
APCluster plugin User's Manual

Release 2.0

Michał Woźniak, Janusz Dutkowski, Jerzy Tiuryn

05 March 2010

Contents

1	About	1
1.1	About the plugin	1
1.2	About the authors	2
2	Manual	2
2.1	Requirements	2
2.2	Installation	2
2.3	Examples	2
2.4	Parameters	2
2.5	Additional options	3
	Assigning nodes to the nearest center	3
	Directed/undirected	3
2.6	Implementations	3

1 About

1.1 About the plugin

The *APCluster* plugin provides a Cytoscape-compliant implementation of the Affinity Propagation algorithm (Frey and Dueck, Science 2007). The algorithm identifies meaningful clusters by passing messages which encode the affinity of one data point to become an exemplar for another data point. The main benefit our implementation is that it enables the user to run AP on any network straight from the Cytoscape GUI and immediately visualize the results. *APCluster* also makes it easy to control the parameters of the algorithm by keeping track of parameter settings and results from multiple executions. The advantages of AP include its speed and ability to cluster a wide range of networks (both weighted and unweighted, as well as directed and undirected).

For details on the algorithm, original implementations and references please visit the official Affinity Propagation website:

<http://www.psi.toronto.edu/affinitypropagation/>

This plugin is an open source application (GPL 3 license). The plugin jar file, instructions and sources are available at:

<http://bioputer.mimuw.edu.pl/modevo>

Sources are available at the code.google website:

<http://code.google.com/p/misiej/>

Please don't hesitate to contact us with any comments and suggestion or if you are interested in co-developing this software.

1.2 About the authors

This plugin was implemented by Michal Wozniak. Project idea and guidance came from Janusz Dutkowski and Jerzy Tiurny. Affiliation: Institute of Informatics, University of Warsaw

E-mails: m.wozniak, januszd, tiurny@mimuw.edu.pl

2 Manual

2.1 Requirements

- Cytoscape 2.6.*
- Java 1.6

Plugin was tested on Windows, Linux and Mac platforms.

2.2 Installation

Download plugin jar file *APCluster.jar* from our project's website:

<http://bioputer.mimuw.edu.pl/modevo>

Just copy downloaded file to *plugins* folder in Cytoscape home directory.

2.3 Examples

Examples of usage are available on the plugin homepage in manual section:

<http://bioputer.mimuw.edu.pl/modevo/manual.html>

2.4 Parameters

Affinity Propagation parameters:

- Edge weight attribute (click refresh if no attribute names are listed) - this attribute provides the similarities between data-points: Affinity Propagation takes as input a collection of real-valued similarities between data points, where the similarity $s(i, k)$ indicates how well the data point with index k is suited to be the exemplar for data point i .
- Number of iterations - maximum number of iterations to be performed
- Stop criterion (n) - algorithm will stop after the n -th iteration without any change in centers (exemplars). In this situation, we say that the algorithm converged. If the maximum number of interactions is reached before the stop criterion is satisfied, then the algorithm did not converge. The current clusters can still be retrieved in this situation.
- Preference - the preference for a node to be an exemplar (cluster center); currently equal for every node.
- Lambda (damping factor) - when updating the messages, each message is set to λ times its value from the previous iteration plus $1 - \lambda$ times its prescribed updated value, where the damping factor λ is between 0 and 1.
- Noise option - for some networks adding noise is crucial to arrive at meaningful clusters. One example is in situations where two or more nodes have equal affinity to become exemplars. In such cases adding noise breaks the symmetry and allows one of the nodes to be selected. The original implementation adds noise to the similarity values associated with edges. We found that more stable results are achieved when noise is added to node preferences. Currently noise is equal to preference* epsilon * (random real number from a uniform distribution on $[0, 1]$) where epsilon = 0.0000001.

Other implementation-specific parameters:

- Cluster ID - name of the node attribute which will store cluster assignments
- Center ID - name of the node attribute which will store center assignments
- Take log - if selected, edges weights will be set to $\log(\text{edge_weight})$

2.5 Additional options

Assigning nodes to the nearest center

After the cluster centers (exemplars) are determined in the first phase of the algorithm, each node is assigned to the nearest center. Our implementation provides the following options:

- Original method (follows the original AP implementations):
For every node we look for neighbors which are cluster centers and assign the node to the nearest one. If depth is greater than 1 then we allow indirectly connected nodes to be assigned to the center (this extends the original method).
- Weighted BSF method (specific to our plugin):
In the case of incomplete graphs (especially sparse graphs), it is possible that none of the node's neighbors will be a cluster center. When this option is selected, we look for shortest paths in the weighted network from nodes to cluster centers.

Directed/undirected

This option tells the program to treat the edges in the graph as either directed or undirected. If "undirected" option is selected, then for each edge, the plugin will create edges in each direction with the same weight. In the directed case the edges will be read as specified (lacking edges in one direction will be assigned similarity equal to $-\infty$).

2.6 Implementations

We have prepared two implementations of the clustering algorithm:

- the first works on a 2D matrix (recommended for dense graphs),
- the second uses sibling lists (for sparse graph representations).