Peer-to-Peer Usage Analysis

Florent Masseglia
INRIA Sophia Antipolis, France

Pascal Poncelet
EMA-LGI2P Site EERIE, France

Maguelonne Teisseire
LIRMM UMR CNRS 5506, France

INTRODUCTION

With the huge number of information sources available on the Internet and the high dynamics of their data, *peer-to-peer* (P2P) systems propose a communication model in which each party has the same capabilities and can initiate a communication session. These networks allow a group of computer users with the same networking program to connect with each other and directly access resources from one another. P2P architectures also provide a good infrastructure for data and computer intensive operations such as data mining.

In this article we consider a new data mining approach for improving resource searching in a dynamic and distributed database such as an unstructured P2P system, that is, in Masseglia, Poncelet, and Teisseire (2006) we call this problem *P2P usage analysis*. More precisely we aim at discovering frequent behaviors among users of such a system. We will focus on the sequential order between actions performed on each node (requests or downloads) and show how this order has to be taken into account for extracting useful knowledge. For instance, it may be discovered, in a P2P file sharing network that for 77% of nodes from which a request is sent for "*Mandriva Linux*," the file "*Mandriva Linux 2005 CD1 i585-Limited-Edition-Mini.iso*" is chosen and downloaded; then a new request is performed with the possible name of the remaining iso images (i.e., "*Mandriva Linux 2005 Limited Edition*"), and in the large number of returned results the image corresponding to "*Mandriva Linux 2005 CD2 i585-Limited-Edition-Mini.iso*" is chosen and downloaded.

Such knowledge is very useful for proposing the user with often downloaded or requested files according to a majority of behaviors. It could also be useful in order to avoid extra bandwidth consumption, which is the main cost of P2P queries (Ng, Chu, Rao, Sripanidkulchai, & Zhang, 2003).

BACKGROUND

Mining either association rules or sequential patterns in very large distributed databases as unstructured P2P systems is far away from trivial. For instance, in Wolff and Schuster (2003), authors propose to mine association rules, that is, sets of objects that tend to associate with one another, in a P2P system. The proposed algorithm combines, at each node, association rule mining algorithm with a majority voting protocol to discover all of the association rules that exist in the distributed database.

By nature P2P systems are dynamic, that is, nodes act independently of one another, and intermediate results may be overturned as new data arrives. Furthermore, whenever a node departs, the sequence of that node also disappears, and the global database has to be reconsidered. Traditional approaches for mining sequential patterns (Agrawal & Srikant, 1995; Pei et al., 2001) are irrelevant in such a dynamic context because they consider that the whole database is available. In Masseglia, Teisseire, and Poncelet (2003), we proposed to discover relationships and global patterns that exist between connecting users (Web usage mining). We proposed a new "client/server/engine" architecture for taking advantage of the computing power available on the machine a user navigates with. In this article our goal is different since it takes into account the dynamic nature of the considered system. We consider that the connected nodes can act with a special peer (called *meter peer*) in order to provide the end user with a

good approximation of patterns embedded in this very large distributed database.

MAIN FOCUS OF THE ARTICLE

In this section we first define the problem statement, and then we propose a new solution to efficiently mine frequent sequences embedded in a P2P system.

Problem Statement

Let $I = \{x_1, ..., x_n\}$ be a set of distinct literals called *items*. In the following we assume that for each item we are provided with the action performed, that is, request or download. An item x_i is thus denoted either [d, x_i] for downloading or [r, x_i] for requesting. A *sequence* is an ordered list of itemsets denoted by $<s_1\ s_2 \ldots s_n>$, where s_j is an itemset, that is, a set of items that occur together. Let us consider the following sequence: *< ([r, Mandriva Linux]) ([d, Mandriva-Limited-Edition-CD1.iso], [r, Mandriva Limited Edition]) ([d, Mandriva-Limited-Edition-CD2.iso])>*. The user connected through that peer first sent a request ("r") for searching any resource containing the words "*Mandriva Linux*," then downloaded and requested, at the same time, respectively the file "*Mandriva-Limited-Edition-CD1.iso*," and any resource containing the string "*Mandriva Limited Edition*," and finally downloaded the file "*Mandriva-Limited-Edition-CD2.iso*."

A sequence $s_1 = < a_1\ a_2 \ldots a_n >$ is a *subsequence* of another sequence $s_2 < b_1\ b_2 \ldots b_m >$, denoted $s_1 < s_2$, if there exist integers $i_1 < i_2 < \ldots < i_n$ such that $a_1 \subseteq b_{i1}$, $a_2 \subseteq b_{i2}$, … , $a_n \subseteq b_{in}$.

Let D be a database of customer data-sequences. The support for a sequence S, also called *supp(S)*, is defined as the number of total data-sequences that contain S. If *supp(S)* ≥ *minSupp*, with a minimum support value *minSupp* given by the user, S is considered as a *frequent* sequential pattern.

Here we adapt the problem statement proposed by Wolff and Schuster (2003) to our concern. When the database is dynamically updated (adding or deleting transactions) we denote D_t the database at time t. Let us assume that the database is also partitioned among an unknown number of nodes. We denote such a partition of node u, at time t, D_{ut}. In fact D_{ut} corresponds to the sequence of actions performed on the node. Let us assume that $D_t = D_{ut} \cup ... \cup D_{vt}$, where u_t ... v_t are the available nodes at time t. The problem of sequential pattern mining in such a large-scale distributed systems D_t is thus to find the set of frequent sequential patterns in D_t according to the minSupp value. Let us consider that F_{Dt} is the result to obtain (the result that would be exhibited by an algorithm that would explore the whole set of solutions), F_{Dt} is thus the set of frequent sequential patterns to find in D_t. Let us now consider $\sim F_{Dt}$, the set of approximate sequential patterns. We require that, as nodes u_t ... v_t are dynamics, $\sim F_{Dt}$ will converge as fast as possible to F_{Dt}.

In order to illustrate the problems related to mining sequential patterns in such a dynamic context, let us consider three sequences standing for downloading operations performed on nodes u_t, v_t, and w_t:

D_{ut}	<([d,1]) ([d,2]) ([d,3]) ([d,4]) ([d,5])>
D_{vt}	<([d,1]) ([d,2]) ([d,1]) ([d,3]) ([d,5])>
D_{wt}	<([d,1]) ([d,2]) ([d,4]) ([d,5]) ([d,6])>

From such sequences, the set of items with their associated support is the following: ([d, 1]) [100%], ([d, 2]) [100%], ([d, 3]) [66%], ([d, 4]) [66%], ([d, 5]) [100%] and ([d, 6]) [33%]. Let us assume that a support value, minSupp, is set to 100%, then the set of frequent sequences at time *t[i]* on $F_{Dt[i]}$ is: $F_{Dt[i]}$ = *<([d,1]) ([d,2]) ([d,5])>*. Let us now assume, at time *t[i+1]* that the node w_t departs, then the set of frequent sequences becomes $F_{Dt[i+1]}$ = *<([d,1]) ([d,2]) ([d,3]) ([d,5]) >* since the support of the item ([d, 3]) is now 100%.

A New Heuristic for Mining Sequential Patterns in P2P Systems

As a matter of fact, the nodes in a P2P context may connect or depart frequently while D_t is still being analyzed. Our proposal is to consider D_t as a unit able to receive candidate sequences, to evaluate the support of each candidate on sequence in D_t, and to send back the result. This kind of "scan," distributed on all the connected nodes, relies on a stochastic algorithm for combinatorial optimization problems.

First D_t is empty until a node u_t sends its sequence. The unstructured P2P architecture we propose allows a special peer (hereafter the "$Distributed_{SP}$ peer") to get