CS195Z: Problem Set 3

Solution Key
Questions: email Crystal (clkahn)

April 23, 2009

1. (a) Two identical permutations (i.e. that exhibit reversal distance 0) have 0 breakpoints. Given two permutations with b breakpoints, we can transform one permutation into the other by a series of reversals in which each reversal decreases the total number of breakpoints by no more than 2. Therefore, $b/2$ gives us a lower bound on the reversal distance between two permutations.

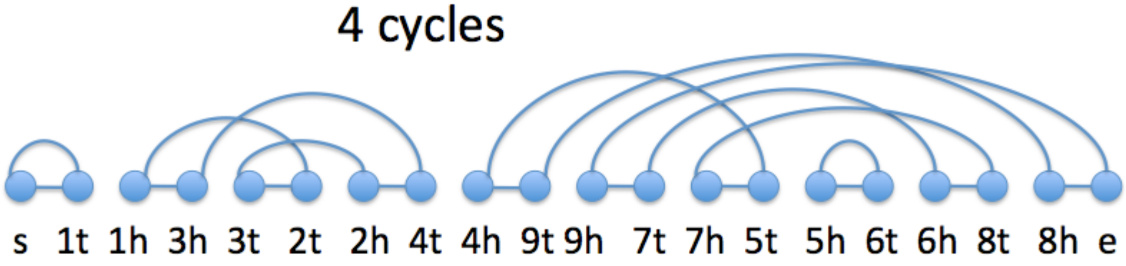
   The two permutations given exhibit 7 breakpoints (only the +5 +6 adjacency is shared between the two permutations). Therefore, the distance is at least $\lceil 7/2 \rceil = 4$.

   (b) A lower bound on the the reversal distance is given by:

$$d(\pi) \geq n + 1 - c(\pi), \tag{1}$$

   where $c(\pi)$ is the number of cycles in the cycle decomposition and n is the length of the permutations.

   The cycle decomposition is:



   We can see that $c(\pi) = 4$ and $n = 9$, so $d(\pi) \geq 9 + 1 - 4 = 6$.

2. As b increases, $\Gamma(b)$ also increases (until it reaches n). Consider $n = 3$.

b	$\Gamma(b)$
0	1
1	2
2	3
3	0

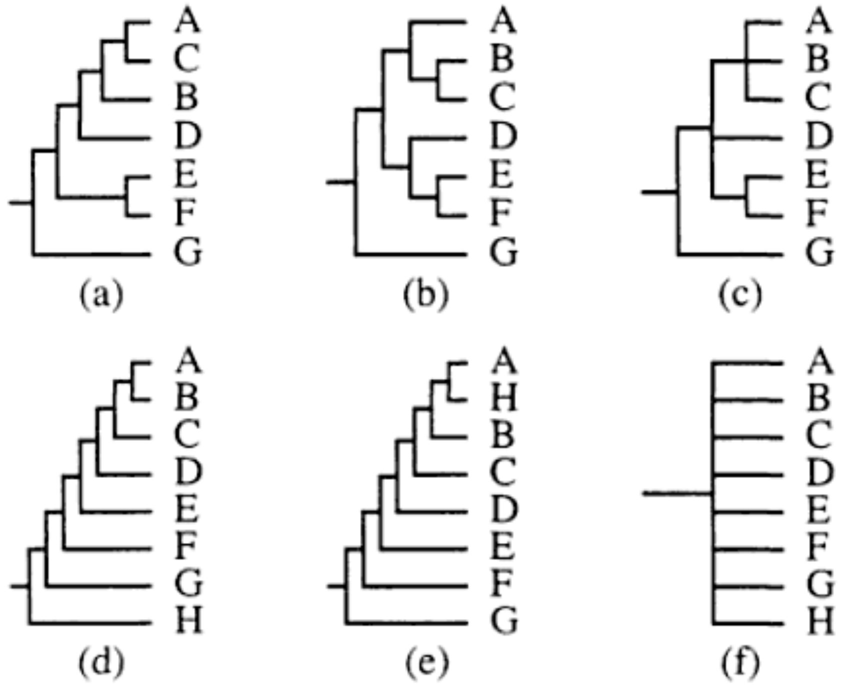
Suppose we want to build a permutation with a given number b of breakpoints. There are $\binom{n-1}{b}$ configurations for where in the permutation we might place a breakpoint. Suppose we choose to place a breakpoint between indices i and $i+1$. Then, having placed integer i, we have $n - i - 1$ possible choices for the next character. If we had no breakpoint between i and $i+1$ then we would have only 1 choice for the next character.

Therefore, as b increases, we have greater freedom in choosing how to construct a permutation with b breakpoints, making $\Gamma(b)$ also greater.

3. There are many possibilities for how the CAST algorithm may not converge. A valid example must exhibit some node(s) that is a member of a cluster C_i and for which its affinity to C_i is less than its affinity to some other cluster C_j. A drawing of a graph with a verbal description (i.e. not necessarily containing a full distance matrix) will suffice.

4. (a) The strict consensus tree for (a) and (b) is given in (c).

   (b) The strict consensus tree for (d) and (e) is given in (f).



(a) (b) (c)

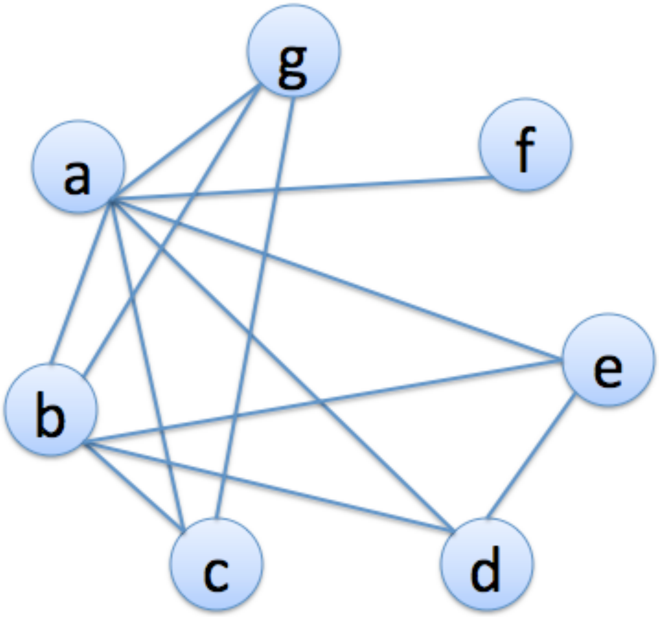
(d) (e) (f)

   (a) Give the strict consensus tree for rival trees (a) and (b).

   (b) Give the strict consensus tree for rival trees (c) and (d).

5. The strict and combinable-component consensus methods are equivalent when there are exactly 2 rival trees and they are fully dichotomous, in which case any unreplicated group will be noncombinable with at least one group appearing on the other tree.

6. The graph at the first iteration is connected, therefore, the algorithm will output `fail`.



7. (a) $v_{g_2} : (s_4, s_1, s_2, s_3) \rightarrow (+, -, +, -)$, score: 1
$v_{g_3} : (s_1, s_3, s_2, s_4) \rightarrow (-, -, +, +)$, score: 0
$v_{g_4} : (s_1, s_2, s_3, s_4) \rightarrow (-, +, -, +)$, score: 1
$v_{g_5} : (s_3, s_1, s_2, s_4) \rightarrow (-, -, +, +)$, score: 0

(b) The number of vectors of length $p + q = m$ that contain exactly p +'s (or equivalently q -'s) is: $\binom{p+q}{p} = \binom{m}{p} = \binom{m}{q}$.

8. (a) *Proof.* Let $\pi(i)$ be the position in sample i. Then $\pi(i) = p(i) - q(i)$. Let's assume that on the left there should be '+'. This leads to a TNoM(i) = q(i) + p - p(i) = p - $\pi(i) \leq$ s. Therefore $\pi(i) \geq$ p - s. The same could be done assuming there should be '-' on the left, yielding $\pi(i) \leq$ s - q. □

(b) The reflection principle states that there is a one-to-one correspondence and onto mapping between the paths that start with (0,0) and end at (p+q,p-q) and cross p-s and the paths that start with (0,2*(p-s)) and end in (p+q,p-q).

The number of paths starting with (0,0) and ending at (p+q,p-q) is $\binom{p+q}{p}$. The number of paths starting at (0,2*(p-s)) and ending at (p+q,p-q) is the same as the number of paths from (0,0) to (p+q,p-q-2*(p-s)), which is equal to $\binom{p+q}{s}$. The same analysis can be done for the lower bound s-q. However doing this some paths are counted twice. The paths that are counted twice are those that cross both bounds. Counting these paths is not possible. However, we can count the path that first crosses the higher bound and then crosses the lower bound or vice versa. But again, doing so will not suffice because the paths can recross the bounds again and be recounted. This leads us to the full probability theorem. However one does not need all the arguments for the number of recrosses. Since the path is p+q long and in order to recross after a first cross the path will grow in at least p-s-s+q = p + q -2s then the maximum number is bounded.