Learning Concepts from Intelligence Data Embedded in a Supervised Graph

Joseph T. Potts, Diane J. Cook, Lawrence B. Holder, and Jeffrey Coble
University of Texas at Arlington
Arlington, TX 76019 USA
{potts, cook, holder,coble}@cse.uta.edu

Keywords: knowledge discovery and dissemination, graph-based data mining, terrorism

Abstract

We describe a method of learning concepts from examples that are represented as a graph. We use this method, implemented using the Subdue system, to find concepts when the training examples are embedded into a single connected graph, or supervised graph. We demonstrate the technique using homeland security data.

1. Introduction

The ability to mine relational data has become a crucial challenge in security-related domains. For example, the U.S. House and Senate Intelligence Committees' report on the activities of the intelligence community before and after the September 11, 2001 terrorist attacks revealed the necessity for "connecting the dots" (2002); that is, focusing on the relationships between entities in the data, rather than merely on an entity's attributes. A natural representation for this information is a graph, and the ability to learn concepts from such graphs could lead to significant improvement in our ability to identify threats.

Graph-based learning systems typically require each training example to be represented as a disjoint graph. In a highly relational domain, however, the positive and negative examples of a concept are not easily separated. We call such a graph a *supervised graph*, because the graph contains embedded class information which may not easily be separated into individual labeled components. Here we describe a method of learning concepts from supervised graphs.

Our goal is to develop a concept learner that allows positive and negative examples of a concept to be interconnected in one input graph. This type of learning algorithm is useful for domains in which relationships can exist between individuals from different classes. For example, consider a social network in which we want to find relational patterns distinguishing various income levels. Individuals of a particular income level can appear anywhere in the graph and may be related to individuals at other income levels, so we cannot easily partition the graph into separate cases without potentially severing the target relationships.

Traditional learning and data mining algorithms will encounter difficulties in preparing data for input when it is embedded in a single connected graph. If individual graphs are required for each example, one can excise each example along with some amount of surrounding data to create a disconnected graph containing the example. If the examples are close enough to each other in the original graph, then this surrounding data may overlap with the surrounding data of another example or even the example itself. This will result in some data appearing in more than one example graph. There is also some risk of taking the wrong amount of surrounding data, either too large a region around the example causing extra data to be handled, or too small a region resulting in the loss of potentially vital information. In addition, it may be impossible to determine the "shape" of the area that should be excised. Since processing graph-based data is very resource intensive, any redundant information can have a drastic effect on performance.

2. Subdue-EC

Subdue was originally written to discover interesting patterns in structural data (Cook and Holder 2000). Subdue finds patterns, or substructures, in data that is represented as a labeled graph. In accordance with the Minimum Description Length principle, the substructure that Subdue deems to be the most interesting is the one that yields the smallest descriptive length when it is used to compress the graph.

Subdue's search through the space of candidate substructures terminates upon reaching a user-specified limit on the number of substructures extended, or upon exhaustion of the search space. Once Subdue returns the list of best substructures found, the graph can be compressed using the best substructure. The compression procedure replaces all instances of the substructure in the input graph by single vertices, which represent the substructure definition. Iterating over this process until the graph can no longer be compressed will produce a hierarchical, conceptual clustering of the input data.

To allow Subdue to perform concept learning, all graphs that are positive examples are kept distinct from the graphs that are negative examples. The goal in this case is find substructures which occur in as many positive examples as possibly but rarely or never in the negative graphs. However, this method of learning concepts is still not able to accommodate a single graph containing all of the examples.

The new Subdue-EC algorithm processes supervised graphs, in which training examples can consist of single vertices or entire subgraphs. To embed examples and determine their class, Subdue-EC requires an additional vertex for each user-defined example labeled with the class name and connected to each vertex of the example by an edge. If an example is represented by an entire subgraph, a new representative class vertex is created and connected by an edge to edge vertex of the example subgraph. Note that Subdue's initial state is much smaller because it is focused on instances of the single vertex subgraph with the class label.

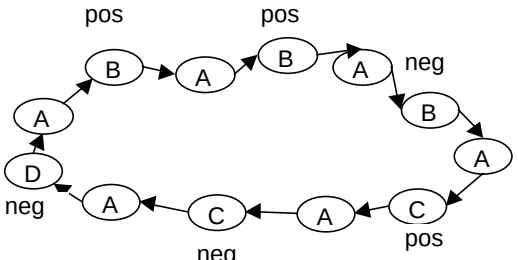

Figure 1. Example supervised graph.

When the graph in Figure 1 is processed by Subdue-EC the following substructures are discovered: D (neg), B→A→C (pos), C (neg), B→A→B (pos), and B (neg). The underlined vertex here is the one being classified.

3. Detecting Security Threats

As part of a government-sponsored program, a domain has been built to simulate the evidence available about terrorist groups and their plans prior to execution. The domain consists of threat and non-threat actors and groups, targets, exploitation modes, capabilities, resources, communications, visits to targets, and transfer of resources.

The simulator generates a plan of starting a group, recruiting members with needed capabilities, acquiring needed resources, visiting a target, and then exploiting the target. All data is generalized so that no specific names are used.

For our experiments, graph vertices represent member agents from Threat and Non-threat groups. Anyone with whom these agents communicate is also added to the graph (if necessary) and connected to the agent with an undirected "association" edge. Each person may also be described using attribute and capability vertices.

Our experiments were conducted on a large graph (graph1) consisting of 435,429 vertices and 763,504 edges representing 61,105 people as well as a smaller graph (graph2) consisting of 217,901 vertices and 314,793 edges representing 30,715 people. The target THREAT class is members of known threat groups and NONTHREAT for all others.

Our goal of the experiments was to see how well Subdue-EC could classify threat and non-threat individuals, given training examples embedded in a single connected graph. We randomly sampled non-threat individuals to create a training set size equal to that of threat individuals. For individuals that included one or more of the classifying substructures, Subdue's classification accuracy was 71.98%. At the point we terminated the algorithm, however, 2,304 individuals remained unclassified. The greatest number of misclassifications were false positives / false threats, which is a preferred type of mistake for this problem.

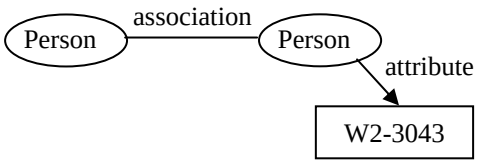

Figure 2. A discovered substructure. The individual on the left is a known threat.

As seen in Figure 2, some of the substructures Subdue-EC discovers highlight an association between two individuals in addition to attributes and capabilities of the individuals. If the individuals to be classified needed to be separated into disjoint examples, the relationship may not have been found. If we tried to extract individuals from the graph with a large enough neighborhood of information to find these discoveries, we would have to decide how much information to retain. The user cannot always know a priori how much of a neighborhood must be extracted in order to retain all potentially useful information.

In a separate experiment, we evaluated the generalizability of Subdue's results by using the substructures discovered in the first two experiments to classify individuals from a separate dataset, graph2. Of the new graphs, 69% contained the classifying substructures, and of these, 67% were classified correctly. As can be seen, while the percentage of accurate classifications does drop for the new dataset, Subdue still is able to perform fairly well on previously-unseen data.

As we have demonstrated, the Subdue-EC algorithm is effective in learning patterns to distinguish threats from non-threats, especially when focusing on group members and communication between group members. These results show the potential of this algorithm for helping intelligence analysts better identify and assess possible security threats.

Acknowledgements

This research was supported by Air Force contract F30602-01-2-0570.

References

Cook, D., and Holder, L. 2000. Graph-Based Data Mining. *IEEE Intelligent Systems* 15(2): 32-41.

U.S. Senate and House Committees on Intelligence. 2002. Joint Inquiry into Intelligence Community Activities Before and After the Terrorist Attacks of September 11, 2001.