DIMACS Technical Report 2002-05
March 2002

The Split Systems Generated by the Bridges of a Connected Graph

by

Melvin F. Janowitz[1] ,[2]
DIMACS
Rutgers University
Piscataway, New Jersey 08854

[1]Permanent Member and Associate Director
[2]Research supported in part by NSF Grant IIS 01-00921

DIMACS is a partnership of Rutgers University, Princeton University, AT&T Labs-Research, Bell Labs, NEC Research Institute and Telcordia Technologies (formerly Bellcore).

DIMACS was founded as an NSF Science and Technology Center, and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

The bridges of a connected graph G are shown to generate splits of G that are compatible in the sense that they generate a tree. The results are extended to generalized bridges, and the theory is applied to develop new clustering algorithms. These algorithms operate by removing at each level any bridges (or generalized bridges) from the threshold graph at that level before applying any clustering criterion.

1 Notation

A basic knowedge of clustering algorithms will be assumed. Any of the standard texts on the subject will provide the needed background material, and in particular the reference [1] would be more than adequate. One of the most commonly used clustering programs is called single-linkage clustering. It operates by transforming a reflexive symmetric relation into its transitive closure. While this computationally is a very efficient process, it has the defect of grouping together objects that are only remotely related. One way out of this is to first remove from the relations any links that are in some sense weak. Now any reflexive symmetric relation R has associated with it an undirected graph G. The edges of the graph are the pairs xy for which xRy. From the viewpoint of graph theory one could remove the bridges from G and then take the transitive closure of the resulting relation. One could even look for generalizations of bridges and remove those links. This was the original motivation for what follows. It might also be noted that bridges and generalizations of bridges are natural concepts for study in connection with networks, as these are the potential obstructions to traffic flowing through the network.

All graphs will be finite and undirected. Let G be a graph, with $V(G)$ its vertex set and $E(G)$ its edge set. Denote the edge between x and y as simply xy. I will assume a knowledge of the definitions of a *path* and a *cycle*. To say that G is *connected* is to say that there is a path between any pair of vertices. A *connected component* is a maximal connected subgraph of G. Every graph is trivially the disjoint union of its connected components. Let $k(G)$ denote the number of connected components of G, so $k(G) = 1$ for any connected graph G.

We shall have occasion to remove selected edges from G. If xy is an edge, then $G - xy$ is the graph formed with $V(G - xy) = V(G)$ and $E(G - xy) = E(G) - xy$. If two edges are to be removed, we agree to let $G - xy, uv = (G - xy) - uv$. We are now able to call xy a *bridge* if $k(G - xy) = k(G) + 1$. This is of course a well known idea, and can be found in any standard text on graph theory.

A *split* of the connected graph G is a decomposition of G into the union of two disjoint connected subgraphs. The splits (A, A'), (B, B') are called *compatible* if one of $A \cap B$, $A \cap B'$, $A' \cap B, A' \cap B'$ is empty. A moment's thought should convince you that this just says that one of the following must hold:

1. A and B are comparable.

2. A and B' are comparable.

The point is that

$$A \cap B = \emptyset \Longleftrightarrow A \subseteq B'.$$

$A \cap B' = \emptyset \iff A \subseteq B.$

$A' \cap B = \emptyset \iff B \subseteq A.$

$A' \cap B' = \emptyset \iff B' \subseteq A.$

2 Bridges and Splits

We shall indicate how a bridge defines a split of the connected graph G, so we let xy be a bridge of G. Then for any vertex a, there must be a path connecting a and y. Note first that though there may be more than one such path, there cannot be one path P_1 containing x and another one P_2 not containing x. For if this were to happen, we could travel from a to x to y along P_1, and from y back to a along P_2, thus inducing a cycle that contains the bridge xy, a contradiction. So there are exactly two possibilities:

(1) Every path from a to y must pass through x, or

(2) No path from a to y can contain x.

But this defines a split (A, A') of G according to whether the vertex a satisfies (1) or (2). It is much easier to visualize the situation by going back to the definition of a bridge. If xy is a bridge, then $k(G - xy) = 2$, so $G - xy$ is the disjoint union of two connected subgraphs, and this constitutes the desired split. The split consists of the two connected components of $G - xy$. The split (A, A') has the property that any path from a vertex in A to a vertex in A' must contain the edge xy; any path between vertices in A (or A') must lie entirely in A (or A').

Now let xy and uv be two distinct bridges of G. Let (B, B') denote the split associated with xy, and (A, A') the split associated with uv. By symmetry, we may assume that $x \in B$, $y \in B'$, $u \in A$, and $v \in A'$. We clearly cannot have $u \in B$, $v \in B'$ or $u \in B'$, $v \in B$ since either of these events would create a cycle containing xy. Thus with no loss of generality, we may assume that $\{u, v\} \subseteq B$. Similarly, it must be that $\{x, y\}$ is contained in A or A', so we may assume that $\{x, y\} \subseteq A$. If $A' \cap B'$ is not empty we will arrive at a contradiction. We illustrate the situation schematically in Figure 1.

In the diagram, $\{u, x\} \subseteq A \cap B$, $v \in A' \cap B$, $y \in A \cap B'$, and $w \in A' \cap B'$. Since $\{y, w\} \subseteq B'$, there must be a path P_1 between them that lies entirely in B'. Hence P_1 does not pass through x or u. Similarly, since $\{v, w\} \subseteq A'$, there must be a path P_2 that lies entirely in A', so $P2$ also does not pass through x or u. Combining the paths P_1 and P_2 induces a path $P3$ between y and v. But $y \in A \cap B'$, and $v \in A' \cap B$, and the path P_3 cannot contain either of the edges uv or xy. This contradicts the fact that any path from A to A' must contain uv, while any path between B and B'

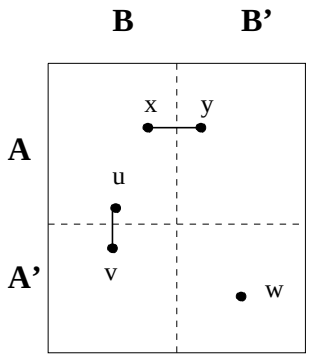

Figure 1: Two Bridge Splits

must contain xy. This then shows that $A' \cap B' = \emptyset$. In short, we have established the next Theorem.

Theorem 2.1 *The splits defined by the bridges of any connected graph are pairwise compatible.*

Example 2.2 We illustrate the connection between the structure of G and of the tree induced by its splits. In Figure 2, the upper part of the display is the underlying graph G, and the lower part is the induced tree.

3 Generalized Bridges

Interestingly enough, the same sort of behavior is exhibited by edges that are not quite bridges.

Definition 3.1 Let G be a graph. We say that a pair of edges $<ab, st>$ is a *generalized bridge* if neither ab nor st is a bridge, but ab is a bridge of $G - st$. This is evidently equivalent to st being a bridge of $G - ab$. By dropping down to a connected component of G, there is no loss in generality in assuming that G itself is connected.

Lemma 3.2 *Let xy be a bridge of the connected graph G. If (A, A') is the split defined by xy, and if $cd \in E(G)$ is not a bridge, then (A, A') is also the split defined by xy in $G - cd$.*

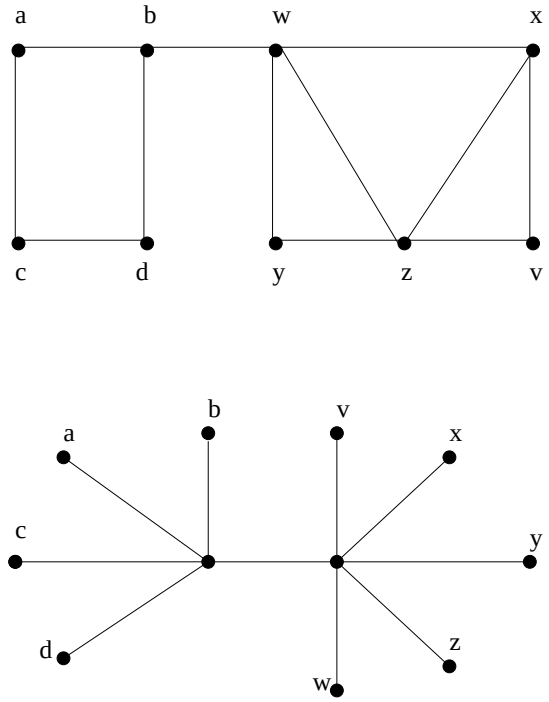

Figure 2: Induced Tree

Proof: We are given that xy is a bridge with induced split (A, A'). Since cd is not a bridge, $G - cd$ is connected. Since xy is the only edge connecting A with A', it must be that for $cd \in V(G)$ with $cd \neq xy$ and cd not a bridge, then either $\{c, d\} \subseteq A$ or $\{c, d\} \subseteq A'$. In either event, removal of the edge cd does not change the connectivity or the vertices in A or A'. Thus (A, A') is the split defined by xy in $G - cd$. ∎

Lemma 3.3 *The splits induced by a bridge and a given generalized bridge of any connected graph G are compatible.*

Proof: Let ab be a bridge and $<cd, xy>$ a generalized bridge of G. Then ab is a bridge of $G - xy$, and by Lemma 3.2, the split of G induced by ab coincides with the split it induces on $G - xy$. Dropping down to $G - xy$, we may apply Theorem 2.1. ∎

Lemma 3.4 *The splits induced by the generalized bridges $<ab, xy>$ and $<cd, xy>$ are compatible.*

Proof: This is immediate upon applying Theorem 2.1 and Lemma 3.3 to $G - xy$. ∎

Example 3.5 At this point it is natural to ask whether the splits induced by a pair of generalized bridges must be compatible. It is easy to see that this is false, For example, if $V(G) = \{a, b, c, d\}$, and $E(G) = \{ab, bc, cd, da\}$, then any pair of edges is a generalized bridge. The splits $<ab, cd>$, $<ac, bd>$ are each induced by generalized bridges, but are not compatible.

Question: When do pairs of generalized bridges induce compatible splits? The remainder of this section will be devoted to answering this question.

Definition 3.6 The generalized bridges $<ab, st>$ and $<cd, xy>$ are called *orthogonal* if the edges ab, st, cd, and xy are all distinct, and none of the pairs (st, cd), (st, xy), (ab, cd), (ab, xy) is a generalized bridge.

Lemma 3.7 *Orthogonal pairs of generalized bridges induce compatible splits.*

Proof: Let $< ab, st >$ and $< cd, xy >$ be an orthogonal pair of generalized splits. Recall that $H = G - st, xy$ is just G with the edges $\{st, xy\}$ removed. Note that ab is a bridge of $G - st$, and cd is a bridge of $G - xy$. It is immediate that both ab and cd are bridges of H. By Lemma 3.2, the split induced by $<ab, st>$ in H coincides with the one it induces in $G - st$; similarly, the split induced by $<cd, xy>$ in H coincides with the one it induces in $G - xy$. By Theorem 2.1, the two splits are compatible. ∎

Remark 3.8 It is easy to see that if $<ab, cd>$ is a generalized bridge if and only if ab is on a cycle, and cd is on every cycle containing ab. There is an easy consequence of this. If $<ab, cd>$ and $<cd, xy>$ are generalized bridges, let C be any cycle containing ab. Then cd is on C, and from this it follows that xy is on C. But this tells us that $<ab, xy>$ is a generalized bridge, Indeed, $<ab, cd>$ is a generalized bridge if and only there is a cycle containing both of them, and the set of cycles containing ab coincides with the set containing cd.

Note that if (A, A') is the split defined by the generalized bridge $<ab, st>$, then A and A' are the connected components of the graph $G - ab, st$. Working in G, if $g \in A$ and $h \in A'$, there is a path P_1 between g and h containing the edge ab but not st, and a second path P_2 containing st but not ab. This of course induces a cycle containing ab and st.

Lemma 3.9 *Let $<ab, st>$ be a generalized bridge whose associated split is (A, A'). Then A is completely determined by its restriction to any cycle containing the edge ab.*

Proof: Let $C = \{x_0, x_1, \ldots, x_n\}$ be a cycle containing ab, and note that st is necessarily an edge of C. Let $y \in V(G)$. If the path P from y to x_i contains no other members of C, then P is called a path from y to C, or a *leg* of C. Evidently there is a path P from any vertex y to some $x_i \in C$. Clearly any leg of C must lie entirely in A or entirely in A', since it cannot contain the edges ab or st. It follows that $y \in A \Longleftrightarrow x_i \in A$. ∎

The situation is summarized in Theorem 3.10.

Theorem 3.10 *For the edges ab and st of the connected graph G, the following are equivalent:*

(1) *$<ab, st>$ is a generalized bridge.*

(2) *There is a cycle containing ab and st is on every such cycle.*

(3) *There is a cycle containing st and ab is on every such cycle.*

(4) *The set of cycles containing ab is not empty and coincides with the set of cycles containing st.*

The next Corollary follows now from repeated applications of Theorem 3.10.

Corollary 3.11 *If $\simeq$ on $E(G)$ is defined by $ab \simeq st$ if $<ab, st>$ is a generalized bridge or if $ab = st$, then $\simeq$ is an equivalence relation.*

Corollary 3.12 *If $<ab, st>$ and $<cd, xy>$ are generalized bridges, and if the edges ab, st, cd, xy are all distinct, then each of the assertions $ab \simeq cd$, $ab \simeq xy$, $st \simeq cd$, $st \simeq xy$ implies the remaining assertions.*

It follows from Corollary 3.11 that if $<ab, st>$ and $<cd, xy>$ are non-orthogonal generalized bridges, then the edges ab, st, cd, xy all lie on the same cycles, and there is at least one cycle containing all of them. In fact, in view of Lemma 3.9, the issue of whether the splits (A, A'), (B, B') defined by $<ab, st>$ and $<cd, xy>$ are compatible can be resolved entirely by whether their restriction to a cycle containing ab (and consequently st, cd, xy) are compatible.

In view of this, it seems appropriate to see what happens if G itself is a cycle. Before looking at the general case, we consider a specific example.

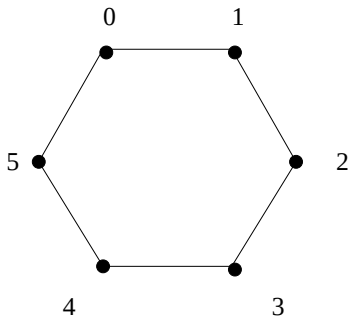

Figure 3: A Six Member Cycle

Example 3.13 Here is an example to consider. There are 15 generalized bridges, and the next table lists them all.

	Gen. bridge	Split
1.	<01, 12>	1 23450
2.	<01, 23>	12 3450
3.	<01, 34>	123 450
4.	<01, 45>	1234 50
5.	<01, 50>	12345 0
6.	<12, 23>	2 34501
7.	<12, 34>	23 4501
8.	<12, 45>	234 501
9.	<12, 50>	2345 01
10.	<23, 34>	3 45012
11.	<23, 45>	34 5012
12.	<23, 50>	345 012
13.	<34, 45>	4 50123
14.	<34, 50>	45 0123
15.	<45, 50>	5 01234

Here is the compatibility table for the generalized bridges. An X shows compatibility, and a 0 shows that the entry is incompatible.

Gen. bridge	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
1.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2.	X	X	X	X	X	X	0	0	0	X	X	X	X	X	X
3.	X	X	X	X	X	X	X	0	0	X	0	0	X	X	X
4.	X	X	X	X	X	X	X	X	0	X	X	0	X	0	X
5.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
6.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
7.	X	0	X	X	X	X	X	X	X	X	0	0	X	X	X
8.	X	0	0	X	X	X	X	X	X	X	X	0	X	0	X
9.	X	0	0	0	X	X	X	X	X	X	X	X	X	X	X
10.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
11.	X	X	0	X	X	X	0	X	X	X	X	X	X	0	X
12.	X	X	0	0	X	X	0	0	X	X	X	X	X	X	X
13.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
14.	X	X	X	0	X	X	X	0	X	X	0	X	X	X	X
15.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

By Lemma 3.9, note that we could resolve the issue of compatibility for generalized bridges if we could just resolve it for cycles. Let G be a cycle with vertices $0, 1, 2, \ldots, n$ and edges $(0,1), \ldots, (i, i+1), \ldots, (n, 0)$. To simplify the notation, denote the edge $(i, i+1)$ by $\hat{i}$, and $(n, 0)$ by $\hat{n}$. Then the generalized bridge with edges $(0, 1)$, $(i, i+1)$ can be represented by $<\hat{0}, \hat{i}>$. So here is the setup.

$<\hat{0}, \hat{i}>$ is a generalized bridge with split (A, A'),

so $A = \{1, 2, \ldots, i\}$, and $A' = \{i+1, \ldots, n, 0\}$.

$<\hat{j}, \hat{k}>$ with $j < k$ is a generalized bridge with split (B, B'), where $B = \{j+1, j+2, \ldots, k\}$ and $B' = \{k+1, k+2, \ldots, n, 0, \ldots, j\}$.

(A, A') and (B, B') fail to be compatible iff $1 \leq j < i < k \leq n$.

Using this and Lemma 3.9 now produces the characterization of when generalized bridges produce incompatible splits. Intuition can be obtained by realizing that we may restrict our attention to a single cycle, and referring to Figure 3.

Theorem 3.14 *Let $<ab, st>$, and $<cd, xy>$ be generalized bridges whose splits are, respectively (A, A') and (B, B'). The following are equivalent:*

(1) *The splits are incompatible.*

(2) *$ab \simeq cd$, with $c, d \in A$ and $x, y \in A'$ or $x, y \in A$ and $c, d \in A'$.*

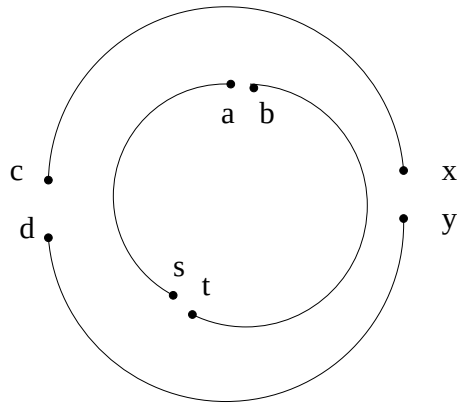

Figure 4: Schematic Illustration of Incompatible Splits

(3) $ab \simeq cd$, *with* $a, b \in B$ *and* $s, t \in B'$ *or* $s, t \in B$ *and* $a, b \in B'$.

Example 3.15 This example involves the upper graph in Figure 2. Though there is only one bridge bw, there are several generalized bridges. We illustrate the tree that results from considering this bridge and the splits resulting from the generalized bridges $<ab, cd>$ and $<wy, yz>$. Here are the three splits:

bw	$\{a, b, c, d\}$	$\{v, w, x, y, z\}$
$<ab, cd>$	$\{a, c\}$	$\{b, d, v, w, x, y, z\}$
$<wy, yz>$	$\{y\}$	$\{a, b, c, d, v, w, x, z\}$

The associated tree is displayed in Figure 5. The internal vertices are denoted by large circles, while the edges between internal vertices are indicated by dashed lines and represent the edges whose deletion produces the three splits displayed above.

Remark 3.16 Here is another possible version of a generalized bridge. One could call ab a generalized bridge if the set of cycles containing ab has a smallest member. One could even further weaken the notion of a bridge by decreeing that for the positive integer k, ab is a k-weak bridge if the removal of k edges from G makes ab into a bridge, but the removal of $k - 1$ edges does not.

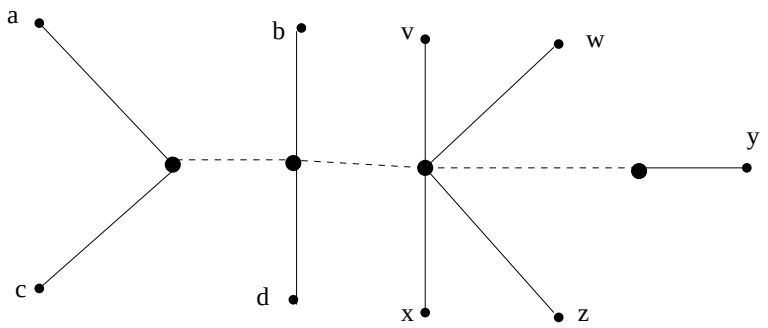

Figure 5: Tree Induced By Three Splits

4 Examples from Cluster Analysis

One motivation for looking at bridges and generalized bridges comes from cluster analysis. We assume here a basic knowledge of the subject – at least as it applies to single linkage clustering. Any of the standard texts on cluster analysis (including [2]) will provide the needed background. We shall look at low cardinality examples so that the results can be hand checked.

Example 4.1 This example is taken from [2], p. 217. We have selected eight points from the original sixteen Sneath and Sokal presented. The eight points all lie in the Euclidean plane so there is a natural geometric visualization (See Figure 6). We will use squared Euclidean distance as the DC. The outputs are not terribly informative because the clusters in the data are by their nature rather long and straggly and these are not what the bridge methods are seeking. Here then is the data.

	a	b	c	d	e	f	g	h
x	0	0	1	2	3	2	2	1
y	4	3	5	4	3	2	1	0

And here is the dissimilarity coefficient matrix.

δ	b	c	d	e	f	g	h
a	1	2	4	10	8	13	17
b		5	5	9	5	8	10
c			2	8	10	17	25
d				2	4	9	17
e					2	5	13
f						1	5
g							2

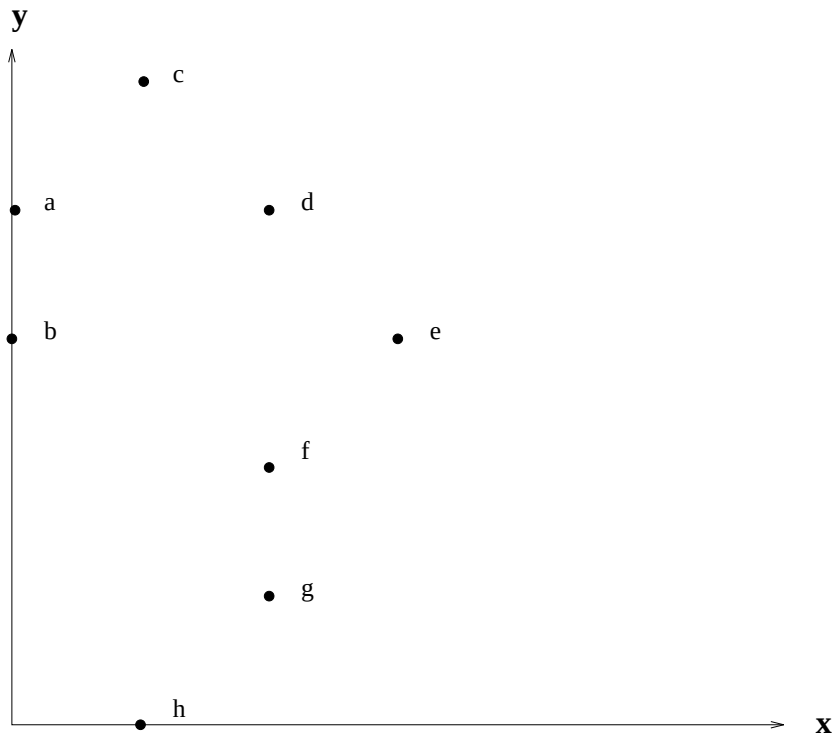

Figure 6: Eight point example

Figure 7 illustrates the threshold graphs at levels 4 and 5.

Here briefly is how single-linkage clustering works. For each $h \geq 0$, we let $R_h = \{(a, b) : \delta(a, b) \leq h$. Viewed as a graph (often called a *threshold graph*), we form the edge xy if x and y are first linked at a level at most h. The output relation at level h is the transitive closure of R_h. We may associate a dissimilarity coefficient with the output by associating with each pair (x, y) the lowest level at which x and y get grouped. Here, for example is the output for single linkage clustering for the eight point example.

$\delta_s l$	b	c	d	e	f	g	h
a	1	2	2	2	2	2	2
b		2	2	2	2	2	2
c			2	2	2	2	2
d				2	2	2	2
e					2	2	2
f						1	2
g							2

The next output was produced by removing any bridges from each R_h, and then

taking the transitive closure. The idea for doing this was originally due to R. Wille, and occurs in [3]. Here then is this output.

δ	b	c	d	e	f	g	h
a	5	4	4	4	4	5	5
b		4	4	4	4	5	5
c			4	4	4	5	5
d				4	4	5	5
e					4	5	5
f						5	5
g							5

Finally here is the result removing all bridges and all generalized bridges.

δ	b	c	d	e	f	g	h
a	5	5	5	5	5	5	10
b		5	5	5	5	5	10
c			5	5	5	5	10
d				5	5	5	10
e					5	5	10
f						5	10
g							10

Single-linkage clustering illustrates a type of clustering algorithm that takes as its input a family $\{R_{h_i} : h_0 < h_1 \cdots h_n\}$ of threshold graphs, and converts each R_h into an equivalence relation E_h. Such a method is called agglomerative if at level h_i it only considers the inputs from levels $h_0, h_1, \ldots, h_i$. Agglomerative cluster methods often suffer from the fact that at low levels of h_i, clusters are formed that are based on insufficient evidence, and once formed they necessarily persist into the final output. By removing bridges and generalized bridges, these clusters are often suppressed.

Example 4.2 Here is a second example. It is taken from the ClustanGraphics software package offered by Clustan Limited. The data consists of a chemical analysis analysis of the milk of 25 different species of mammal. The analysis consists of the percentage of water, protein, fat, lactose and ash, so there are five attributes for each case. Each attribute standardized to produce z-scores and a dissimilarity coefficient was formed using Euclidean distance. Let's begin by presenting a typical clustering output. The clustering was done using both single linkage and complete linkage clustering. In Figure 8, the left tree is from single linkage, while the right one is complete linkage clustering. It is worth noting the rather different outputs.

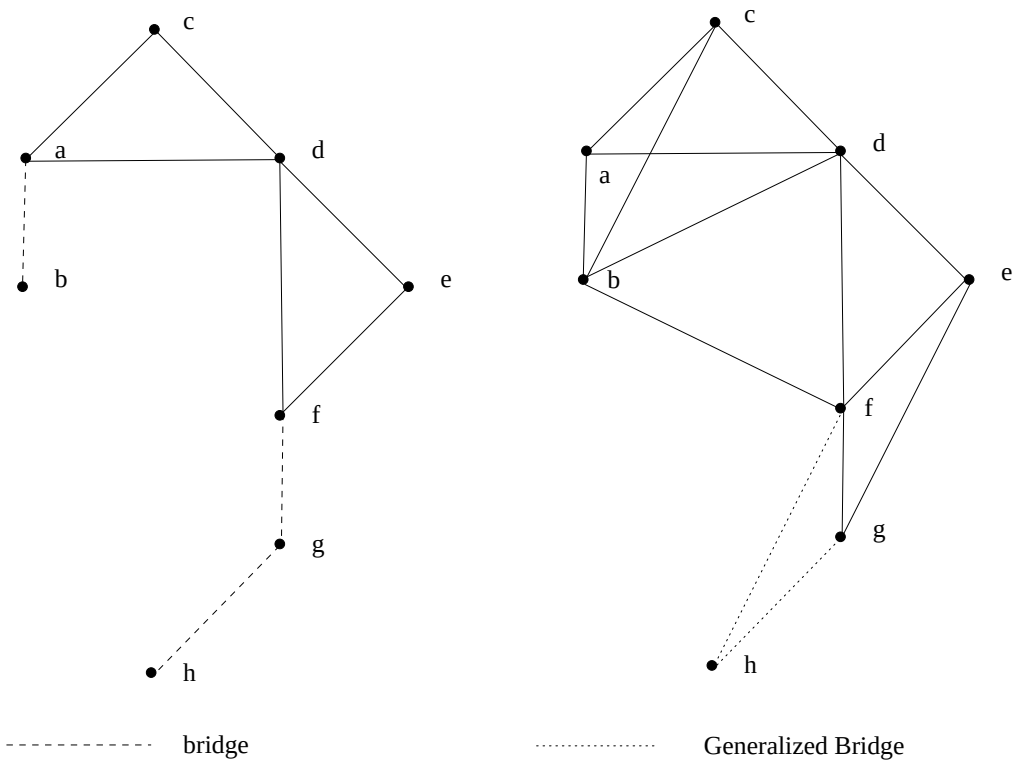

Figure 7: Levels 4 and 5 Threshold Graphs

Let's concentrate on the first four cases, and refer to the complete linkage output. Bison and Llama cluster at Level 0.025, as do Camel and Zebra. These two clusters then merge at level 0.108. This information comes from the Clustan software. But let's look at the dissimilarities for these four objects.

	Bison	Llama	Camel	Zebra
Bison	0	0.0248	0.108	0.107
Llama		0	0.049	0.030
Camel			0	0.025
Zebra				0

If one looks at the average distance of each case from the remaining three, one can argue that the merger of Bison with the remaining three cases should be deferred. This is precisely what the bridge methods do. (Bison, Llama) is a bridge till level 0.108. Thus the cluster (Llama, Camel, Zebra) forms at level .049, and the next merger takes place at level .108. To illustrate what effect the bridge methods have, we hand calculated the effect of removing the bridges from the data in the first cluster of the complete linkage output. Thus the bridge methods tend to compensate for

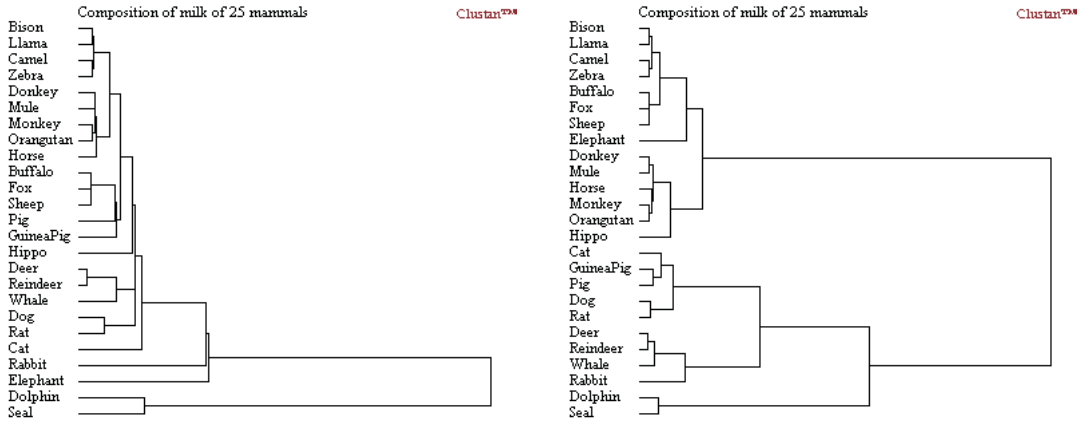

Figure 8: Single and Complete Linkage Clustering on Mammal Milk Data

any premature formation of low level clusters by agglomerative algorithms. Bridge methods should be used with caution, and only as a comparison with other more standard cluster algorithms. Note that there are other possible approaches to this type of problem. Similar results may be obtained by taking as the dissimilarity between cases i and j the lowest level at which a triple of the form $\{i, j, k\}$ is connected.

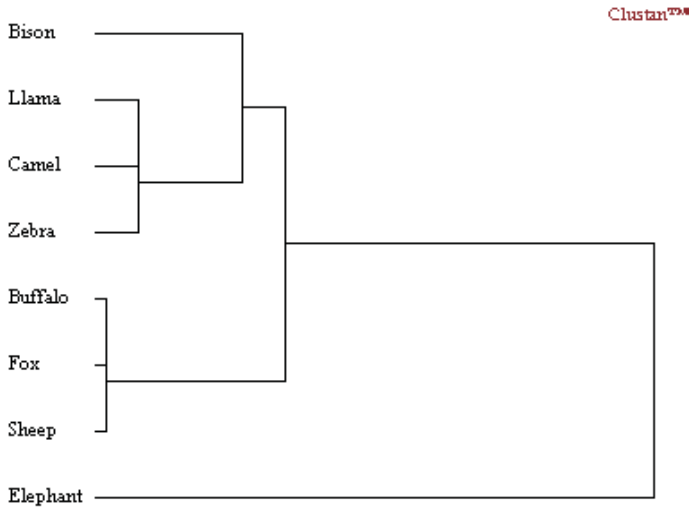

Figure 9: Bridge Method Output for Eight Mammals

References

[1] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, 1988.

[2] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy*, W. H. Freeman, San Francisco, 1973.

[3] M. F. Janowitz and R. Wille, *Classification of Cluster Methods*, in progress.