How to Take Advantage of the Limitations with Markov Clustering?

The Foundations of Branching Markov Clustering (BMCL)

Hiroyuki Akama
Tokyo Institute of Technology
W9-10, 2-12-1,
O-okayama, Meguroku,
152-8552 Tokyo, Japan
akama.h.aa@m.titech.ac.jp

Maki Miyake
University of Osaka
Machikane-machi, Toyonaka-shi,
560-0043 Osaka, Japan
mmiyake@lang.osaka-u.ac.jp

Jaeyoung Jung
Tokyo Institute of Technology
W9-10, 2-12-1,
O-okayama, Meguroku,
152-8552 Tokyo, Japan
jung.j.aa@m.titech.ac.jp

Abstract

In this paper, we propose a novel approach to optimally employing the MCL (Markov Cluster Algorithm) by "neutralizing" the trivial disadvantages acknowledged by its original proposer. Our BMCL (Branching Markov Clustering) algorithm makes it possible to subdivide a large core cluster into appropriately resized sub-graphs. Utilizing three corpora, we examine the effects of the BMCL which varies according to the curvature (clustering coefficient) of a hub in a network.

1 MCL limitations?

1.1 MCL and modularity Q

The Markov Cluster Algorithm (MCL) (Van Dongen, 2000) is well-recognized as an effective method of graph clustering. It involves changing the values of a transition matrix toward either 0 or 1 at each step in a random walk until the stochastic condition is satisfied. When the hadamard power for each transition probability value is divided by the sum of each column, the rescaling process yields a transition matrix for the next stage. After repeatedly alternating for about 20 times between two steps—random walk (*expansion*) and probability modification (*inflation*)—the process will finally reach a convergence stage in which the whole graph is subdivided into a set of 'hard' clusters that have no overlap. Although this method has been generally applied in various domains with notable successes (such as Tribe-MCL clustering of proteins (Enright et al., 2002); Synonymy Network, created by the addition of noise data (Gfeller, 2005); and Lexical Acquisition (Dorow et al., 2005)), Van Dongen et al. (2001) frankly acknowledge that there are limitations or weaknesses. For instance, the readme file, which is included with the free MCL software available via the Internet from Van Dongen's group, remarks that "MCL is probably not suited for clustering tree graphs".

It should also be noted, however, that the group has provided no mathematical evidence for their claim of the MCL's unsuitability for hierarchical applications. What prompts this subtle caveat in the first place? Is this a limitation on the type of graph clustering that can employ random walks for spectral analysis? Or, is it difficult for this technique to (re-)form or adjust graph clusters that have already been clustered into a kind of multilayered organization? Such questions are very important when comparing the MCL with other graph clustering methods that employ (greedy) algorithms developed step by step in a tree form.

A tree graph is essentially a kind of dendrogram, which means clustering results can be generated solely by making a cross cut at some height between the root and the leaves. In other words, as there is no horizontal connection at the same level, it is not possible to create triangle circulation paths in a single stroke. However, the graph coefficient known as "curvature" (Dorow, 2005) is appropriate for defining such structures. The curvature, or the cluster coefficient, of a vertex is defined as a fraction of existing links among a node's neighbors out of all possible links between neighbors. Thus, a tree graph may be regarded as a chain of star graphs where all the vertices have a curvature value of 0.

It is certainly true that when a hub has a low curvature value, the corresponding cluster will be less cohesive and more sparse than usual. The modularity Q value is very low in such cases when we try to measure the accuracy of results from MCL clustering. Modularity Q indicates differences in edge distributions between a graph with meaningful partitions and a random graph for identical vertices conditions. According to Newman and Girvan, $Q=\sum_i (e_{ii}-a_i^2)$, where i is the cluster number of cluster c_i, e_{ii} is the proportion of internal links in the whole graph and a_i is the expected proportion of c_i's edges calculated as the total number of degrees in c_i divided by the total of all degrees (2*the number of all edges) in the whole graph. This value has been widely used as an index to evaluate the accuracy of clustering results.

1.2 Karate club simulation

However, it would be an exaggeration to regard Modularity Q is an almighty tool for accurately determining the attribution value of each vertex in a graph cluster. That is only true for modularity-based greedy algorithms that select vertices pairings be merged into a cluster at each step of the tree-form integration process based on modularity optimization criterion. However, such methods suffer from the problem that once a merger is executed based on a discrimination error, there is no chance of subsequently splitting pairings that belong to different subgroups.

This fatal error can be illustrated as follows. Zachary's famous "Karate Club" is often used as supervised data for graph clustering, because the complex relationships among the club members are presented as a graph composed of edges representing acquaintances and vertices coded indicating final attachments to factions. If the results of graph clustering were to match with the actual composition of sects within the club, one could claim that the tested method was capable of simulating the social relationships.

However, the real difficulties lie at boundary positions. It is worth pointing out that the degree of ambiguity is the same (0.5) for both vertices 3 and 10 in Figure I, indicating that they occupy neutral positions while in reality they belong to different subgroups. All modularity-based greedy algorithms would inevitably bind the two nodes at an earlier step in the dendrogram construction (at the first merging step in experiments conducted by Newman and Danon and at the second in Pujol's experiment). In contrast, MCL is one of the rare clustering methods that avoids this type of misjudgment (accurate results for the karate club network were also obtained with the Ward method), even though the modularity Q value for MCL is a little lower (0.371) than values for greedy algorithms (for example, 0.3807 for Newman et al.'s fast algorithm and 0.418 for Danon et al.'s modified algorithm).

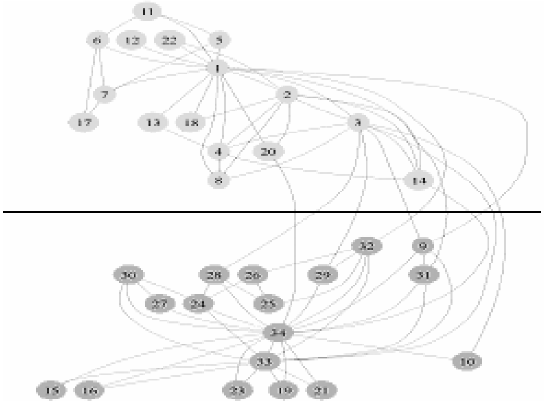
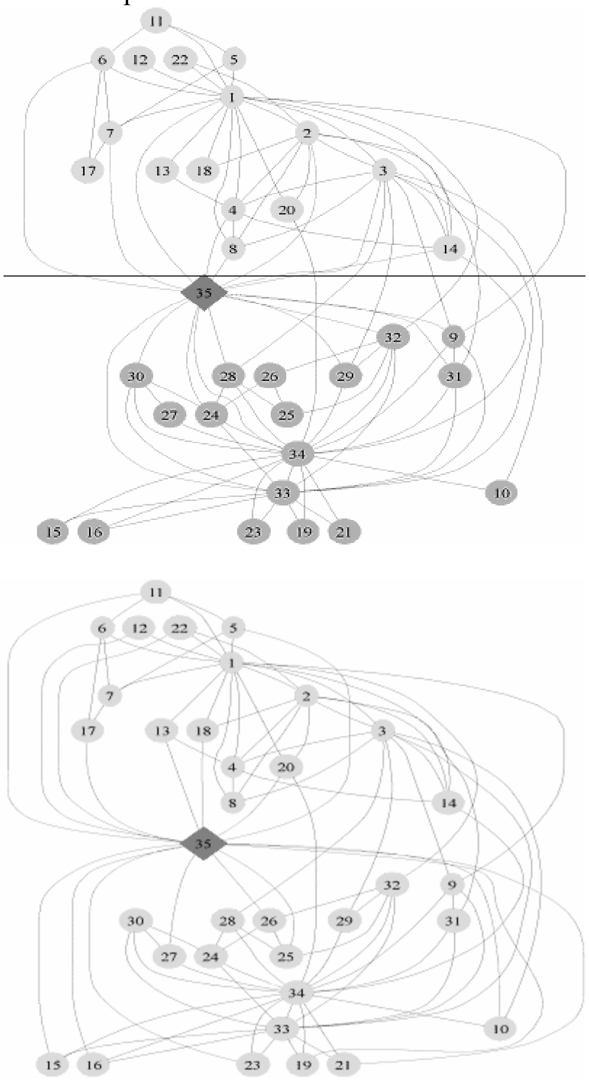


Figure I Karate club

The karate club case suggests the possibility of using both graph clustering and modularity Q from different perspectives. MCL allows us to regard both *clustering* and *discrimination* on the same plan if we do not treat modularity Q as an optimization index but rather as an index of structuring dynamics balancing assembly and division. To the extent that a graph clustering method is evaluated in terms of its effectiveness in a variety of discrimination analyses with learning data extracted from real situations, it should be useful as a simulation tool. For example, it is possible to test with the karate club network the effects of supplementing the network by adding to the original graph another hub with the highest degree value. As the curvature value of this new hub varies according to the selection of vertices which become adjacent to it, we can re-execute MCL for the overall graph to see how curvature is closely related with how it influences clustering results. In general cases, the hub of a whole graph also tends to be the representative node for the large-sized Markov cluster called the "core cluster" (Jung, 2006).

Let us imagine that a highly influential newcomer joins the karate club and tries to contact with half (17) of all the members, functioning as a hub within the network. Even though this is a purely hypothetical situation, it is possible to predict the impact on the network with MCL.



Figures II, III Hub to high or low degree nodes

For example, one could classify the 34 vertices into higher and lower degree subgroups, and set a hub that is adjacent to all vertices for one subgroup but is far from the other subgroup. MCL results would indicate that even when adding a hub with the highest curvature value, it would be ineffectual in preventing a split (Figure II). However, if the newcomer were to be a friend with less sociable members, the club would be saved from being torn apart. A hub connected with the lower degree subgroup, and thus having the lowest curvature value, would become part of the largest core cluster, because the MCL would not subdivide the graph (Figure III). In short, the results of MCL computation hinge on the curvature value of the hub with the highest degree value.

2 The basic concept of BMCL

This connection-sensitive feature of MCL brings us back to the limitations that Van Dogen et al. inform their software users of. Do these limitations really render the MCL unsuitable for tree graphs? Should we not regard a low modularity Q value for a graph as a positive attribute if it is due to the low curvature value for a hub? In a very real sense, these questions are actually asking about the same thing. The point can be clearer if conceived of in relation to a non-directed and cascading type of three-layer graph, as depicted in Figure IV.

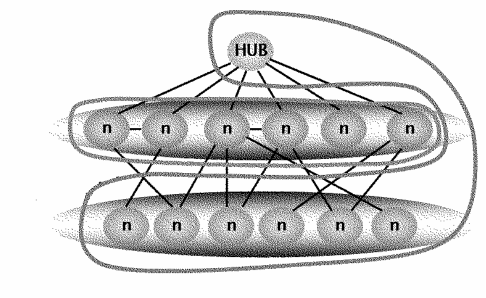


Figure IV Three-layer tree-form network

The root node at the top (the hub) is linked to all the vertices in the intermediate layer but to none at the bottom layer, even though there are moderate levels of connectivity between the layers. Connections within a layer are extremely rare or absent. Clearly, the curvature of the hub would be influenced by the very low connectivity within layer 2.

	0.01	0.02	0.03
0.1	1core cluster & singleton clusters	1core cluster & singleton clusters	1cluster (not divided)
0.15	1cluster or 2 core clusters	1cluster (not divided)	1cluster (not divided)
0.2	2core clusters	2core clusters	1cluster (not divided)

Table I. MCL results for the structured Random Graph

We have executed computations at least 10 times under the same condition in order to generate this type of structured random graph with 500 vertices in the two layers respectively. A random graph was produced by using a binominal distribution. Although between connection rates were varied from 0.1 to 0.2 and within connection rates for the intermediate layer from 0.01 to 0.03, no edges were inserted into the lower layer. MCL results obtained for this architecture are almost constant, as shown in Table I.

In this experiment, all singleton clusters consisted of vertices belonging to layer 2. In cases where the whole graph was split into 2 core clusters, one cluster would correspond to the hub plus layer 3 while the other would correspond to layer 2. There was no exception when the between connection rate was 0.2. This means that, quite curiously, the hub formed a core cluster around itself with vertices that were not all adjacent to it, so that ones that were connected with it in the raw data were all segregated into the other cluster. In this case, the Modularity Q value for each core cluster was zero or extremely low.

Nevertheless, in spite of this inaccuracy, this type of network can easily be by modified by the BMCL method that we discuss later. It can be indirectly subdivided by graph clustering, if inside the same cluster, a latent shortcut is set between one vertex and another. Such a latent connection can be counted in place of a path of length 2 that is traced in the original adjacency as a detour via a vertex of another cluster. If all latent adjacency relationships are enumerated in this way, except for those for the hub, the core cluster will be reclustered by a second application of the MCL to realize a sort of hierarchical clustering (in this case for a quasi-tree graph), which has been regarded as being a limitation with the MCL.

This principle can be called Branching Markov Clustering (BMCL) in the sense that it makes it possible to correct for unbalances in cluster-sizes by dividing large Markov clusters into appropriate branches. In other words, BMCL is a way of rebuilding adjacency relationships "inside" MCL clusters, by making reference to "outside" path information. It then becomes natural to realize that the lower the curvature value of the hub is—reflecting sparse connectivity inside the hub's cluster—the more effective BMCL will be in subdividing the core cluster, which will augment the modularity Q value for the clustering results.

3 Applying BMCL corpora data

3.1 The BMCL algorithm

In this section, we apply our BMCL method to a semantic network that is almost exhaustively extracted from typical documents of a specific structure. It is supposed that if the MCL is applied to word association or co-occurrence data it will yield concept clusters where words are classified according to similar topics or similar meanings as paradigms. However, because the word distribution of a corpus approximately follows Zipf's law and produces a small-world scale-free network (Steyvers et al., 2005), the MCL will result in a biased distribution of cluster sizes, with a few extraordinarily large core clusters that lack any particular features.

In order to overcome such difficulties in building appropriate lexical graphs for corpus data, we propose an original way of appropriately subdividing core clusters by taking into account graph coefficients, especially the curvature of a hub word. As mentioned above, BMCL is most effective for clusters that, containing a high-degree and low-curvature vertex, display a local part of a network with highly sparse connectivity when a hub is eliminated. This feature increases the efficiency of the BMCL by making it possible to introduce moderate connection rates for latent adjacencies.

In contrast to a 'real' adjacency between the vertices i,k represented here by $d(i,k)=1$, the 'latent' adjacency $d_v(i,j)=1$ will subsequently be defined to closely adapt to the connection state for the dataset, which we will utilize in testing the BMCL. The hub M_h of each Markov cluster M is supposed to be the vertex with the largest degree for M . Here, we set a sufficiently large core cluster C, a set of hubs H and the hub of C as C_h .Under such conditions, we can formulize the set of external hubs bypassing the intra-core connections $K_{i,j}$ as;

$\{K_{i,j} \mid K_{i,j} \subset H, k \in K_{i,j}, d(i,k)=d(j,k)=1\}$,

where $i,j \in \underset{i \neq j, i \neq C_h, j \neq C_h}{C}$ $, C_h \notin H$. We also propose an additional function called $\underset{n}{ArgTopn}$, which identifies the set of n nodes that have the highest connection values. This is to produce a moderate connection rate which allows us to execute appropriate MCL operations by appropriately setting two pruning thresholds, θ_{p} and θ_{q}. These are applied in the row direction by fixing i in the intra-core connection matrix to the number of the shortest paths between i, j --$|K_{i,j}|$-- to make the following pruning rule:

$$if(|K_{i,j}| \geq \theta_p \,\&\&\, j \in \underset{n=\theta_q}{ArgTopn} |K_{i,j}|)$$
$$-> d_v(i,j) = 1$$

This rule extracts from the intra-core connection matrix a latent adjacency matrix to which the MCL is applied once again in order to obtain appropriately resized sub-clusters from a huge core cluster.

3.2 A range of corpus data

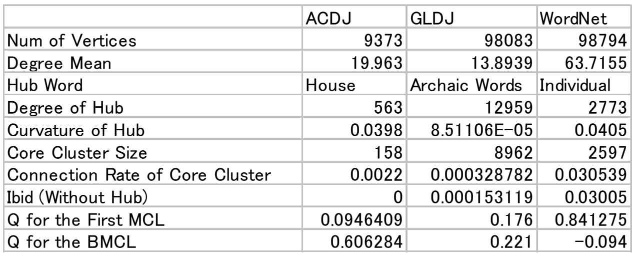
In this section, three documents were selected taking into consideration the curvature value of a hub with the highest degree and the density of connections with or without this hub among the vertices of a core cluster at the level of a raw data graph.

I. Associative Concept Dictionary of Japanese Words (Ishizaki et al., 2001), hereafter abbreviated as ACDJ, which consists of 33,018 words and 240,093 word pairing collected in an association task involving 10 participants. Of these, 9,373 critical words were selected to create well-arranged semantic network by removing the rarest 1-degree dangling words and rarer words with a degree of 2 but curvature values of 0.

II. Gakken's Large Dictionary of Japanese (Kindaichi & Ikeda, 1988), hereafter abbreviated as GLDJ, which is an authoritative Japanese dictionary with some features of an encyclopedia in terms of its rich explanatory texts and copious examples. We selected 98,083 words after removing noise words, functional words, and 1,321 isolated words to extract word pairs by combining every headword with every other headword included within an entry text.

III. WordNet. We used only the "data.noun" file where the lexical information for each noun is defined by a set of index numbers corresponding not with words themselves but with their senses. The co-occurrence relationships for 98,794 meanings were extracted from every data block that contains a series of indexes, which also covers other parts-of-speech.

The principle for building a semantic network for each of these documents was to select relevant 'word pairs' or 'index pairs' indicating the lexical relationships of adjacency, association or co-occurrence, respectively. Table II presents graph information for the three data sets and the results of applying both the MCL and the BMCL to them.



	ACDJ	GLDJ	WordNet
Num of Vertices	9373	98083	98794
Degree Mean	19.963	13.8939	63.7155
Hub Word	House	Archaic Words	Individual
Degree of Hub	563	12959	2773
Curvature of Hub	0.0398	8.51106E-05	0.0405
Core Cluster Size	158	8962	2597
Connection Rate of Core Cluster	0.0022	0.000328782	0.030539
Ibid (Without Hub)	0	0.000153119	0.03005
Q for the First MCL	0.0946409	0.176	0.841275
Q for the BMCL	0.606284	0.221	-0.094

Table II Data about the three corpora

Although the first data (ACDJ) is much smaller, it is worthwhile executing because it represents a concrete example of the network type discussed earlier, namely, a three-layer architecture around a hub (quasi-tree graph). The connection rate in the core cluster is very low (0.002 with and 0 without the hub), as is the modularity Q value for the MCL (0.094). However, subdivision of the core cluster in the BMCL results yielded a high modularity Q value (0.606) when latent adjacencies derived from bypassing connections with a threshold of θ_q =3 were used.

The last two data (GLDJ and WordNet) are directly comparable because they are quite similar in size and provide sharp contrast, particularly in terms of curvature values (GLDJ: 8.51106E-05 << WordNet: 0.0405), and modularity Q values for the MCL (GLDJ: 0.176 << WordNet: 0.841). For WordNet, the high connection rate in the core cluster (0.03) makes it difficult for it to be subdivided by any clustering method, even if the hub is eliminated. In terms of the GLDJ, the core cluster was repeatedly divided by the BMCL and the modularity for the subdivision turned out to be 0.2214 with a threshold of θ_p = 1.

However, there is another way to split the core cluster into sub graphs, which does not require the use of the latent adjacency information which is crucial for the BMCL. That other method, which can be called the 'Simply-Repeated MCL (SR-MCL)', involves applying the MCL once again to the part of the original adjacency matrix that corresponds to the vertices apart from the hub, and which become members of the core cluster as a result of the first MCL. In the case of ACDJ, it is impossible to execute the SR-MCL, because there is no edge that is not connected to the hub within the core cluster, and so all the vertices apart from the hub would be isolated if the hub were removed.

A similar problem is also encountered with the core cluster of the GLDJ, even though the SR-MCL increases the modularity Q value (0.769) much more than the BMCL. Vertices that dangle from the hub—37% of the core members—would be dropped from the second MCL computation if the latent adjacency is not used, which, on the other hand, assures a high recall rate (0.88). Thus, we have adopted an eclectic way to maintain both relatively high *recall* (the proportion of non-isolated nodes) and relatively high *precision* (the modularity Q of the intra-core clustering). This is what we may call a 'Mixed BMCL' which involves combining the latent adjacency matrix exclusively for the vertices dangling to the hub and the raw adjacency part matrix for the remaining ones that are connected among them. As Figure V highlights, the F-measure $\frac{PR}{(1-\alpha)P+\alpha R}$ (R: recall; P: precision) underscores the effectiveness of the Mixed BMCL for the GLDJ.

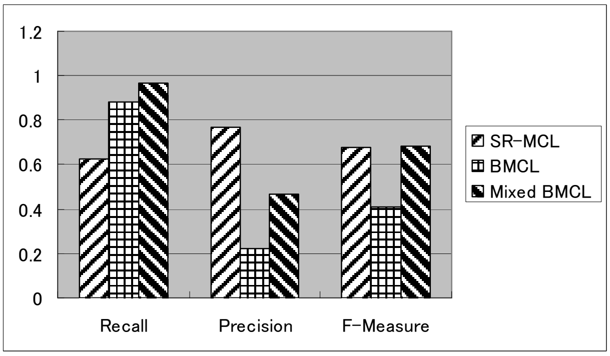


Figure V Comparison of the methods (α =0.4)

4 Conclusion

This paper has examined MCL outputs obtained for some rather problematic conditions, such as the clustering of a tree graph and clustering for a network that contains a hub that has a very low curvature value. In such cases, many of the vertices adjacent to the hub are removed from the cluster that it represents. However, compensating for that, the hub cluster will absorb many other vertices—some of which are not directly connected to the hub itself—to form a large-sized core cluster. That is when our proposed method of Branching MCL (BMCL) is most effective in adjusting cluster sizes by utilizing latent adjacency. Subdivision of the core cluster can facilitate the interpretation of the classified concepts.

When the curvature of the hub is a little higher than in such extreme conditions, the combination of the ordinary MCL and the BMCL (a Mixed BMCL) can work well in increasing the F-Measure score. However, it is not possible to reapply the MCL to a dense core cluster that is organized around a hub with a very high curvature value. A direction for further research will be to automatically select from between the BMCL and the Mixed-BMCL. The SR-MCL or similar modifications may yield the optimal approach to dividing massive Markov clusters into appropriate subsets.

References

Clauset, A, Newman M.E.J., and Moore, C. Finding Community Structure in Very Large Networks, Phys. Rev. E 70, 066111 (2004)

Danon, L., Diaz-Guilera, A., and Arenas, A. Effect of Size Heterogeneity on Community Identification in Complex Networks, J. Stat. Mech. P11010 (2006)

Dorow, B. et al. Using Curvature and Markov Clustering in Graphs for Lexical Acquisition and Word Sence Discrimination, MEANING-2005,2nd Workshop organized by the MEANING Project, February,3rd-4th.(2005)

Kindaichi, H., Ikeda, Y. Gakken's Large Dictionary of Japanese, GAKKEN CO, LTD. (1988)

Newman M. E. J. and Girvan M., Finding and evaluating community structure in networks, Physical Review E 69. 026113, (2004)

Okamoto, J., Ishizaki, S. Associative Concept Dictionary and its Comparison with Electronic Concept Dictionaries, http://afnlp.org/pacling2001/pdf/okamoto.pdf, (2001).

Pujol, J.M., Béjar, J. and Delgado, J. "Clustering Algorithm for Determining Community Structure in Large Networks".Physical Review E 74 (2007):016107

Van Dongen, S. Graph Clustering by Flow Simulation. PhD thesis, University of Utrecht. (2000)