Parallel GPU-based Plane-Sweep Algorithm for Construction of iCPI-Trees

Witold Andrzejewski, Faculty of Computing, Poznan University of Technology, Poznan, Poland

Pawel Boinski, Faculty of Computing, Poznan University of Technology, Poznan, Poland

ABSTRACT

This article tackles the problem of efficient construction of iCPI trees, frequently used in co-location pattern discovery in spatial databases. It discusses the methods for parallelization of iCPI-tree construction and plane-sweep algorithms used in state-of-the-art algorithms for co-location pattern mining. The main contribution of this paper is threefold: (1) a general algorithm for parallel iCPI-tree construction is presented, (2) two variants of parallel plane-sweep algorithm (which can be used in conjunction with the aforementioned iCPI-tree construction algorithm) are introduced and (3) all three algorithms are implemented on CUDA GPU platform and their performance is tested against an efficient multithreaded parallel implementation of iCPI-tree construction on CPU. Experiments prove that our solutions allow for large speedups over CPU version of the algorithm. This paper is an extension of the conference paper (Andrzejewski & Boinski, 2014).

Keywords: Co-location Pattern Mining, Data Mining, Data Structures, GPGPU, GPU, Parallel Computing

INTRODUCTION

Widespread usage of remote sensing and GPS technology, combined with easily available mass storage devices, allows registering and storing enormous volumes of spatial data. For example, weather and climate datasets obtained from satellite observations can grow by terabytes each day. It is obvious that human abilities are not sufficient to interpret such data. Therefore, automatic methods, known as Knowledge Discovery in Databases (KDD) are used to obtain valid, novel and potentially useful patterns (Fayad, Piatetsky-Shapiro & Smyth, 1996) from such datasets.

Discovered patterns differ based on types of mined data. In spatial data mining, one of the most popular types of patterns is called a spatial co-location pattern or in short a co-location. Co-locations were defined by Shekhar and Huang (2001) as sets of spatial features which instances are frequently located close to each other. A spatial feature is a class of spatial object,
i.e., a characteristic of space in a given location. Exemplary spatial features include: species, business types or points of interest like shops, cinemas, gas stations etc. As there is a great variety of spatial feature types, the co-location pattern discovery can be applied in a wide range of domains, such as marketing, ecology, epidemiology and public health, meteorology, mobile advertising, astronomy etc.

Multiple algorithms for the efficient co-location pattern mining have been developed recently (Andrzejewski & Boinski, 2013; Boinski & Zakrzewicz, 2014). In (Andrzejewski & Boinski, 2013), a GPU-CM algorithm has been proposed to utilize parallel processing capabilities of graphics processing units (GPUs). The algorithm is an adaptation of the state-of-the-art solution in the field of co-location pattern mining proposed by Wang, Bao and Lu (2009). Wang’s solution uses a dedicated auxiliary structure (called iCPI-tree) to store neighborhood information. The authors of GPU-CM assume that this structure is prebuilt and supplied in advance before the execution of the co-location pattern mining algorithm on the GPU. In this paper we remove this limitation. We propose two variants of the GPU-based algorithm for building iCPI-tree structure directly on the GPU.

The structure of this paper is as follows. In Section “Co-location Pattern Mining” we provide the definition of the co-location pattern mining problem and related terms. In Section “Plane Sweep” we describe the most popular strategy for the iCPI-tree construction. In Section “General Processing on GPUs” we shortly introduce characteristics of GPU programming and provide related terms. Main contribution of this paper is presented in Section “Parallel Algorithm for iCPI-hashmap Construction” and Section “Parallel Plane Sweep Algorithms”. Section “Experiments” contains the results of experimental evaluation of the performance of our solutions. We finalize the paper with a short summary in Section “Summary and Future Work”.

CO-LOCATION PATTERN MINING

A spatial dataset $S$ consists of $n$ spatial objects. Each such object has assigned: its location, unique identifier and a category called a spatial feature $f$. A spatial object $o$ is an instance of the feature $f$ if $o$ has a type $f$. A set of $m$ spatial features of objects in $S$ is denoted by $F$. To prevent duplicates in results of the co-location pattern mining we assume, without loss of generality, that there is a total order (e.g., lexicographical) among spatial features.

In the co-location pattern mining two objects satisfy a neighborhood relationship if they are located close to each other in space. Let $R$ be a neighbor relation and $r$ be a neighborhood distance threshold. We define $\text{dist}: S \times S \to R^+ \cup \{0\}$ as a metric function that computes the distance between two objects. Relation $R$ is satisfied for objects such that the distance between them is less or equal than the threshold $r$, i.e.,

$$R = \{(o_i, o_j) \in S \times S : \text{dist}(o_i, o_j) \leq r\}.$$

A co-location pattern (a co-location in short) $C = \{c_1, c_2, \ldots, c_k\}$ is a subset of spatial features $C \subseteq F$ whose instances $I \subseteq S$ form a clique w.r.t. the relation $R$. An instance $I$ of the co-location pattern $C$ is a set of objects $\{o_1, o_2, \ldots, o_k\}$ such that $o_i$ has a type $c_i$ and $(o_i, o_j) \in R$ for all $1 \leq i, j \leq k$. To retrieve only potentially interesting co-location patterns,
Using a Model Quality Framework for Requirements Specification of an Enterprise Modeling Language
www.igi-global.com/chapter/using-model-quality-framework-requirements/4372?camid=4v1a

The Information Quality of Databases
www.igi-global.com/chapter/information-quality-databases/11160?camid=4v1a