Mining Frequent Boolean Expressions: Application to Gene Expression and Regulatory Modeling

Mohammed J. Zaki, Rensselaer Polytechnic Institute, USA
Naren Ramakrishnan, Virginia Tech., USA
Lizhuang Zhao, Microsoft Corp., USA

ABSTRACT

Regulatory network analysis and other bioinformatics tasks require the ability to induce and represent arbitrary boolean expressions from data sources. In this paper, the authors introduce a novel framework called BLOSOM for mining (frequent) boolean expressions over binary-valued datasets. Boolean expressions can be grouped into four categories: pure conjunctions, pure disjunctions, conjunction of disjunctions, and disjunction of conjunctions. The authors’ main focus is on mining the simplest expressions (the minimal generators), but also to propose closure operators that yield closed (or unique maximal) boolean expressions. BLOSOM efficiently mines frequent boolean expressions by utilizing a number of methodical pruning techniques. Experiments showcase the behavior of BLOSOM for different input settings and parameter thresholds. Application studies on gene expression and gene regulation patterns showcase the effectiveness of this approach.

Keywords: Bioinformatics, Boolean Expressions, Closed Patterns, Itemset Mining, Minimal Generators

INTRODUCTION

A key class of problems in biological knowledge discovery pertains to modeling complex relationships between entities, such as GO category descriptions, genetic regulatory connections, and metabolic pathway networks. To effectively address these tasks, we require data mining techniques that possess the expressiveness to model arbitrary boolean connections.

However, many state-of-the-art techniques are rather restrictive in the class of expressions they can mine.

For example, itemset mining (Agrawal et al., 1996) takes as input a binary-valued dataset and discovers patterns that are pure conjunctions of items. Admittedly, such techniques are useful in bioinformatics, but have limited scope. For example, given the expression levels (high or low) of genes for various cancers, we may find that cancerous tissues have high levels of genes \((g_1 \text{ AND } g_5 \text{ AND } g_{13}) \text{ OR } (g_{10} \text{ AND} \ldots\)
This might indicate that these groups of genes are co-regulated and somehow linked to cancer. These techniques can be generalized to include negations of descriptions and used to mine redescriptions (Ramakrishnan et al., 2004), that is, finding alternative ways of describing a group of genes. For instance, from yeast gene expression data, we may find that the gene highly expressed at time point 15 mins \( (d_{183}) \), but not involved in mannose transportation \( (d_{388}) \) or fructose metabolism \( (d_{515}) \) or not part of external protective structure \( (d_{460}) \), are the exact same genes also highly expressed at time point 20 mins \( (d_{184}) \), provided they do not have unknown molecular function \( (d_{309}) \). This redescription can be written as an equivalence of boolean expressions: \( d_{183} \text{ AND } \neg d_{388} \text{ AND } \neg d_{460} \text{ AND } (d_{515}) \leftrightarrow d_{184} \text{ AND } (d_{309}) \).

We seek to generalize these approaches to mine arbitrary boolean expressions. Boolean expression mining can provide tremendous value, but there are two main challenges to contend with. The first deals with the problem of high computational complexity. With \( n \) items (or variables), there are \( 2^n \) possible distinct-valued Boolean expressions. With \( n \) items, there are \( 2^n \) distinct ways of assigning true/false values to the \( n \) terms, and for any such assignment the truth value of the expression can be set to true or false, giving the \( 2^2 \) value. These are far too many to enumerate. To make the search practical we focus on only the frequent boolean expressions. Also instead of mining all frequent boolean expressions, we focus on mining a lossless subset that retains complete frequency information, namely closed boolean expression. The second challenge relates to comprehension of the patterns, i.e., they may be complex and difficult to understand. Here we focus on mining the simplest or minimal expressions (which are in fact the minimal generators of the closed expressions) that still from a lossless representation of all possible boolean expressions.

In this paper, we present a novel framework, called BLOSOM (an anagram of the bold letters in BOOLean expression MIning over attribute Sets), the first such approach to simultaneously mine closed boolean expressions over attribute sets and their minimal generators. Our main contributions are as follows: We organize boolean expressions into four categories: (i) pure conjunctions (AND-clauses), (ii) pure disjunctions (OR-clauses), (iii) conjunctive normal form (CNF; conjunction of disjunctions), and (iv) disjunctive normal form (DNF; disjunction of conjunctions). For both CNF and DNF expressions we propose a closure operator, and we give a characterization of the minimal generators. BLOSOM employs a number of effective pruning techniques for searching over the space of boolean expressions, yielding orders of magnitude in speedup. We conducted several experiments on synthetic datasets to study the behavior of BLOSOM with respect to (w.r.t.) different input settings and parameter thresholds. We also highlight some of the patterns found using BLOSOM on real datasets from bioinformatics applications such as analysis of gene expression patterns, and boolean gene regulatory network discovery.

**PRELIMINARY CONCEPTS**

**Lattice Theory:** Let us first review a few facts from lattice theory (Davey & Priestley, 1990), which will be useful in our discussion. Let \( (P, \subseteq) \) be a partially ordered set (also called a poset). Let \( X, Y \in P \) and let \( f : P \rightarrow P \) be a function on \( P \). \( f \) is called monotone if \( X \subseteq Y \Rightarrow f(X) \subseteq f(Y) \). We say that \( f \) is idempotent if \( f(f(X)) = f(X) \). \( f \) is called extensive (or expansive) if \( X \subseteq f(X) \). Finally, \( f \) is called intensive (or contractive) if \( f(X) \subseteq X \). A closure operator on \( P \) is a function \( \subseteq : P \rightarrow P \) such that \( C \) is monotone, idempotent, and extensive. \( X \) is called closed if
Emerging Web Tools and Their Applications in Bioinformatics
www.igi-global.com/chapter/emerging-web-tools-their-applications/76147?camid=4v1a

State-of-the-Art GPGPU Applications in Bioinformatics
www.igi-global.com/article/state-of-the-art-gpgpu-applications-in-bioinformatics/105596?camid=4v1a