structured patterns. Furthermore, we prove that the problem is approximable within ratios: (i) $2\mathcal{H}(k)$ for $\{<, \between\}$-structured patterns, (ii) $k^{1/2}$ for $\{\sqsubset, \between\}$-structured patterns, and (iii) $O(\sqrt{k \log k})$ for $\{<, \sqsubset, \between\}$-structured patterns, where k is the size of the optimal solution and $\mathcal{H}(k) = \sum_{i=1}^{k} 1/i$ is the kth harmonic number. Also, we provide combinatorial results concerning different types of structured patterns that are of independent interest in their own right.

1. Introduction

Many biological molecules such as RNA and proteins exhibit a three-dimensional structure that determines most of their functionalities. This three-dimensional structure can be modeled in two dimensions by an edge-disjoint linear graph, *i.e.*, a graph with linearly ordered vertices that are incident to exactly one edge. The corresponding structure-similarity or structure-prediction problems that arise in such contexts usually translate to finding common edge-disjoint subgraphs, or common *structured patterns*, that occur in a family of general linear graphs. Examples of such problems are the Longest Common Subsequence [22,23] problem, the Maximum Common Ordered Tree Inclusion [2,9,24] problem, the Arc-Preserving Subsequence [4,17,20] problem, and the Maximum Contact Map Overlap [18] problem. In this paper, we study a general framework for such problems which we call the Maximum Common Structured Pattern (MCSP) problem.

The MCSP problem was originally introduced (under a different name) by Davydov and Batzoglou [11] in the context of (non-coding) RNA secondary structure prediction via multiple structural alignment. There, an RNA sequence of n nucleotides is represented by a linear graph with n vertices, and an edge connects two vertices if and only if their corresponding

☆ A preliminary version of the paper appeared in Combinatorial Pattern Matching, 18th Annual Symposium, CPM 2007, London, Canada, July 9–11, 2007, Proceedings. Lecture Notes in Computer Science, 4580, Springer, 2007, ISBN 978-3-540-73436-9.

* Corresponding author. Tel.: +33 1 60 95 77 49; fax: +33 1 60 95 75 57.
E-mail addresses: Guillaume.Fertin@lina.univ-nantes.fr (G. Fertin), danny@cri.haifa.ac.il (D. Hermelin), Romeo.Rizzi@dimi.uniud.it (R. Rizzi), vialette@univ-mlv.fr (S. Vialette).

0304-3975/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.02.017

BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

SING: Subgraph search In Non-homogeneous Graphs

Raffaele Di Natale[1], Alfredo Ferro[1*], Rosalba Giugno[1], Misael Mongiovì[1], Alfredo Pulvirenti[1], Dennis Shasha[2]

Abstract

Background: Finding the subgraphs of a graph database that are isomorphic to a given query graph has practical applications in several fields, from cheminformatics to image understanding. Since subgraph isomorphism is a computationally hard problem, indexing techniques have been intensively exploited to speed up the process. Such systems filter out those graphs which cannot contain the query, and apply a subgraph isomorphism algorithm to each residual candidate graph. The applicability of such systems is limited to databases of small graphs, because their filtering power degrades on large graphs.

Results: In this paper, SING (Subgraph search In Non-homogeneous Graphs), a novel indexing system able to cope with large graphs, is presented. The method uses the notion of *feature*, which can be a small subgraph, subtree or path. Each graph in the database is annotated with the set of all its features. The key point is to make use of feature locality information. This idea is used to both improve the filtering performance and speed up the subgraph isomorphism task.

Conclusions: Extensive tests on chemical compounds, biological networks and synthetic graphs show that the proposed system outperforms the most popular systems in query time over databases of medium and large graphs. Other specific tests show that the proposed system is effective for single large graphs.

Background

Graphs naturally model a multitude of complex objects in the real world. A chemical compound can be represented by a graph where atoms are vertices and bonds are edges. Biological networks model the complex of interactions among components in cells, (e.g. proteins, genes, metabolites). Social networks, the web, the water system and the power grid are all represented by graphs. A basic operation is the search of a query graph in a target graph or, more generally, in a database of graphs. Searching a molecular structure in a database of molecular compounds is useful to detect molecules that preserve chemical properties associated with a well known molecular structure. This can be used in screening and drug design. Searching subnetworks in biological networks helps to identify conserved complexes, pathways and motifs among species, and assist in the functional annotation of proteins and other cell components. The problem of searching for a query graph in a target graph is called *subgraph isomorphism* and is known to be NP-complete. Since the subgraph isomorphism test is expensive, screening all graphs of a large database can be unfeasible. Recently, indexing techniques for databases of graphs have been developed with the purpose of reducing the number of subgraph isomorphism tests involved in the query process. In a *preprocessing* phase the database of graphs is analyzed and an index is built. A query is processed in two phases. In the *filtering* step the index is used to discard the graphs of the database which cannot contain the query, producing a small set of *candidate* graphs. The set of candidates is then verified (*verification* step) by a subgraph isomorphism algorithm and all the resulting matches are reported.

Most graph indexing tools are based on the concept of *feature*. Depending on the particular system, a feature can be either a small graph [1-3], a tree [4] or a path [5,6]. The filtering property is based on checking whether the features of the query are contained in each target graph. In the preprocessing phase the database of graphs is scanned, the features are extracted from each graph and stored in the index data structure. During the

* Correspondence: ferro@dmi.unict.it
[1]Dipartimento di Matematica ed Informatica, Università di Catania, Catania, Italy

Mining Strongly Correlated Sub-graph Patterns by Considering Weight and Support Constraints

Gangin Lee and Unil Yun[1]

Department of Computer Science,
Chungbuk National University, Republic of Korea
{abcnarak, yunei}@chungbuk.ac.kr

Abstract

Frequent graph mining is one of famous data mining fields that receive the most attention, and its importance has been raised continually as recent databases in the real world become more complicated. Weighted frequent graph mining is an approach for applying importance of objects in the real world to the graph mining, and numerous studies related to this have been conducted so far. However, all of the results obtained from this approach do not become actually useful information, and a significant portion of them may be meaningless ones even though they are weighted frequent sub-graph patterns. To overcome this problem, in this paper, we propose a novel method which can consider whether any sub-graph pattern has close correlation among elements in the pattern, called MSCG (Mining Strongly Correlated sub-Graph). In experimental results, we demonstrate that our MSCG outperforms a state-of-the-art method with respect to runtime and memory usage.

Keywords: *Affinity, Correlated pattern, Graph mining, Weighted frequent pattern mining*

1. Introduction

Data mining means a series of processes for finding hidden and useful information from large databases. Frequent pattern mining is one of the data mining fields which are most actively researched, and accordingly numerous techniques and methods related to this have been studied. However, as data derived from the real world have been complicated increasingly, the previous frequent pattern mining approaches have been faced with limitations since they deal with only simple databases composed of itemsets. To overcome this problem and mine complex data with graph forms, frequent graph mining methods [1, 2, 4, 5, 16] have been proposed, and thereafter advanced methods applying weight conditions [3, 6, 7, 10, 11] have been suggested to consider characteristics in the real world. Although the above methods find weighted frequent sub-graph patterns, they cannot determine how closely elements in any graph pattern are related. In this paper, to solve this issue, we propose a novel method for mining weighted frequent sub-graphs considering correlations among sub-graphs' elements, called MSCG (Mining Strongly Correlated sub-Graph), using a special and complex measure, called *weighted support affinity*. Through the method, we can obtain advantages in terms of mining performance as well as extract actually useful graph patterns from graph databases. Generally, mining sub-graphs from graph databases causes enormous overheads compared to mining itemsets from simple databases since lots of execution times are needed for graph isomorphism (NP-hard problem). However, since MSCG conducts pre-pruning operations with respect to sub-graphs with weak correlation, we can decrease the

[1] Corresponding Author, Unil Yun

EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

A hierarchical graph model for object cosegmentation

Yanli Li, Zhong Zhou* and Wei Wu

Abstract

Given a set of images containing similar objects, cosegmentation is a task of jointly segmenting the objects from the set of images, which has received increasing interests recently. To solve this problem, we present a novel method based on a hierarchical graph. The vertices of the hierarchical graph involve pixels, superpixels and heat sources, and cosegmentation is performed as iterative object refinement in the three levels. With the inter-image connection in the heat source level and the intra-image connection in the superpixel level, we progressively update the object likelihoods by transferring message across images via belief propagation, diffusing heat energy within individual image via random walks, and refining the foreground objects in the pixel level via guided filtering. Besides, a histogram based saliency detection scheme is employed for initialization. We demonstrate experimental evaluations with state-of-the-art methods over several public datasets. The results verify that our method achieves better segmentation quality as well as higher efficiency.

Keywords: Cosegmentation, Hierarchical graph, Heat source, Saliency detection, Belief propagation, Random walks, Guided filtering

1 Introduction

The term "cosegmentation" is first introduced by Rother et al. [1] in 2006, referring to the problem of simultaneously segmenting "similar" foreground objects in a set of images. The definition of "similar" commonly indicates the constraint that the distribution of some appearance cues such as color and texture in each image has to be similar. Cosegmentation has many potential applications. It can be used for summarizing personal photo album, guiding multiple images' editing, boosting unsupervised object recognition, improving content based image retrieval and so on.

Since the introduction of the problem, various methods have been presented. One type of methods handles the problem of multi-class cosegmentation, while others focus on binary cosegmentation. In this article, we are interested in binary cosegmentation and observe that for most applications of binary cosegmentation several criteria should be followed: (1) automation, i.e., it is executed without user interactions; (2) scalability, i.e., it can be applied to hundreds of images instead of two images or small sized image sets; (3) focusing on "object" instead of "stuff". Here the "object" refers to "foreground things" such as a person or a bird, while "stuff" refers to "background regions" such as road or sky; (4) high segmentation accuracy; (5) low running time. According to these criteria, existing methods have some limitations. For example, the iCoseg system presented by Batra et al. [2] can obtain highly accurate results, but requires user input. The methods reviewed by Vicente et al. [3] all focus on cosegmenting two images. The recently presented CoSand [4] only extracts similar large regions, thus it often omits the small foreground objects in the images. Methods based on topic discovery like [5-7] all take superpixels as computation nodes, and hence they suffer from detail loss because superpixels tend to merge foreground regions with the backgrounds. Some unsupervised object segmentation methods [8-11] extract objects from multiple images via iteratively learning class models and segmenting objects in pixel level, while they are time-consuming because the employed optimization schemes like graphcut [12] and belief propagation [13] are inefficient with a large number of pixel nodes.

In this article, we try to meet these criteria by extracting the foreground objects with a three-level hierarchical

*Correspondence: zz@vrlab.buaa.edu.cn
State Key Laboratory of Virtual Reality Technology & Systems, Beihang University, Beijing, China

Classification and Analysis of Frequent Subgraphs Mining Algorithms

Mohammad Reza Keyvanpour
Department of Computer Engineering, Alzahra University, Tehran, Iran
Email: keyvanpour@alzahra.ac.ir

Fereshteh Azizani
Department of Computer Engineering, Islamic Azad University, Qazvin Branch, Qazvin, Iran
Email: fereshteh.azizani@gmail.com

Abstract— **In recent years, data mining in graphs or graph mining have attracted much attention due to explosive growth in generating graph databases. The graph database is one type of database that consists of either a single large graph or a number of relatively small graphs. Some applications that produce graph database are as follows: Biological networks, semantic web and behavioral modeling. Among all patterns occurring in graph database, mining frequent subgraphs is of great importance. The frequent subgraph is the one that occurs frequently in the graph database. Frequent subgraphs not only are important themselves but also are applicable in other aspects of data analysis and data mining tasks, such as similarity search in graph database, graph clustering, classification, indexing, etc. So far, numerous algorithms have been proposed for mining frequent subgraphs. This study aims to create overall view of the algorithms through the analysis and comparison of their characterizations. To achieve the aim, the existing algorithms are classified based on their graph database and their subgraph generation way. The proposed classification can be effective in choosing applications appropriate algorithms and determination of graph mining new methods in this regard.**

Index Terms—**Graph database, Data mining, Graph mining, Frequent subgraph**

I. Introduction

Presenting data as graph make expressing the existing connection between data as natural. This characteristic of graphs causes the increasing application of them for modeling complex structures such as images [8], chemical components [20], protein structure [29], biological networks [28], social networks [41], web [30] and XML documents [5]. Due to the speed of creating and increasing the number of graphs of modeling of complex structures, it is necessary to use a method to analysis efficiently this extensive amount of data. This makes data mining among graphs or graph mining an important field in data mining.

Among all patterns occurring in graph database, mining frequent subgraphs is of great importance. The frequent subgraph is the one that occurs frequently in the graph databases. Frequent subgraphs not only are important themselves but are applicable in other aspects of data analysis and data mining tasks, such as similarity search in graph database, graph clustering, classification, indexing, etc. In classification and similarity search, using frequent subgraphs as feature leads to exact results and high scalability [7, 38, 39]. Clustering of spaces with high dimension is very challenging. As the exploration of frequent subgraphs in subsets of dimensions are easily possible, we can use frequent subgraphs as a solution in the clustering of subspace and clustering of spaces with high dimension [31]. Efficient search in graph database is an unavoidable issue in many applications including detection of cancer structures. Large amount of data in these databases makes the ordinal research and one-to-one test of the objects impossible and inefficient. By using frequent subgraphs mining strategies via indexing of frequent graphs, search speed can be increased considerably [37].

Mining frequent subgraphs is an iterative process consisting of two main steps [24]. The first step is candidate generation. In this step, the subgraphs that are probably iterative or they are frequent candidates are generated. The next step is counting step. In this step, the frequencies of generated candidates in database are counted. To do this, the interested candidate should be searched in graph database. The approach of this step is different depending upon the type of graph database on which graph mining process is done. If database is just including one single large graph, the number of occurrences of subgraph is counted in it. But if database is consisting of multiple small graphs, the number of graph occurrences is not important in a special graph and it is equal to the number graphs in which that subgraph is existing [17]. The generated algorithms for mining on a single large graph can be applied for a set of graphs but the opposite is not true. In both cases, counting the number of occurrences of candidates requires the investigation of subgraph isomorphism that is an NP-complete issue [10] and it is costly for great databases.

Considering the increasing importance of frequent subgraphs, this paper attempts to create a general view toward frequent subgraphs mining algorithms by introducing these algorithms and presenting their

doi:10.4304/jsw.7.1.220-227

Graph based Approach and Clustering of Patterns (GACP) for Sequential Pattern Mining

Ashish Patel [1], Amisha Patel [2]

[1] Department of Computer Engineering/Information Technology
Shri S'ad Vidya Mandal Institute of Technology
Bharuch, Gujarat – India

Abstract The sequential pattern mining generates the sequential patterns. It can be used as the input of another program for retrieving the information from the large collection of data. It requires a large amount of memory as well as numerous I/O operations. Multistage operations reduce the efficiency of the algorithm. The given GACP is based on graph representation and avoids recursively reconstructing intermediate trees during the mining process. The algorithm also eliminates the need of repeatedly scanning the database. A graph used in GACP is a data structure accessed starting at its first node called root and each node of a graph is either a leaf or an interior node. An interior node has one or more child nodes, thus from the root to any node in the graph defines a sequence. After construction of the graph the pruning technique called clustering is used to retrieve the records from the graph. The algorithm can be used to mine the database using compact memory based data structures and cleaver pruning methods.

Index Terms-GACP, data mining, sequential data mining, clustering

I. I. INTRODUCTION

Data mining is a relatively new research area that extracts knowledge which is hidden in the database and hence very useful in information retrieval. Frequent pattern mining from sequential data is one of the most important tasks. Frequent patterns are required in satellite images, customer databases, telecommunication systems, frequent buying patterns etc. Agrawal R. and Shrikant R. have first found out some algorithms for mining frequent pattern from a large collection of data sequences [1]. They have used support for analyzing the percentage of data sequences containing the pattern. Later Agrawal R. and Shrikant R. have used some constraints like minimum and maximum gap between adjacent elements of a pattern [2]. Gradually in the field of computing storage and processing devices are become boundless and have allowed the users to store and process huge collection of data.

Example of such collection includes web site usage analysis, medical reports, science and engineering databases etc. They have drawn the attention of a number of researchers in the field of data mining. Mainly the collected data is in sequential form, hence arises the scope for different techniques for exploring sequential patterns.

The goal is to find trends across large number of transactions that can be used to understand and exploit sequential patterns. Given a Sequence Database, the problem to find frequently occurring Sequential patterns on the basis of minimum support provided. Here a brief study of Generalized Sequential Pattern and Web access pattern mine is done which is much more efficient than the candidate generation based algorithms. But it required much space to store the intermediate trees which are generated during the process. So a new algorithm GACP is proposed to make the mining more efficient in terms of storage and time. GACP uses the concept of graph traversal by constructing the graph in one database scan only. The constructed graph then can be used by the algorithm to find the sequential patterns or order list of events from the database. The algorithm uses clustering techniques to prune the paths of the graph.

II. BACKGROUND

The sequential pattern mining problem was first introduced by Agrawal and Srikant[1]. Given a set of sequences, where each sequence consists of a list of elements and each element consists of a set of items, and given a user-specified min support threshold, sequential pattern mining is to find all of the frequent

J Braz Comput Soc
DOI 10.1007/s13173-010-0027-x

ORIGINAL PAPER

A graph clustering algorithm based on a clustering coefficient for weighted graphs

Mariá C.V. Nascimento · André C.P.L.F. Carvalho

Received: 27 May 2010 / Accepted: 29 November 2010

Abstract Graph clustering is an important issue for several applications associated with data analysis in graphs. However, the discovery of groups of highly connected nodes that can represent clusters is not an easy task. Many assumptions like the number of clusters and if the clusters are or not balanced, may need to be made before the application of a clustering algorithm. Moreover, without previous information regarding data label, there is no guarantee that the partition found by a clustering algorithm automatically extracts the relevant information present in the data. This paper proposes a new graph clustering algorithm that automatically defines the number of clusters based on a clustering tendency connectivity-based validation measure, also proposed in the paper. According to the computational results, the new algorithm is able to efficiently find graph clustering partitions for complete graphs.

Keywords Clustering coefficient · Graph clustering · Combinatorial optimization

1 Introduction

Data clustering deals with the discovery of data patterns, in the form of data clusters, in the objects from a dataset. For such, objects with similar characteristics are placed into the same group (cluster) and objects with different features are placed into different clusters. Several clustering algorithms have been proposed in the literature, based on several approaches. These algorithms have been successfully applied to a wide variety of problems, including applications from areas like bioinformatics [11], image processing [25] and market segmentation [2]. Despite the good results obtained by the use of clustering algorithms in several problem domains, cluster analysis is still seen as a challenging problem. Some particular clustering problems require more sophisticated algorithms.

A specific problem of clustering is known as graph clustering. Graph clustering looks for patterns among nodes in a graph in order to produce a meaningful node partitioning. Many inferences about the node partition may provide useful information regarding the data. A few examples of the benefits, as well as different approaches for graph clustering, can be found in [24].

Additionally, in spite of the large number of clustering algorithms, few of them are able to automatically discover partitions without the information of the number of clusters beforehand. Automatic graph clustering algorithms, able to define by themselves the number of clusters, play an important role in data analysis, since they allow a more efficient application of clustering algorithms to a dataset without prior knowledge of the data conformation. Therefore, the investigation of new clustering algorithms able to deal with graph clustering problems and to automatically define the number of clusters is an important research issue.

Automatic clustering algorithms usually rely on a validation criterion to select the number of clusters. Moreover, the evaluation of the quality of clustering partitions is not as simple and direct as the evaluation of classification models. Several validation measures have been specifically designed to assess graph clustering partitions. Among them, we can cite modularity [19]. Modularity evaluates the difference be-

M.C.V. Nascimento (✉) · A.C.P.L.F. Carvalho
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, Caixa Postal 668, São Carlos, SP,
CEP 13560-970, Brazil
e-mail: mariah@icmc.usp.br

A.C.P.L.F. Carvalho
e-mail: andre@icmc.usp.br

Published online: 21 December 2010

Integrated Computer-Aided Engineering 18 (2011) 91–106
DOI 10.3233/ICA-2011-0359
IOS Press

A graph-mining algorithm for automatic detection and counting of embryonic stem cells in fluorescence microscopy images

Geisa M. Faustino[a,*], Marcelo Gattass[a], Carlos J. P. de Lucena[a], Priscila B. Campos[b] and Stevens K. Rehen[b]
[a]*Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, Brazil*
[b]*Instituto de Ciências Biomédicas, Universidade Federal do Rio de Janeiro, UFRJ, Rio de Janeiro, RJ, Brazil*

Abstract. Many cell-based research studies require the counting of cells in order to understand and validate experiments through statistical analyses. Although progress in imaging technology has enabled the automation of cell counting for many different cell types, this process still has to be done manually in the case of images of embryonic stem cells. In this paper, we present a new automatic algorithm to detect and count embryonic stem cells in fluorescence microscopy images that identifies pluripotent stem cells cultured *in vitro*. Our approach uses luminance information to generate a graph-based image representation. The cell pattern is defined as a subgraph, and a graph-mining process is applied to detect the cells. The method is tolerant to variations in cell size and shape. Moreover, it can easily be parameterized to handle different image groups resulting from distinct differentiation protocols. The paper presents numerical results from tests made on a database with more than two hundred images, including EB cryosection, embryoid body cell migration, murine embryonic stem cell colonies under murine embryonic fibroblast, and neurosphere images. The results from our algorithm were validated by expert biologists, and provide good precision, recall and F-measure. Finally, a comparative study with the widely used watershed algorithm is presented.

Keywords: Automatic cell counting, fluorescence microscopy image, graph-based image representation, graph mining, graph clustering

1. Introduction

Researchers have been developing a number of new techniques and software applications for biological and medical health care purposes [18,19,22,32,40,41]. Recent efforts have focused on eliminating subjectivity, accelerating processes such as cell counting or microcalcification detection, and helping physicians improve their daily practice, make accurate diagnosis, and prevent diseases.

Embryonic stem (ES) cell is the term used to define cells that have unlimited proliferation and can originate any kind of cell in an organism. Since ES cells were discovered, the possibility of using them to treat a variety of diseases has encouraged many research projects in biomedical areas. Because of their pluripotency property and the possibility to direct their differentiation *in vitro* to specific cell types, ES cells have become an alternative for cell-based treatments of various diseases, such as diabetes, Parkinson's, stroke, heart disease, and spinal cord injury, among others.

Cell counting plays an important role in statistical analysis and allows specialists to understand and validate experiments. When cells are stained properly, visual analysis can reveal biological mechanisms. Using different cell markers, researchers are able to determine, for instance, the total number of cells, how many have become specialized mature cells, and how many cells died. Although progress in imaging technology

*Corresponding author: Geisa M. Faustino, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, Brazil. E-mail: gfaustino@inf.puc-rio.br.

ISSN 1069-2509/11/$27.50

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 2, August 2012

Innovative Study to the Graph-based Data Mining: Application of the Data Mining

Amit Kr. Mishra, Pradeep Gupta, Ashutosh Bhatt, Jainendra Singh Rana

Abstract*— Graph-based data mining represents a collection of techniques for mining the relational aspects of data represented as a graph. Two major approaches to graph based data mining are frequent sub graph mining and graph-based relational learning. This article will focus on one particular approach embodied in the Subdue system, along with recent advances in graph-based supervised learning, graph-based hierarchical conceptual clustering, and graph-grammar induction. The need for mining structured data has increased rapidly. One of the best studied data structures in computer science and discrete mathematics are graphs. Graph based data mining has become quite popular in the last few years. This paper introduces the theoretical basis of graph based data mining and surveys the state of the art of graph-based data mining. Brief descriptions of some representative approaches are provided as well.

***Index Terms*— Graph, Tree, Path, Structured Data, Data Mining.**

I. INTRODUCTION

During the past decade, the field of data mining has emerged as a novel field of research, investigating interesting research issues and developing challenging real-life applications. The objective data formats in the beginning of the field were limited to relational tables and transactions where each in-stance is represented by one row in a table or one transaction represented as a set. However, the studies within the last several years began to extend the classes of considered data to semi-structured data such as HTML and XML texts symbolic sequences, ordered trees and relations represented by advanced logics. Graph mining has a strong relation with the afore mentioned Multi-relational data mining. However, the main objective of graph mining is to provide new principles and efficient algorithms to mine topological substructures embedded in graph data, while the main objective of multi-relational data mining is to provide principles to mine and/or learn the relational patterns, represented by the expressive logical languages. The former is more geometry oriented and the latter more logic and relation oriented in this paper, the theoretical basis of graph-based data mining is explained in the following section. Second the approaches to graph-based data mining are reviewed and some representative approaches are briefly described.

A. Theoretical Approaches of Graph Based Data Mining

There are five theoretical based approaches of graph-based data mining. They are sub graph categories, sub graph isomorphism, graph invariants, mining measures and solution methods. The sub graphs are categorized into various classes, and the approaches of graph-based data mining strongly depend on the targeted class. Sub graph isomorphism is the mathematical basis of substructure matching and/or counting in graph-based data mining. Graph invariants provide an important mathematical criterion to efficiently reduce the search space of the targeted graph structures in some approaches. Furthermore, the mining measures define the characteristics of the patterns to be mined similarly to conventional data mining. In this paper, the theoretical basis is explained for only undirected graphs without labels but with/without cyclic edges and parallel edges due to space limitations. But, an almost identical discussion applies to directed graphs and/or labeled graphs. Most of the search algorithms used in graph-based data mining come from artificial intelligence, but some extra search algorithms founded in mathematics are also used.

B.Recent Developments Carried Out On Graph Based Data Mining

Researchers have proposed a variety of unsupervised-discovery approaches for structural data. One approach is to use a knowledge base of concepts to classify the structural data. Systems using this approach learn concepts from examples and then categorize observed data. Such systems represent examples as distinct objects and process individual objects one at a time. In contrast, Subdue stores the entire database (with embedded objects) as one graph and processes the graph as a whole. Scientific discovery systems that use domain knowledge have also been developed, but they target a single application domain. An example is Mechem, which relies on domain knowledge to discover chemistry hypotheses. In contrast, Subdue performs general-purpose, automated discovery with or without domain knowledge and hence can be applied to many structural domains. Logic-based systems have dominated relational concept learning, especially inductive logic programming (ILP) systems. However, first-order logic can also be represented as a graph and, in fact, is a subset of what graphs can represent. Therefore, learning systems using graphical representations potentially can learn richer concepts if they can handle the larger hypothesis space. FOIL, the ILP system discussed in this article, executes a

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 73 (2013) 136 – 144

Procedia
Social and Behavioral Sciences

The 2nd International Conference on Integrated Information

Graphical Representation and Exploratory Visualization for Decision Trees in the KDD Process

Mr Wilson A. Castillo Rojas[a], Mr Claudio J. Meneses Villegas[b]*

[a]*Arturo Prat University, Engineering - Area Computing and Informatics, Av. Arturo Prat 2120, Iquique - Postcode: 110-0000, Chile*
[b]*North Catholic University, Systems and Computer Engineering, Av. Angamos 0610 Postcode: 1280, Antofagasta, Chile*

Abstract

This article presents a proposal of representation and scheme of exploratory visualization for Decision Trees in the KDD (*Knowledge Discovery in Database*) process, specifically in the data mining stage. With this, the improvement of the understandability of the internal operation of the model is pursued. This exploratory visualization is based on the well-known technique named Treemap (maps of trees) that allows representing hierarchical structures like the Decision Trees, being used grids to represent the nodes of the Decision Trees. The proposed visualization represents the number of instances or weight associated to a node with a scale of colors in degradation. In this way it is managed to heighten the rules of the Decision Trees in a 2D and 3D graphical representation of this visualization. Finally, a first attempt of subjective evaluation, based on for criteria, of the proposed visualization, is made. In this sense, this work pursues to introduce new schemes of visualizations that allow specifically understand how the data mining models work internally.

Selection and peer-review under responsibility of The 2nd International Conference on Integrated Information

Keywords: Visualization of Decision Trees; Visual Data Mining Models; Schemes of Visualization; Visualization for Data Mining; Data Mining.

1. Introduction

The Knowledge Discovery in Database term (KDD) is defined as "*the nontrivial process of identifying valid patterns, novel, potentially useful and understandable data*" [1]. Therefore, KDD is the overall process of information analysis and knowledge extraction, which covers the stages from selecting the data to analyze, until eventually the end user gets a new knowledge. While data mining (DM) is a typical stage of this process and is

* Corresponding author. Tel.: +56-57-394403 ; fax: +56-57-394472 .
E-mail address: wilson.castillo@unap.cl

1877-0428
Selection and peer-review under responsibility of The 2nd International Conference on Integrated Information
doi:10.1016/j.sbspro.2013.02.033

Journal of Power Sources 196 (2011) 10442–10454

Contents lists available at SciVerse ScienceDirect

Journal of Power Sources

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

Dynamic modeling of electrochemical systems using linear graph theory

Thanh-Son Dao*, John McPhee

Department of Systems Design Engineering, University of Waterloo, 200 University Ave West, Waterloo, Ontario, Canada N2L 3G1

ARTICLE INFO

Article history:
Received 27 June 2011
Received in revised form 14 August 2011
Accepted 15 August 2011
Available online 22 August 2011

Keywords:
Linear graph
Electrochemical cell
NiMH battery simulation
Hybrid electric vehicle

ABSTRACT

An electrochemical cell is a multidisciplinary system which involves complex chemical, electrical, and thermodynamical processes. The primary objective of this paper is to develop a linear graph-theoretical modeling for the dynamic description of electrochemical systems through the representation of the system topologies. After a brief introduction to the topic and a review of linear graphs, an approach to develop linear graphs for electrochemical systems using a circuitry representation is discussed, followed in turn by the use of the branch and chord transformation techniques to generate final dynamic equations governing the system. As an example, the application of linear graph theory to modeling a nickel metal hydride (NiMH) battery will be presented. Results show that not only the number of equations are reduced significantly, but also the linear graph model simulates faster compared to the original lumped parameter model. The approach presented in this paper can be extended to modeling complex systems such as an electric or hybrid electric vehicle where a battery pack is interconnected with other components in many different domains.

1. Introduction

Due to the recent interests in battery electric and hybrid electric vehicles, a significant amount of research has been focused on secondary batteries or electrochemical energy storage devices. For this reason, many of these battery works have been developed as a part of simulation models of these vehicles. These works are sometimes based on empirical relationships, at other times on a detailed description of the physical and chemical processes that take place in the cell [1–4], and even on the development of equivalent circuits [5,6]. Various techniques have been used to develop these models such as lookup tables, lumped parameter models [4], or distributed models using porous electrode theory [1–3].

In this paper, we propose a formalism which, we believe, is more appropriate for the phenomenological description of electrochemical systems which usually consists of complex phenomena across multiple domains; namely the chemical domain, electrical domain, thermal domain, and other domains especially when the battery is placed in a larger system such as a hybrid electric vehicle system. Modeling engineers usually cope with the generation and solution of the equations governing the motion of such systems.

Linear graph theory is a branch of mathematics that studies the manipulation of topology [7,8]. Although this theory has been extensively incorporated into formulation of a wide range of physical systems, namely electrical, mechanical, and hydraulic systems, the extent to which this theory has been applied to modeling electrochemical and thermal processes remains from nil to minimum. It is the goal of this paper to examine this particular problem in some detail. It will be shown in this paper that the electrochemical processes and thermodynamic behaviors of batteries, in general, can be described as equivalent electrical components interconnected to each other, making it possible to use graph theory to develop the dynamic equations for the whole system.

The paper begins with a brief overview of linear graph theory and associated mathematical theorems, followed by a discussion of the applications of linear graphs to modeling electrochemical cells. An example will also be provided to demonstrate the use of linear graphs to model a NiMH battery including the thermal effects and side reactions. Finally are some concluding remarks.

2. Linear graph theory

2.1. Overview

A linear graph representation of a physical system is seen as a collection of oriented line segments called *edges* which intersect only at their *node* points. Although physical systems in different energy domains use different interpretations of nodes and edges, the linear graph topological interpretations of these systems are the same: nodes are the boundaries of a component, while a set of edges represent the component itself. For example, the linear graph for the electrical network given in Fig. 1 can be constructed by drawing a node for each point at which two physical elements connect, and

* Corresponding author. Tel.: +1 519 747 2373; fax: +1 519 746 4791.
E-mail addresses: tdao@maplesoft.com (T.-S. Dao), mcphee@real.uwaterloo.ca (J. McPhee).

0378-7753/$ – see front matter.
doi:10.1016/j.jpowsour.2011.08.065

Extraction of Frequent Association Patterns Co-occurring across Multi-sequence Data

Takahiro Miura and Yoshifumi Okada

***Abstract*— The progress in computer performance and ubiquitous sensor technology has made it possible to yield a large amount of data such as vital data and earthquake data. In general, sensor data are measured from multiple observation points. Extracting correlation or causal association from such huge multi-sequences is a challenging issue in the field of data-mining technology. Development of high-performance sequence mining tools will make great contributions for analyzing various real data such as disease risk prediction or earthquake prediction. In this paper, we propose a new method for extracting sets of patterns (called *association patterns*) that co-occur repeatedly across multiple sequences. Extraction of association patterns is performed by a combination of frequent pattern extraction and interval graph mining. Our method is different from traditional multi-sequence mining methods in that it does not assume similarity of patterns among different sequences. Namely, even if frequent patterns in different sequences show no similarity, our method extracts them as an association pattern if these patterns exhibit a frequent co-occurrence relation along a time-sequence. In this paper, we evaluate the usefulness of our method using synthetic datasets.**

***Index Terms*— frequent pattern, association pattern, interval graph, data mining**

I. Introduction

The progress in computer performance and ubiquitous sensor technology has made it possible to yield a large amount of data such as vital data and earthquake data. In general, sensor data (typically time sequence data) are measured from multiple observation points. Extracting correlation or causal association from such huge multi-sequences is a challenging issue in the field of data-mining technology. Development of high-performance sequence mining tools will make great contributions for analyzing various real data such as disease risk prediction or earthquake prediction.

So far, many data-mining approaches have been proposed to discover frequent patterns (or motifs) for a single sequence data [1]-[3]. Other several research groups suggested methods for discovering similar patterns among multi-sequences and applied them to sensor fault detection or server load balancing [4], [5].

Manuscript received December 30, 2011. This work was supported in part by JUTEN KENKYU PROJECT form Muroran Institute of Technology.

T. Miura is with the Department of Information and Electronic Engineering, Muroran Institute of Technology, 27-1, Mizumoto-cho, Muroran 050-8585, Japan (e-mail: miura@cbrl.csse.muroran-it.ac.jp).

Y. Okada is with College of Information and Systems, Muroran Institute of Technology, 27-1, Mizumoto-cho, Muroran 050-8585, Japan(corresponding author to provide phone: +81-143-5408; fax: +81-143-5408; e-mail: okada@csse.muroran-it.ac.jp)

In this paper, we propose a new method for extracting sets of patterns (called *association patterns*) that co-occur repeatedly across multiple sequences. Extraction of association patterns is performed by a combination of frequent pattern extraction and interval graph mining. Our method is different from traditional multi-sequence mining methods in that it does not assume similarity of patterns among different sequences. Namely, even if frequent patterns in different sequences show no similarity, our method extracts them as an association pattern if these patterns exhibit a frequent co-occurrence relation along a time-sequence. In this paper, we evaluate the usefulness of our method using synthetic datasets.

This paper is organized as follows. Section II describes the basic idea of the method. Section III explains the procedure of our method. Section IV and V show the experimental method and the results, respectively. Finally, Section VI summarizes our conclusions and suggests future work.

II. The Basic Idea of Our Method

Fig. 1(a) summarizes traditional pattern mining for a single sequence [1]-[3] and for multiple sequences [4], [5]. Methods for a single sequence extract patterns that occur repeatedly in a sequence. On the other hand, methods for multiple sequences discover patterns that are similar or common among different sequences.

In contrast to those traditional approaches, we aim at discovering sets of patterns that co-occur repeatedly across multiple sequences, as shown in Fig.1b. Respective patterns in a set do not necessarily need to be identical. In this paper, such set of patterns is referred to as an *association pattern*.
We first apply traditional frequent pattern mining to each sequence and subsequently employ a concept of interval graph [3] to extract association pattern. An interval graph is a graph for representing a set of intervals as depicted in Fig. 2(a) and Fig. 2(b) is the interval graph for Fig. 2(a), in which a node indicates an interval and an edge means that two intervals overlap. In this study, a frequent pattern in each sequence is regarded as a node, and an overlap of any two frequent patterns is considered as an edge. Extraction of association patterns can be achieved by finding connected graphs from the interval graph generated above.

III. Method

Fig. 3 illustrates the procedure of the method. This method consists of two steps: (a) frequent pattern extraction from each sequence and (b) association pattern extraction from the interval graph. These steps are performed after preprocessing for each sequence.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Inducing Decision Trees with an Ant Colony Optimization Algorithm

Fernando E. B. Otero*, Alex A. Freitas, Colin G. Johnson

School of Computing, University of Kent, UK

Abstract

Decision trees have been widely used in data mining and machine learning as a comprehensible knowledge representation. While ant colony optimization (ACO) algorithms have been successfully applied to extract classification rules, decision tree induction with ACO algorithms remains an almost unexplored research area. In this paper we propose a novel ACO algorithm to induce decision trees, combining commonly used strategies from both traditional decision tree induction algorithms and ACO. The proposed algorithm is compared against three decision tree induction algorithms, namely C4.5, CART and *c*ACDT, in 22 publicly available data sets. The results show that the predictive accuracy of the proposed algorithm is statistically significantly higher than the accuracy of both C4.5 and CART, which are well-known conventional algorithms for decision tree induction, and the accuracy of the ACO-based *c*ACDT decision tree algorithm.

Keywords: ant colony optimization, data mining, classification, decision tree

1. Introduction

One of the most studied data mining tasks in the literature is the classification task [15, 29]. In essence, the classification task consists of learning a predictive relationship between input values and a desired output. Each example (data instance or record) is described by a set of features (attributes)—referred to as predictor attributes—and a class attribute. Given a set of examples, a classification algorithm aims at creating a model, which represents the relationship between predictor attributes values and class values (labels), and which is able to predict the class label of a new (unseen) example based on the values of its predictor attributes.

Classification problems can be viewed as optimisation problems, where the goal is to find the best function (model) that represents the predictive relationships in the data. A classification problem can be formally specified as:

*Corresponding author

Email addresses: F.E.B.Otero@kent.ac.uk (Fernando E. B. Otero), A.A.Freitas@kent.ac.uk (Alex A. Freitas), C.G.Johnson@kent.ac.uk (Colin G. Johnson)

Efficient Algorithm for Mining Frequent Subgraphs (Static and Dynamic) based on gSpan

K.Lakshmi
Asst.Prof, Departmentof MCA,
Sir M.Visvesvaraya Institute of Technology,
Bangalore.

T.Meyyappan, PhD.
Professor, Department of Computer Science & Engineering,
Alagappa University, Karaikudi.

ABSTRACT
Frequent sub graph mining is another active research topic in data mining. A graph is a general model to represent data and has been used in many domains like chemo informatics and bioinformatics. Mining patterns from graph databases is challenging since graph related operations, such as sub graph testing, generally have higher time complexity than the corresponding operations on item sets, sequences, and trees.

We investigated new approaches for frequent graph-based pattern mining in graph datasets and found that a novel algorithm called span (graph-based Substructure pattern mining), has been used as the standard for comparing performance of new algorithms. It is based on the pattern growth approach of frequent sub graph mining and hence discovers frequent substructures without candidate generation. span is based on a lexicographic order, and maps each graph to a unique minimum DFS code as its canonical label. Based on this lexicographic order, gSpan adopts the depth-first search strategy to mine frequent connected sub graphs efficiently. Based on the literature survey done, algorithms based on pattern growth approach and DFS strategy are found to be better in performance than algorithms based on Apriori approach and BFS strategy. In this paper we propose a new algorithm based on gSpan, for a special class of graphs characterised by the existence of unique node lablels.

General Terms
Graph mining, subgraph, Lexicographic ordering, labeled graphs.

Keywords
Parallel programming, frequent subgraph mining, DFS code, Isomorphism.

1. INTRODUCTION
Graph mining has become a very active area of research. Frequent subgraph mining refers to the problem of subgraph isomorphism detection. In sub graph isomorphism detection, a mapping f from the nodes of one given graph g1 to the nodes of another given graph g2 is a bijection that preserves all edges and labels. The main problem with this sub graph isomorphism is its high computational complexity. Also it is a known fact that it is NP-complete. Many heuristics have been developed to speed up sub graph isomorphism by using special canonical labeling of the graphs; none of them, however, can avoid an exponential worst-case computation time.

Contribution. In this paper, we propose a new algorithm based on gSpan. It can mine frequent sub graph from a special kind of graphs, characterized by nodes with unique node labels. It also targets to reduce the time complexity, using parallel programming. If the entire graph dataset can fit in main memory, the proposed method can be applied directly; To the best of our knowledge, the two techniques, DFS lexicographic order and minimum DFS code, introduced in gSpan are the best , which form, a novel canonical labeling system, to support DFS search . But still the problem of finding minimum DFS code used in gSpan is also NP- complete. The proposed algorithm addresses this issue by using a modified DFS representation. It retains all the advantages of gSpan, while taking advantage of the multi core processing technology by using the concept of parallel programming to improve the performance of the algorithm. Number of duplicate graphs generated may be comparatively little more than gSpan algorithm as mining of sub graphs from frequent single edge graphs are done in parallel.

The remainder of the paper is organised as follows. In Sect. 2, we introduce our basic concepts and terminology. Graphs with unique node labels and their possible representations are discussed in Sect.3. Proposed graph mining algorithm based on gSpan and its time complexity is discussed in Sect. 4. In Sect. 5, we discuss the possible applications of the graphs with unique labels. Theoretical conclusions from the proposed work are discussed in Sect. 6.

2. BASIC CONCEPTS AND NOTATIONS
In this section, the basic concepts and terminology used is introduced. In this paper we focus on a special class of undirected labelled simple graphs, graphs with unique no labels. For any graph G and any pair of vertices x,y., the condition L(x) <> L(y) holds if x <>y. Throughout the rest of this paper we consider graphs from this class only.

2.1 Definition1
A Labeled graph can be represented by a 3-tuple G=(V,E,L) where V is a set of vertices, E is a set of edges, L is a set of labels for the vertices. For simplicity we assume the labels for all the edges to be empty. According to this definition a Labelled graph can be represented by a 3-tuple G=(V,E,L) where V is a set of vertices, E is a set of edges, L is a set of labels.

2.2 Definition2
Given a graph G = (V, E), a graph Gs = (Vs , Es) is a subgraph of G if Vs ⊆ V and Es ⊆ E, and is denoted by Gs ⊆ G. Two graphs G1 = (V1, E1) and G2 = (V2, E2) are isomorphic, if they are topologically identical to each other, that is, there is a vertex mapping from V1 to V2 such that each edge in E1 is mapped to a single edge in E_2 and vice versa. In the case of labeled graphs, this mapping must also preserve the labels on the vertices and edges. When a set of graphs {Gi } are isomorphic to each other, they all are said to belong to the same equivalence class.

Journal of Informetrics 4 (2010) 291–312

Contents lists available at ScienceDirect

Journal of Informetrics

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

Graph-based data mining: A new tool for the analysis and comparison of scientific domains represented as scientograms

Arnaud Quirin [a], Oscar Cordón [a,*], Benjamín Vargas-Quesada [b,c], Félix de Moya-Anegón [d]

[a] European Centre for Soft Computing, Edf. Científico Tecnológico, 33600 Mieres, Spain
[b] Communication and Information Science Faculty, University of Granada, 18071 Granada, Spain
[c] CSIC, Unidad Asociada Grupo Scimago, 18071 Granada, Spain
[d] CSIC/CCHS/IPP, 28037 Madrid, Spain

ARTICLE INFO

Article history:
Received 14 October 2009
Received in revised form
25 December 2009
Accepted 20 January 2010

Keywords:
Domain analysis
Social networks
Scientograms
Graph-based data mining
Scientogram mining
Subdue algorithm

ABSTRACT

The creation of some kind of representations depicting the current state of Science (or *scientograms*) is an established and beaten track for many years now. However, if we are concerned with the automatic comparison, analysis and understanding of a set of scientograms, showing for instance the evolution of a scientific domain or a face-to-face comparison of several countries, the task is titanically complex as the amount of data to analyze becomes huge and complex. In this paper, we aim to show that graph-based data mining tools are useful to deal with scientogram analysis. Subdue, the first algorithm proposed in the graph mining area, has been chosen for this purpose. This algorithm has been customized to deal with three different scientogram analysis tasks regarding the evolution of a scientific domain over time, the extraction of the common research categories substructures in the world, and the comparison of scientific domains between different countries. The outcomes obtained in the developed experiments have clearly demonstrated the potential of graph mining tools in scientogram analysis.

1. Introduction

The construction of a great map of sciences[1] has been a persistent idea in the modern ages. This need arises from the general conviction that an image or graphic representation of a domain favors and facilitates its comprehension and analysis. The visualization of scientific information has long been used to uncover and divulge the essence and structure of science (Börner & Scharnhorst, 2009; Chen, 1999a, 2004). Yet despite its ripe age, information display is still in an adolescent stage of evolution in the context of its application to scientific domain analysis. Never before data have been generated at such high volumes as it is today. Exploring and analyzing the vast volumes of scientific data is becoming increasingly difficult. There is a large number of information visualization techniques which have been developed over the last decade within this area (Chen, 1999b; Lucio-Arias & Leydesdorff, 2008; Moya-Anegón et al., 2007, 2005; Small & Garfield, 1985), but none of them has been designed to support the exploration of large datasets. Besides, all the latter approaches require a large amount of expertise from the user, which reduces the chances to automate the analysis procedure. Nevertheless, it is clear that information visualization and visual data mining (Keim, 2002) can provide the theoretical and practical backgrounds to deal with scientific information analysis.

* Corresponding author. Tel.: +34 985 456545; fax: +34 985 456699.
E-mail addresses: arnaud.quirin@softcomputing.es (A. Quirin), oscar.cordon@softcomputing.es (O. Cordón), benjamin@ugr.es (B. Vargas-Quesada), felix_demoya@cchs.csic.es (F. de Moya-Anegón).

[1] In the following we will consider (*visual*) *science maps*, *scientograms*, *graphs*, or simply *maps* as synonyms within our domain.

1751-1577/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.joi.2010.01.004

Discrete Applied Mathematics 160 (2012) 1416–1428

Contents lists available at SciVerse ScienceDirect

Discrete Applied Mathematics

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

Inferring a graph from path frequency☆

Tatsuya Akutsu [a,*], Daiji Fukagawa [b], Jesper Jansson [c], Kunihiko Sadakane [d]

[a] Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Kyoto 611-0011, Japan
[b] Faculty of Culture and Information Science, Doshisha University, Kyoto 610-0394, Japan
[c] Ochanomizu University, Tokyo 112-8610, Japan
[d] National Institute of Informatics, Tokyo 101-8430, Japan

ARTICLE INFO

Article history:
Received 11 September 2009
Received in revised form 12 November 2011
Accepted 7 February 2012
Available online 3 March 2012

Keywords:
Kernel method
Graph algorithms
Feature vector
Pre-image

ABSTRACT

This paper considers the problem of inferring a graph from the number of occurrences of vertex-labeled paths, which is closely related to the pre-image problem for graphs: to reconstruct a graph from its feature space representation. It is shown that both exact and approximate versions of the problem can be solved in polynomial time in the size of an output graph by using dynamic programming algorithms if the graphs are trees whose maximum degree is bounded by a constant and the lengths of given paths and alphabet size are bounded by constants. On the other hand, it is shown that this problem is strongly NP-hard even for trees of bounded degree if the maximum length of paths is not bounded. The problem of inferring a string from the number of occurrences of fixed size substrings is also studied.

1. Introduction

Kernel methods have become a standard tool in machine learning and have been applied to various areas [8,24,25], which include bioinformatics and chemoinformatics. In order to apply kernel methods to target problems, it is usually required to develop a mapping from the set of objects in the target problem to a *feature space* (i.e., each object is transformed to a vector of reals) and a kernel function is defined as an inner product between two *feature vectors*. For instance, in the *spectrum kernel* method [17], each biological sequence is mapped to a frequency vector of fixed length substrings (i.e. frequency of n-grams). In some cases, a feature space can be an infinite dimensional space (Hilbert space) and the kernel trick is applied for efficient computation of the value of a kernel function without explicitly computing feature vectors [25].

Recently, a new approach has been proposed for designing and/or optimizing objects using kernel methods [4,5]. In this approach, a desired object is computed as a point in the feature space using suitable objective function and optimization technique and then the point is mapped back to the input space, where this mapped back object is called a *pre-image*. Let ϕ be a mapping from an input space $\mathcal{G}$ to a feature space $\mathcal{F}$. Then, the problem is, given a point y in $\mathcal{F}$, to find a pre-image x in $\mathcal{G}$ such that $y = \phi(x)$. It should be noted that ϕ is not necessarily injective or surjective. If ϕ is not surjective, we should compute the approximate pre-image x^* for which the distance between y and $\phi(x)$ is minimized (see Fig. 1):

$$x^* = \arg \min_x \mathrm{dist}(y, \phi(x)).$$

Bakir et al. proposed a method to find pre-images in a general setting by using Kernel Principal Component Analysis and regression [4]. Bakir et al. developed a stochastic search algorithm to find pre-images for graphs [5]. It should be noted that

☆ A preliminary version of this paper [1] appeared in Proc. 16th Annual Symposium on Combinatorial Pattern Matching, 2005.

* Corresponding author. Tel.: +81 774 38 3015; fax: +81 774 38 3022.
E-mail addresses: takutsu@kuicr.kyoto-u.ac.jp (T. Akutsu), dfukagaw@mail.doshisha.ac.jp (D. Fukagawa), Jesper.Jansson@ocha.ac.jp (J. Jansson), sada@nii.ac.jp (K. Sadakane).

0166-218X/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2012.02.002

Knowledge and Information Systems manuscript No. KAIS-2416R1

MOSubdue: A Pareto Dominance-based Multiobjective Subdue Algorithm For Frequent Subgraph Mining

Prakash Shelokar · Arnaud Quirin · Óscar Cordón

Received: Mar 23, 2010 / Revised: Feb 16, 2011 / Accepted: Oct 04, 2011

Abstract Graph-based data mining approaches have been mainly proposed to the task popularly known as frequent subgraph mining subject to a single user preference, like frequency, size, etc. In this work, we propose to deal with the frequent subgraph mining problem from multiobjective optimization viewpoint, where a subgraph (or solution) is defined by several user-defined preferences (or objectives), which are conflicting in nature. For example, mined subgraphs with high frequency are often of small size, and *vice-versa*. Use of such objectives in the multiobjective subgraph mining process generates Pareto-optimal subgraphs, where no subgraph is better than another subgraph in all objectives. We have applied a *Pareto-dominance approach* for evaluation and search subgraphs regarding to both proximity and diversity in multiobjective sense, which has incorporated in the framework of Subdue algorithm for subgraph mining. The method is called Multi-Objective subgraph mining by Subdue (MOSubdue), and has several advantages: i) generation of Pareto-optimal subgraphs in a single run, ii) selection of subgraph-seeds from the candidate subgraphs based on all objectives, iii) search in the multiobjective subgraphs lattice space, and iv) capability to deal with different multiobjective frequent subgraph mining tasks by customizing the tackled objectives. The good performance of MOSubdue is shown by performing multiobjective subgraph mining defined by two and three objectives on two real-life datasets.

Keywords Graph-based data mining · Frequent subgraph mining · Subdue · Gaston · Multiobjective graph-based data mining · Pareto-based multiobjective optimization · Evolutionary multiobjective optimization

1 Introduction

Graph-based data mining (GBDM) has been prevalently used in a wide range of application domains, such as computing communities [11, 31], subgraph discovery [7, 41, 48, 51], topic

Prakash Shelokar · Arnaud Quirin · Óscar Cordón
European Centre for Soft Computing, 33600-Mieres, Spain.
Dr. Óscar Cordón is also affiliated to the Department of Computer Science and Artificial Intelligence (DECSAI) and the Research Centre on Information and Communication Technologies (CITIC-UGR), University of Granada. 18071-Granada, Spain.
E-mail: {prakash.shelokar, arnaud.quirin, oscar.cordon}@softcomputing.es, ocordon@decsai.ugr.es

Object Categorization using Bone Graphs

Diego Macrini, Sven Dickinson, David Fleet, Kaleem Siddiqi

Abstract

The bone graph [23, 25] is a graph-based medial shape abstraction that offers improved stability over shock graphs and other skeleton-based descriptions that retain unstable ligature structure. Unlike the shock graph, the bone graph's edges are attributed, allowing a richer specification of relational information, including how and where two medial parts meet. In this paper, we propose a novel shape matching algorithm that exploits this relational information. Formulating the problem as an inexact directed acyclic graph matching problem, we extend a leading bipartite graph-based algorithm for matching shock graphs [41]. In addition to accommodating the relational information, our new algorithm is better able to enforce hierarchical and sibling constraints between nodes, resulting in a more general and more powerful matching algorithm. We evaluate our algorithm with respect to a competing shock graph-based matching algorithm, and show that for the task of view-based object categorization, our algorithm applied to bone graphs outperforms the competing algorithm. Moreover, our algorithm applied to shock graphs also outperforms the competing shock graph matching algorithm, demonstrating the generality and improved performance of our matching algorithm.

Key words: Medial Shape Representation, Graph-Based Shape Representation, Inexact Graph Matching, Object Categorization

1. Introduction

The recognition of 3-D objects from their silhouettes demands a shape representation which is invariant to minor changes in viewpoint and articulation. This invariance can be achieved by parsing a silhouette into parts and relationships that are stable across similar object views. Medial descriptions, such as skeletons and shock graphs, attempt to decompose a shape into

Graphs and Combinatorics (2012) 28:57–75
DOI 10.1007/s00373-011-1090-6

SURVEY

How Many Conjectures Can You Stand? A Survey

H. J. Broersma · Z. Ryjáček · P. Vrána

Received: 1 September 2011 / Revised: 17 September 2011 / Published online: 9 October 2011

Abstract We survey results and open problems in hamiltonian graph theory centered around two conjectures of the 1980s that are still open: every 4-connected claw-free graph (line graph) is hamiltonian. These conjectures have lead to a wealth of interesting concepts, techniques, results and equivalent conjectures.

Keywords Hamiltonian graph · Hamilton-connected · Claw-free graph · Line graph · Cubic graph · Dominating closed trail · Dominating cycle · Collapsible graph · Supereulerian graph · Snark · Cyclically 4-edge-connected · Essentially 4-edge-connected · Closure · Contractible graph

Mathematics Subject Classification (2000) 05C45 · 05C38 · 05C35

1 Introduction

Before we are going to introduce the necessary terminology for understanding the sequel, let us start by presenting the two conjectures that will play the main role throughout our exposition.

Zdeněk Ryjáček and Petr Vrána were supported by project 1M0545 and Research Plan MSM 4977751301 of the Czech Ministry of Education.

H. J. Broersma (✉)
Faculty of Electrical Engineering, Mathematics and Computer Science,
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
e-mail: h.j.broersma@utwente.nl

Z. Ryjáček · P. Vrána
Department of Mathematics, University of West Bohemia, Univerzitní 8, 306 14 Pilsen, Czech Republic
e-mail: ryjacek@kma.zcu.cz

Z. Ryjáček · P. Vrána
Institute for Theoretical Computer Science, Charles University, Univerzitní 8,
306 14 Pilsen, Czech Republic
e-mail: vranap@kma.zcu.cz

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 3 (2012) 304 – 315

Procedia Technology

The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science

A Graph-based Method to Solve the Economical Dispatch Problem Disregarding Slack Variables

Jaime Cerda Jacobo[a], Nancy P. Cira Perez[a,1], Juan J. Flores Romero[a]

[a]*Graduate Division, School of Electrical Engineering, University of Michoacan, Morelia, Michoacan, Mexico 58000*

Abstract

One of the greatest challenges to confront Nonlinear Programming Problems, it is the selection of the active and non active set of constraints of the system. For this reason many optimization applications prefer to use barrier or penalty methods with their related inefficiencies. This paper describes a graph-based solution for these models which facilitates the handling of such constraints and, therefore, the solution process for the model. To this end some parts of the graph are considered active or non active, depending on the actual model solution as well as the values of the Lagrange multipliers. At every solution step, there will probably be some changes on the graph topology to reflect the current conditions of the problem whose solution is in progress. These solutions besides being efficient, provide an optimal storage scheme as only the fundamental information of the problem is stored.

Keywords: Nonlinear programming, Economic dispatch, Dispersity.

Nomenclature

$\mathbb{G}$	The set of generators
Q	The inelastic compound load
λ	The energy price
z_g	NLP desicion variable
$\overline{\rho_g}$	Dual variable associted to z_g uppper bound
$\underline{\rho_g}$	Dual variable associted to z_g lower bound
$\lceil z_g \rceil$	The upper limit value of variable z_g
$\lfloor z_g \rfloor$	The lower limit value of variable z_g
Δz_g	The variable incrementz_g

1. Introduction

One of the greatest challenges to confront Nonlinear Programming Problems is the selection of the active and non active set of constraints of the system. For this reason many optimization applications prefer to use

Email addresses: jcerda@umich.mx (Jaime Cerda Jacobo), npaolacp@hotmail.com (Nancy P. Cira Perez), juanf@umich.mx (Juan J. Flores Romero)

[1]Supported by CONACYT

2212-0173 © 2012 Published by Elsevier Ltd.
doi:10.1016/j.protcy.2012.03.033

Information Sciences xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Graph theory based model for learning path recommendation

Guillaume Durand *, Nabil Belacel, François LaPlante

National Research Council Canada, Institute for Information Technology, Moncton, NB, Canada

ARTICLE INFO

Article history:
Received 1 December 2011
Received in revised form 8 February 2013
Accepted 12 April 2013
Available online xxxx

Keywords:
Learning object recommendation system
Learning path
Graph theory
Soft computing

ABSTRACT

Learning design, the activity of designing a learning path, can be a complex task, especially for learners. A learning design recommendation system would help self-learners find appropriate learning objects and build efficient learning paths during their learning journey. Educational Data Mining (EDM) has provided an impressive amount of novelties related to learning object recommendation systems. However, most of the solutions proposed thus far do not take into account eventual competency dependencies among learning objects and/or are not designed for large repositories of interdependent learning objects. We propose a model to build a learning design recommendation system based on graph theory. From this model, we propose, implement and test an approach using the concept of cliques to recommend learning paths.

1. Introduction

For many years, learning design has been the subject of a significant amount of research papers and implementations. Learning design [8,12] consists in the process of designing learning activities. Usually, this task is thought to be assumed by a teacher preparing a formal learning session in his/her favorite Virtual Learning Environment (VLE). In the emerging Personal Learning Environments (PLEs [26]), the teacher's role has evolved since learners are expected to be more pro-active in the learning process. It is up to learners to choose learning material and organize their custom-fit learning process; the learner is the learning designer.

Many approaches have emerged in the last few years to facilitate learners' task of finding learning materials. Among those figures the contribution of data mining and recommendation technologies to provide learners with adequate learning materials. By recommending educational papers [35] or internet links [16], authors use collaborative and/or content-based filtering approaches for their recommendation systems. There are two ways of building such a recommendation system. Usually, some cluster profiles are discovered among a learner or content population using unsupervised learning (clustering) and from these new clusters, learners or content are predicted or classified by supervised learning methods (classifiers). Hence, depending on the classification, learners will be recommended with the resources completed by other learners sharing the same cluster and/or with content similar to those they themselves have already completed.

These approaches tend to satisfy the immediate interest of learners rather than considering the recommendation of content fitting in an organized sequential learning process. This approach might be problematic since learning design puts a lot of emphasis on the sequence of the learning material. More precisely, the sequence of the learning material has to match a particular learning process strategy in which each learning object (LO) is consistent with the former and the next one as well as the evolving learner knowledge [11]. To some extent, learning design tends to think of learning with more guidance, considering it as a journey among activities where the proposed itinerary makes as much sense as the learning materials offered. A learning design recommendation system should propose a sequence of learning objects in a well-defined order

* Corresponding author. Tel.: +1 506 861 0961; fax: +1 506 851 3630.
E-mail addresses: Guillaume.Durand@nrc.gc.ca (G. Durand), Nabil.Belacel@nrc.gc.ca (N. Belacel), Francois.Laplante@nrc.gc.ca (F. LaPlante).

0020-0255/$ - see front matter Crown Copyright © 2013 Published by Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.04.017

Journal of Biomedical Informatics 44 (2011) 1086–1092

Contents lists available at SciVerse ScienceDirect

Journal of Biomedical Informatics

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

Neighborhood hash graph kernel for protein–protein interaction extraction

Yijia Zhang *, Hongfei Lin, Zhihao Yang, Yanpeng Li

School of Electronics and Information Engineering, Dalian University of Technology, Dalian, Liaoning 116023, China

ARTICLE INFO

Article history:
Received 15 December 2010
Accepted 17 August 2011
Available online 23 August 2011

Keywords:
Interaction extraction
Hash
Graph kernel
Biomedical literature

ABSTRACT

Automated extraction of protein–protein interactions (PPIs) from biomedical literatures is an important topic of biomedical text mining. In this paper, we propose an approach based on neighborhood hash graph kernel for this task. In contrast to the existing graph kernel-based approaches for PPI extraction, the proposed approach not only has the capability to make use of full dependency graphs to represent the sentence structure but also effectively control the computational complexity. We evaluate the proposed approach on five publicly available PPI corpora and perform detailed comparisons with other approaches. The experimental result shows that our approach is comparable to the state-of-the-art PPI extraction system and much faster than all-path graph kernel approach on all five PPI corpora.

1. Introduction

With the exponential explosion of biomedical literature, information extraction from biomedical literature has been a topic of intense research. Automated protein–protein interaction (PPI) extraction from biomedical literature is an important task in biomedical text mining, which contributes to PPI network analysis and discovery of new functions of proteins. A lot of research interests [1–10] have been reported for this task during recent years. The existing PPI methodologies rely on different approaches broadly divided into three main categories: co-occurrences engineering, pattern engineering and machine learning. With the gradual maturity of the kernel method, machine learning based approach has got the advantage of the performance over the others. In particular, the recent studies [6–10] show most state-of-the-art systems are in the framework of machine learning.

Machine learning based approach for PPI extraction usually tackles the task as a classification problem. A major challenge is how to supply the learner with the semantic/syntactic information needed to distinguish between interactions and non-interactions [6]. Therefore, kernel methods are required by PPI extraction system, which can learn rich structural data such as syntactic parse tree or dependency graph.

However, the existing kernel methods exploit only limited information of the syntactic parse tree or dependency graph and are still computationally expensive. The walk-weighted subsequence kernels [7] match the e-walk and v-walk on the shortest path of the full dependency graph, which can only represent the semantic/syntactic information of the shortest path. The tree kernels [1] can represent more complex structures, but the tree representations are still not enough to completely represent all semantic/syntactic information of the dependency graph. All-paths graph kernel [6] maps the dependency graph into the label pairs feature space, but it cannot match the label sequence on every path of dependency graph. Furthermore, these methods are computationally expensive, particularly when computing the complex dependency graphs.

In this paper, we proposed a neighborhood hash kernel based method for PPI extraction. The framework of neighborhood hash kernel is proposed by Hido [11]. To the best of our knowledge, we first apply the neighborhood hash kernel to the task of PPI extraction. Firstly, we use a mapping function to transform each node label of dependency graph into a bit label which is represented as a binary array of fixed length. Secondly, we replace the bit label of node by a new bit label produced by order-independent logical operations on the bit labels of the node and the neighboring nodes. Updating the node label with the new bit label by the specific logical operation, such as *XOR*, allow us to combine the neighborhood structure into the updated label. We apply this procedure to all of the nodes in dependency graph to exchange the semantic/syntactic information between connected nodes, and repeat several times to propagate the features of the high order substructures over the dependency graph. Finally, we can efficiently compute the similarity of the two dependency graphs based on the intersection ratio of the updated label sets. As the neighborhood hash kernel can calculate the full dependency graphs, our method can reduce the risk of missing important features. Furthermore, the neighborhood hash kernel is a linear time and the overall computational complexity of our method is only $O(n^2 + DR\bar{d}n)$ which is significantly lower than other kernel methods.

* Corresponding author.
E-mail address: zhyj@dlut.edu.cn (Y. Zhang).

1532-0464/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2011.08.011

Discrete Mathematics 312 (2012) 157–173

Contents lists available at SciVerse ScienceDirect

Discrete Mathematics

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

Context-free pairs of groups II — Cuts, tree sets, and random walks

Wolfgang Woess

Institut für Mathematische Strukturtheorie, Technische Universität Graz, Steyrergasse 30, 8010 Graz, Austria

ARTICLE INFO

Article history:
Available online 24 August 2011

Dedicated to Gert Sabidussi on the occasion of his 80th birthday

Keywords:
Finitely generated pair of groups
Context-free grammar
Context-free graph
Cut
Tree set
Random walk

ABSTRACT

This is a continuation of the study, begun by Ceccherini-Silberstein and Woess (2009) [5], of context-free pairs of groups and the related context-free graphs in the sense of Muller and Schupp (1985) [22]. The graphs under consideration are Schreier graphs of a subgroup of some finitely generated group, and context-freeness relates to a tree-like structure of those graphs. Instead of the cones of Muller and Schupp (1985) [22] (connected components resulting from deletion of finite balls with respect to the graph metric), a more general approach to context-free graphs is proposed via tree sets consisting of cuts of the graph, and associated structure trees. The existence of tree sets with certain "good" properties is studied. With a tree set, a natural context-free grammar is associated. These investigations of the structure of context free pairs, resp. graphs are then applied to study random walk asymptotics via complex analysis. In particular, a complete proof of the local limit theorem for return probabilities on any virtually free group is given, as well as on Schreier graphs of a finitely generated subgoup of a free group. This extends, respectively completes, the significant work of Lalley (1993, 2001) [18,20].

1. Introduction and preliminaries

This is a direct continuation of the paper of Ceccherini and Woess [5] but is to a very large extent independent of that reference.

The interplay between combinatorial group theory and formal languages is a very natural one and has implicitly or explicitly been present ever since the beginning of the study of finitely generated groups. Regarding the specific case of context-free languages, a very satisfactory theory and results were provided in the seminal work of Muller and Schupp [21,22]. For several further references, see [5], where we have undertaken a study of context-freeness in the situation of a pair (G, K), where G is a finitely generated group and K is a subgroup. The situation is best understood via Schreier graphs of such a pair, whose context-freeness is equivalent with a specific property of tree-likeness of such labelled graphs [22]. In the present paper, that notion of tree-likeness is refined and generalised. Briefly spoken, in the approach of [22] one considers *cones* in a labelled graph. These are the connected components that remain after removing any ball (with respect to the natural graph metric) around a given "root" vertex, and the graph is called context-free if there are finitely many isomorphism types of such cones as labelled graphs with finite boundaries.

In the present work, the cones are replaced by more general *cuts* (connected subgraphs with finite boundaries). We want to have a collection ("tree set") of such cuts which contains again only finitely many isomorphism classes as above, and fills up the whole graph in a uniform way. With such a tree set, we can associate a context-free grammar that generates the *word problem*, that is, the set of all words that can be read along some closed path that starts and ends at the root.

This is elaborated in detail and then applied to study the asymptotics of random walk return probabilities on such Schreier graphs in the case when the tree set has certain additional "good" properties. In this regard, one of the main aims is to derive

E-mail address: woess@TUGraz.at.

0012-365X/$ – see front matter
doi:10.1016/j.disc.2011.07.026

Computer Communications xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Computer Communications

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

Random walk with jumps in large-scale random geometric graphs ☆

Leonidas Tzevelekas [a,*], Konstantinos Oikonomou [b], Ioannis Stavrakakis [a]

[a] National and Kapodistrian University of Athens, Department of Informatics and Telecommunications, Athens, Greece
[b] Ionian University, Department of Informatics, Corfu, Greece

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Random walk agent
Jumping Random Walk agent
Cover
Partial cover
Random geometric graphs

ABSTRACT

The *information dissemination* problem in large-scale networking environments like wireless sensor networks and ad hoc networks is studied here considering random geometric graphs and random walk based approaches. A new type of random walk based agent is proposed in this paper and an analytical expression with respect to *coverage* (i.e., the proportion of the network nodes visited by the random walk agent) as a function of the number of the agent movements is derived. It is observed that the cover time of many of already existing random walk based variants is large in random geometric graphs of low degree (as it is commonly the case is wireless environments). As this inefficiency is attributed (as discussed in the paper) to the inability of existing random walk based solutions to move away from already likely covered areas, a mechanism for directional movement (i.e., jumping) of the random walk based agent is proposed and studied, that allows the agent to jump to different network areas, most likely not covered yet. The proposed mechanism (Jumping Random Walk) is studied analytically and via simulations and the parameters (of the network topology and the mechanism) under which the proposed scheme outperforms existing random walk based variations are determined.

1. Introduction

One of the main challenges associated with large-scale, unstructured and dynamic networking environments is that of *efficiently reaching out* to *all* or *a portion* of the network nodes (i.e., *disseminating information*) in order to provide, e.g., software updates or announcements of new services or queries. The high dynamicity and the sheer size of such networking topologies ask for the adoption of decentralized approaches to information dissemination [1–4]. In this paper, the problem of efficiently disseminating information (or queries) across a large-scale, resource-limited, ad hoc-structured wireless network, such as a wireless sensor network, is considered. One of the simplest approaches employed for disseminating information in such environments, is the traditional *flooding* approach. Under flooding [5–8], each time a node receives a message for the first time from some node, it forwards it to all its neighbors except from that node. Despite its simplicity and speed (typically achieving the shortest cover time, upper bounded by the network diameter), the associated large message overhead is a major drawback.

As flooding is considered not to be an option for large-scale, wireless sensor networks (WSNs) due to strict energy limitations of individual sensor nodes, solutions based on variations of the random walk based information dissemination paradigm are viewed as reasonable choices for searching and/or routing in WSNs [9–11]. Furthermore, there has been a significant body of work in adopting random walks for search or information dissemination in large peer-to-peer (P2P) networks [12–15]. The random walk based information dissemination paradigm possesses several good characteristics such as simplicity, robustness against dynamic failures or changes to the network topology, and lack of need for knowledge of the network physical and topological characteristics. A *Random Walk agent* (RW-agent) doing a simple random walk within a network of wireless sensors moves from neighbor node to neighbor node in a purely random manner, frequently revisiting previously covered nodes in a circular manner. Even when backtracking (returning to the node it just came from) is not allowed, some circular movement in the topology can not be eliminated; these revisits constitute overhead and impact negatively on the cover time [16] of the process. Such a poor behavior of the RW-agent is attributed to the random manner of its movement, combined with some problematic topological characteristics of large-scale wireless ad hoc networks, such as cliques and bottlenecks. To make sure that there is consistency in terminology in this paper, we note here

☆ This work has been supported in part by the project ANA (Autonomic Network Architecture) (IST-27489), the PENED 2003 program of the General Secretariat for Research and Technology (GSRT) co-financed by the European Social Funds (75%) and by National Sources (25%) and the NoE CONTENT (IST-384239).

* Corresponding author. Address: National and Kapodistrian University of Athens, Department of Informatics and Telecommunications, Ilissia, 15 784 Athens, Greece. Tel.: +30 210 727 5341; fax: +30 210 727 5333.
E-mail addresses: ltzev@di.uoa.gr (L. Tzevelekas), okon@ionio.gr (K. Oikonomou), ioannis@di.uoa.gr (I. Stavrakakis).

0140-3664/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.01.026

Journal of Computer and System Sciences ••• (••••) •••–•••

Contents lists available at SciVerse ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Three-objective subgraph mining using multiobjective evolutionary programming

Prakash Shelokar [a], Arnaud Quirin [a], Óscar Cordón [a,b,*]

[a] European Centre for Soft Computing, 33600-Mieres, Spain
[b] Department of Computer Science and Artificial Intelligence (DECSAI) and Research Centre on Information and Communication Technologies (CITIC-UGR), University of Granada, 18071-Granada, Spain

ARTICLE INFO

Article history:
Received 1 August 2012
Received in revised form 16 November 2012
Accepted 14 March 2013
Available online xxxx

Keywords:
Graph-based data mining
Frequent subgraph mining
Multiobjective optimization
Multiobjective graph mining
Multiobjective evolutionary programming
Subdue

ABSTRACT

The existing methods for graph-based data mining (GBDM) follow the basic approach of applying a single-objective search with a user-defined threshold to discover interesting subgraphs. This obliges the user to deal with simple thresholds and impedes her/him from evaluating the mined subgraphs by defining different "goodness" (i.e., multiobjective) criteria regarding the characteristics of the subgraphs. In previous papers, we defined a multiobjective GBDM framework to perform bi-objective graph mining in terms of subgraph support and size maximization. Two different search methods were considered with this aim, a multiobjective beam search and a multiobjective evolutionary programming (MOEP). In this contribution, we extend the latter formulation to a three-objective framework by incorporating another classical graph mining objective, the subgraph diameter. The proposed MOEP method for multiobjective GBDM is tested on five synthetic and real-world datasets and its performance is compared against single and multiobjective subgraph mining approaches based on the classical Subdue technique in GBDM. The results highlight the application of multiobjective subgraph mining allows us to discover more diversified subgraphs in the objective space.

1. Introduction

Many applications that contain complicated structures and relational objects rely on a graph-based data representation [1,2]. Some examples include scientific information analysis [3], bioinformatics [4], transportation networks [5], web data analysis [6], among others. Subgraph mining in graph-based data is the process of discovering subgraphs subject to some objective function. It usually involves applying some user-defined threshold, such as mining subgraphs whose frequency is above a specified threshold. For this task, several algorithms have been introduced in the graph-based data mining (GBDM) literature, starting with the classical heuristic search-based Subdue method [7] and being followed by some well-known exact search methods such as Gaston, gSpan, FSG, etc., [2,8]. Recently, evolutionary programming [9] has also been applied for frequent subgraph mining [10,11]. The proposal was basically an extension of Subdue and showed an improved performance over it. The performance improvement was a consequence of the use of global search instead of the beam search [12] with no backtracking as applied by the standard Subdue method in the subgraph search space.

Recently some important limitations of the existing approaches that operate by using simple user-defined constraints on the mined subgraphs have been highlighted in [5]. In addition, several authors [4,13–16] have noted that only employing

* Corresponding author at: European Centre for Soft Computing, 33600-Mieres, Spain.
E-mail addresses: prakash.shelokar@softcomputing.es (P. Shelokar), arnaud.quirin@softcomputing.es (A. Quirin), oscar.cordon@softcomputing.es, ocordon@decsai.ugr.es (Ó. Cordón).

0022-0000/$ – see front matter
http://dx.doi.org/10.1016/j.jcss.2013.03.005

Please cite this article in press as: P. Shelokar et al., Three-objective subgraph mining using multiobjective evolutionary programming, J. Comput. System Sci. (2013), http://dx.doi.org/10.1016/j.jcss.2013.03.005

Mining Sequential Patterns Using the Integration of Fuzzy Logic and Graph Search Techniques

Pisit Phokharatkul
Department of Computer Engineering, Faculty of Engineering,
Mahidol University, Nakhon Pathom, 73170, Thailand
Email: egpph@mahidol.ac.th

Sukanya Yuenyong
Information Management Department, KASIKORNBANK PCL,
RatBurana, Bangkok 10140, Thailand
Email: sukanya.yu@kasikornbank.com, lukkaew@live.com

Abstract

Sequential pattern discovery is an important problem in data mining. In recent years, the researchers have been to find the new techniques to extract the sequential patterns from a large database. In this research, an effective way of the integrating fuzzy logic and graph search methods to create the fuzzy logic and graph search (FGS) algorithm for sequential pattern mining is proposed. The execution time of the two graph search techniques was compared. It was found that the depth-first search (DFS) takes less execution time than the breadth-first search (BFS). Also, the FGS algorithm takes less execution time than the GST algorithm when the k-sequence is greater than or equal to the 1-sequence (k≥2). The outcomes of the FGS algorithm are more valuable than the GST algorithm because the quantitative values of each transaction are considered. Finally, it was found that the FGS outcomes are substantially lower than the GST outcomes. Sometimes, the reduction is an advantage but it may not be so for all cases.

Key Words: data mining, sequential pattern, fuzzy logic, graph search.

1. Introduction

Nowadays, we use computers to collect data in various formats such as text file, database and XML formats. The advantages of data collection are more than search and review. The knowledge can be extracted from the existing data; call *data mining*. Data mining is the process of extracting interesting information or patterns from large information repositories. There are many types of data mining. Sequential pattern discovery is an important problem in data mining. In recent years there have been and continue to be many researchers trying to find new techniques to extract the sequential patterns from large database. They used difference algorithms such as: DSG algorithm [1], fuzzy algorithm [2], the algorithm mining path traversal pattern [3], and Generalized Sequential Pattern (GSP) [4].

This paper studies the integrating fuzzy logic and graph search algorithm for sequential pattern mining. There are two important things in the sequential pattern mining. First is the performance of execution time. Secondary is the result, how can extract the most useful sequential patterns. This problem can construe into many issues depending on the interest of an individual such as: case of inventory, mining sequential pattern use for predicting the consumer purchasing behavior. The outcome of sequential pattern mining can predict what the next product or group of products will be purchased when the product or group of products already purchased is known.

Although the algorithms always extract the sequential pattern, the user will always desire a better pattern. However, new algorithms must continue to solve the two important constraints of execution time and give a better result.

2. Data mining

Data mining [1] is the process of extracting interesting (non-trivial, implicit, previously unknown and potentially useful) information or pattern from large information repositories such as: relational database, data warehouse, XML repository, etc. Thus, data mining is known as one of the core processes of knowledge discovery in database. The processes of the knowledge discovery method consists the steps is shown in Figure 1.

First, the data source comes from different databases, which may have some inconsistence and duplications. The system cleans the data source by removing some noises or makes some compromises. And the integrated data sources can be stored in the

Pattern Recognition 43 (2010) 752–766

Contents lists available at ScienceDirect

Pattern Recognition

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

A graph-theoretical clustering method based on two rounds of minimum spanning trees

Caiming Zhong[a,b,c], Duoqian Miao[a,b,*], Ruizhi Wang[a,b]

[a]Department of Computer Science and Technology, Tongji University, Shanghai 201804, PR China
[b]Key Laboratory of Embedded System & Service Computing, Ministry of Education of China, Shanghai 201804, PR China
[c]College of Science and Technology, Ningbo University, Ningbo 315211, PR China

ARTICLE INFO

Article history:
Received 22 January 2009
Received in revised form 19 July 2009
Accepted 24 July 2009

Keywords:
Graph-based clustering
Well-separated cluster
Touching cluster
Two rounds of MST

ABSTRACT

Many clustering approaches have been proposed in the literature, but most of them are vulnerable to the different cluster sizes, shapes and densities. In this paper, we present a graph-theoretical clustering method which is robust to the difference. Based on the graph composed of two rounds of minimum spanning trees (MST), the proposed method (2-MSTClus) classifies cluster problems into two groups, i.e. separated cluster problems and touching cluster problems, and identifies the two groups of cluster problems automatically. It contains two clustering algorithms which deal with separated clusters and touching clusters in two phases, respectively. In the first phase, two round minimum spanning trees are employed to construct a graph and detect separated clusters which cover distance separated and density separated clusters. In the second phase, touching clusters, which are subgroups produced in the first phase, can be partitioned by comparing cuts, respectively, on the two round minimum spanning trees. The proposed method is robust to the varied cluster sizes, shapes and densities, and can discover the number of clusters. Experimental results on synthetic and real datasets demonstrate the performance of the proposed method.

1. Introduction

The main goal of clustering is to partition a dataset into clusters in terms of its intrinsic structure, without resorting to any a priori knowledge such as the number of clusters, the distribution of the data elements, etc. Clustering is a powerful tool and has been studied and applied in many research areas, which include image segmentation [1,2], machine learning, data mining [3], and bioinformatics [4,5]. Although many clustering methods have been proposed in the recent decades, there is no universal one that can deal with all cluster problems, since in the real world clusters may be of arbitrary shapes, varied densities and unbalanced sizes [6,7]. In addition, Kleinberg [8] presented an impossibility theorem to indicate that it is difficult to develop a universal clustering scheme. However, in general, users have not any a priori knowledge on their datasets, which makes it a tough task for them to select suitable clustering methods. This is the dilemma of clustering.

Two techniques have been proposed and studied to alleviate the dilemma partially, i.e. clustering ensemble [9–11] and multiobjective clustering [12]. The basic idea of a clustering ensemble is to use different data representation, apply different clustering methods with varied parameters, collect multiple clustering results, and discover a cluster with better quality [13]. Fred and Jain [13] proposed a co-association matrix to depict and combine the different clustering results by exploring the idea of evidence accumulation. Topchy et al. [10] proposed a probabilistic model of consensus with a finite mixture of multinomial distributions in a space of clusterings, and used the EM algorithm to find the combined partitions. Taking advantage of correlation clustering [14], Gionis et al. [11] presented a clustering aggregation framework, which can find a new clustering that minimizes the total number of disagreements with all the given clusterings. Being different from a clustering ensemble which is limited to the posteriori integration of the solutions returned by the individual algorithms, multiobjective clustering considers the multiple clustering objective functions simultaneously, and trades off solutions during the clustering process [12]. Compared with the individual clustering approach, both clustering ensembles and multiobjective clustering can produce more robust partitions and higher cluster qualities. In addition, some of other clustering methods can automatically cope with arbitrary shaped and non-homogeneous clusters [15].

* Corresponding author at: Department of Computer Science and Technology, Tongji University, Shanghai 201804, PR China. Tel.: +86 21 69589867.
E-mail addresses: charman_zhong@hotmail.com (C. Zhong), miaoduoqian@163.com (D. Miao).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.07.010

BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

Graph-based clustering and characterization of repetitive sequences in next-generation sequencing data

Petr Novák, Pavel Neumann and Jiří Macas*

Abstract

Background: The investigation of plant genome structure and evolution requires comprehensive characterization of repetitive sequences that make up the majority of higher plant nuclear DNA. Since genome-wide characterization of repetitive elements is complicated by their high abundance and diversity, novel approaches based on massively-parallel sequencing are being adapted to facilitate the analysis. It has recently been demonstrated that the low-pass genome sequencing provided by a single 454 sequencing reaction is sufficient to capture information about all major repeat families, thus providing the opportunity for efficient repeat investigation in a wide range of species. However, the development of appropriate data mining tools is required in order to fully utilize this sequencing data for repeat characterization.

Results: We adapted a graph-based approach for similarity-based partitioning of whole genome 454 sequence reads in order to build clusters made of the reads derived from individual repeat families. The information about cluster sizes was utilized for assessing the proportion and composition of repeats in the genomes of two model species, *Pisum sativum* and *Glycine max*, differing in genome size and 454 sequencing coverage. Moreover, statistical analysis and visual inspection of the topology of the cluster graphs using a newly developed program tool, *SeqGrapheR*, were shown to be helpful in distinguishing basic types of repeats and investigating sequence variability within repeat families.

Conclusions: Repetitive regions of plant genomes can be efficiently characterized by the presented graph-based analysis and the graph representation of repeats can be further used to assess the variability and evolutionary divergence of repeat families, discover and characterize novel elements, and aid in subsequent assembly of their consensus sequences.

Background

The ability of next-generation sequencing technologies to analyze eukaryotic genomes in a fast and cost-efficient manner [1-3] is providing new opportunities for investigating biological problems that, due to their complexity, could not be addressed before. One such question concerns the role that repetitive DNA plays in shaping the structure and evolution of plant genomes. Its elucidation depends in large part on performing a comparative analysis of repeat composition in a large number of plant species differing in size and other characteristics of their genomes. However, repetitive sequences, composed of numerous and diverse families of mobile elements and tandem repeats, account for up to 97% of plant nuclear DNA [4,5]. Thus, genome-wide characterization of repetitive elements can only be achieved when large volumes of sequencing data are available, which has long been limited to a few model species due to the speed and cost constraints imposed by classical sequencing. Compared to the conventional, clone-based Sanger sequencing approaches, the next-generation technologies work at unprecedented speed, sequencing up to several gigabases in a single reaction for a fraction of the cost [1-3]. Although this amount of sequencing data is still not sufficient to provide the coverage typically needed for whole genome assembly, it enables representative sampling of

* Correspondence: macas@umbr.cas.cz
[1] Biology Centre ASCR, Institute of Plant Molecular Biology, Branisovska 31, Ceske Budejovice, CZ-37005, Czech Republic
Full list of author information is available at the end of the article

BioData Mining

REVIEW Open Access

Using graph theory to analyze biological networks

Georgios A Pavlopoulos[1,2*], Maria Secrier[3], Charalampos N Moschopoulos[4,5], Theodoros G Soldatos[6], Sophia Kossida[5], Jan Aerts[2], Reinhard Schneider[3,7] and Pantelis G Bagos[1]

* Correspondence: pavlopou@embl.de
[1]Department of Computer Science and Biomedical Informatics, University of Central Greece, Lamia, 35100, Greece
Full list of author information is available at the end of the article

Abstract

Understanding complex systems often requires a bottom-up analysis towards a systems biology approach. The need to investigate a system, not only as individual components but as a whole, emerges. This can be done by examining the elementary constituents individually and then how these are connected. The myriad components of a system and their interactions are best characterized as networks and they are mainly represented as graphs where thousands of nodes are connected with thousands of vertices. In this article we demonstrate approaches, models and methods from the graph theory universe and we discuss ways in which they can be used to reveal hidden properties and features of a network. This network profiling combined with knowledge extraction will help us to better understand the biological significance of the system.

Keywords: biological network clustering analysis, graph theory, node ranking

Introduction

The theory of complex networks plays an important role in a wide variety of disciplines, ranging from computer science, sociology, engineering and physics, to molecular and population biology. Within the fields of biology and medicine, potential applications of network analysis include for example drug target identification, determining a protein's or gene's function, designing effective strategies for treating various diseases or providing early diagnosis of disorders. Protein-protein interaction (PPI) networks, biochemical networks, transcriptional regulation networks, signal transduction or metabolic networks are the highlighted network categories in systems biology often sharing characteristics and properties.

Protein-protein interaction (PPI) networks [1] mainly hold information of how different proteins operate in coordination with others to enable the biological processes within the cell. Despite the fact that for the majority of proteins the complete sequence is already known, their molecular function is not yet fully determined. Predicting protein function is still a bottleneck in computational biology research and many experimental and computational techniques have been developed in order to infer protein function from interactions with other biomolecules. Large-scale and high-throughput techniques can detect proteins that interact within an organism. Among them, the most well-known are the pull down assays [2], tandem affinity purification (TAP) [3], yeast two-hybrid (Y2H) [4], mass spectrometry [5], microarrays [6] and phage display [7]. Some very well-known datasets that have been recently produced by employing

Algorithmica (2012) 62:38–53
DOI 10.1007/s00453-010-9442-9

Complexity of Finding Graph Roots with Girth Conditions

Babak Farzad · Lap Chi Lau · Van Bang Le · Nguyen Ngoc Tuy

Received: 10 August 2009 / Accepted: 1 August 2010 / Published online: 14 September 2010

Abstract Graph G is the square of graph H if two vertices x, y have an edge in G if and only if x, y are of distance at most two in H. Given H it is easy to compute its square H^2, however Motwani and Sudan proved that it is NP-complete to determine if a given graph G is the square of some graph H (of girth 3). In this paper we consider the characterization and recognition problems of graphs that are squares of graphs of small girth, i.e. to determine if $G = H^2$ for some graph H of small girth. The main results are the following.

- There is a graph theoretical characterization for graphs that are squares of some graph of girth at least 7. A corollary is that if a graph G has a square root H of girth at least 7 then H is unique up to isomorphism.
- There is a polynomial time algorithm to recognize if $G = H^2$ for some graph H of girth at least 6.
- It is NP-complete to recognize if $G = H^2$ for some graph H of girth 4.

Nguyen Ngoc Tuy is supported by the Ministry of Education and Training, Vietnam, Grant No. 3766/QD-BGD & DT.

B. Farzad (✉)
Department of Mathematics, Brock University, Saint Catharines, Canada
e-mail: bfarzad@brocku.ca

L.C. Lau
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, Hong Kong
e-mail: chi@cse.cuhk.edu.hk

V.B. Le
Institut für Informatik, Universität Rostock, Rostock, Germany
e-mail: le@informatik.uni-rostock.de

N.N. Tuy
Department of Computer Science, Hong Duc University, Thanh Hoa City, Vietnam
e-mail: nntuy@yahoo.com