Documentation: MIP Function — Subgroup Discovery from Heterogeneous Data

The subgroup discovery algorithm for analysis of heterogeneous data.

Metadata

Category	MIP function, data analysis algorithm
Maintainers	Jan Kralj
Homepage	http://source.ijs.si/hbp/tehin/wikis/home
Documentation	http://source.ijs.si/hbp/mipfunctions/raw/master/doc/r-tehin.pdf
Support	http://source.ijs.si/hbp/tehin/issues
Source Code	http://source.ijs.si/hbp/tehin.git http://source.ijs.si/hbp/mipfunctions/tree/master/r-tehin
License	GNU GPL
Current Version	1.0
All Versions	1.0

Description

Network analysis is an ever growing field of research capable of reasoning with data in a network setting. An important part of network analysis is so called network propositionalization, which allows us to construct feature vectors for each node in a network. An example of network propositionalization is a specific application of the personalized PageRank algorithm.

In a heterogeneous network, network propositionalization becomes less obvious. In a network with several different types of nodes, it does not make sense to construct feature vectors for all nodes because the nodes may be entirely incomparable. In our approach, we therefore take a heterogeneous network and deconstruct it into several homogeneous networks, each containing nodes of the same (so called target) type. Network propositionalization can then be applied to the homogeneous networks and the resulting vectors can be concatenated to construct a single feature vector for each node of the target type.

This function performs network propositionalization on a heterogeneous network by first deconstructing the network into several homogeneous networks. The homogeneous networks are constrcutded using user-supplied meta-paths in the heterogeneous network (for example, in a network consisting of papers and their authors, we can construct a homogeneous network of papers where two papers are connected if they share an author). The result of this function is a set of feature vectors, one for each node of the target type. Recently, the function was updated so that it can accept not only heterogeneous networks, but also standard data instances (with feature vectors) as input. In that case, the function constructs a proximity network of instances and performs network propositionalization on the resulting network.

The development of the algorithm and its implementation was partly paid by HBP.

References

Grčar, M., Trdin, N., and Lavrač, N. (2013). A methodology for mining document-enriched heterogeneous information networks. Comput. J., 56(3):321-335.

Usage

Input: A heterogeneous information network in which one node type is set as the target type, or a set of data instances with feature vectors.

Parameters: Network meta-paths on which the network is split.

Output: A feature vector for each node of the target type in the original network.

Example

We can illustrate the use of the algorithm on the ADNI data.

Task:

Given a data set consisting of examples (i.e., patients) described in terms of several descriptive variables and labelled with several target variables we want to learn a rule ensemble that will predict the values of target variables from the values of descriptive variables. In addition, we want to identify groups of patients that are similar in terms of all the target variables and describe them in terms of descriptive variables.

Input data:

Given a subset of ADNI data with 916 patients described with 34 different variables, collected at baseline evaluation, we construct a network of patients linked by their proximity. We then propositionalize the resulting network with the personalized PageRank algorithm, resulting in 916 features for each patient in the database.

Example output:

$\boldsymbol{y} = f(\boldsymbol{x}) =$ { “patient1”: [0.0, 0.00092593120920346408, 0.011948343914522241, 0.00019166712057959318,
0.0018394643179125622, 0.00013963435798087798, 0.00036063865674823629,…],

“patient2”: [0.00019243251637354329, 0.0, 0.16404962996466196, 0.00039674091625864049,
0.027579302894970886, 0.00023867152181480944, 0.0013423401342847689, …]

…}