The Curse in the Mailbox: The Message of Chain Letters

John Burkardt
Department of Scientific Computing
Florida State University

....

Date to be Determined
Graduate Student Seminar ISC5934

....

http://people.sc.fsu.edu/~jburkardt/presentations/...
chain_2014_fsu.pdf

The envelope in your mail box contains a single photocopied page, with a peculiar message:

```
1    This Prayer has been sent to you and has been around
2    the world four times.  The one that breaks it will
3    have bad luck.
4
5    Please send this and four copies to people you
6    wishes good luck. It must leave in 24 hrs. Don't
7    send money and dont keep this copy. Gen Patton
8    received $1,600 after receiving it.
9
10   You are to have good luck in 4 days. This is not a
11   joke and you will receive by mail.
```

You have just received a **chain letter**.

Will you break the chain? What do most people do? Why?

The fundamental feature of the chain letter is the idea that it connects a sequence of people, one after the other, who innocently receive it, and then purposely copy it and pass it on.

There are very many peculiar things about chain letters, but this idea of transmission and replication is the most fascinating, and can be summed up in a phrase:

People are a chain letter's way of making more chain letters.

Where have we heard this idea before? Diseases, for instance, are passed along and amplified by people, as are jokes, fads, habits, accents, and even DNA.

Why don't we simply toss a chain letter into the trash? Actually, many people do. I don't know who starts a chain letter (someone must!) but most such letters don't get far.

However, some chain letters seem to have discovered ways to avoid death. If a chain letter is successful, it can be around for years, or decades, and researchers, who collect and date copies that they "capture", are even able to observe the letters evolving, that is, improving their chances of reproduction because of changes in their text.

To begin with, the chain letter has to get the recipient to read it!

A typical chain letter is one page in length; some chain letters have even come on post cards. Keeping the message short increases the chances that the recipient will read the message.

There are several techniques for hooking the recipient:

- **money**: offer a monetary reward;
- **luck**: promise luck (good or bad) to the superstitious;
- **joke**: tell a story or joke worth passing on further;
- **heart**: claim to be part of a worthy cause or charity;

THE TASK!

It's necessary to hook the recipient, because the chain letter needs a task to be carried out:

- copy the letter (before copy machines this was not easy!);

- select new recipients;

- address, stamp and post the letters.

The recipient, if susceptible, carries out the task, hoping to receive some cash, or good luck, or to feel the warmth of a being part of a charitable deed, or simply to expect the new recipients to enjoy the joke or story that has been passed on.

Of course the chain letter doesn't care about any of that, because it has accomplished its goal of being reproduced, and its clones are already in the mail, ready to invade a new set of hosts!

Copies of this letter were still being received 40 years later. Especially in the beginning (before Xerox copiers, and then email) a person had to write the copies by hand, or typewrite them. A copy received after 40 years had probably been copied 4,000 times!

All copies of the letter can be traced back to one person, the original sender. But any two copies of the letter may come from a single sender much more recently.

The version that was seen in the 1980's was very interesting, because it mentioned several people by name. However, because of carelessness in copying, these names varied from copy to copy. The copying errors make it possible to try to cluster the chain letters. Letters in which the names do not differ much probably came from the same person not too long ago. If letters have many differences, their common ancestor must be further back.

The names in the chain letter are useful because...they are not important! They can change without the letter losing its meaning.

There is a similar situation in DNA. Substantial amounts of DNA are the same for all animals because it contains vital information that can't easily be changed without harming the animal.

But there is much genetic material that doesn't do anything. It's called **junk DNA**. Since it's useless, it can change and the animal will pass on the mutation. Thus, junk DNA can be used to track relatedness of individuals.

The names in the chain letters are like junk DNA. As we watch them change, we can try to guess the pattern of ancestry!

Here are some key points in the chain letter:

1. The original is in **New England**.
2. An **R.A.F.** Officer received $470,000.
3. In the Philippines, **Gene Welch** lost his wife...
4. The chain...was written by **Saul Anthony de Groda**...
5. Do note the following: **Constantine Dias**...
6. **Carlos Daddit**, an office employee...
7. **Dalan Fairchild** received the message...
8. In 1987, the message received by **a young woman in California**...

Chain Letters: Extracting the "Markers"

To compute the distance between two chain letters, first locate the 8 names that show up in the positions described on the previous slide.

We are going to compute the distance between the two chain letters based on their agreements on these 8 points.
We will assign a score for agreement or disagreement on each point, and then sum to get the distance.

When we say **New England**, we don't mean the chain letters must both have that particular word. We are instead trying to suggest the position in the letter that must be checked. So we mean, do the chain letters both have the same word there (perhaps **New Zealand!** or not.

1. **New England**, 0 if the same, 1 otherwise
2. **R.A.F.Officer**, 0 if the same, 1 otherwise.
3. **Gene Welch**, 0 if the same, 1 if one name different, 2 if both different.
4. **Saul de Groda**, 0 if the same, 1, 2 or 3 if 1, 2 or 3 names different or added
5. **Constantine Dias**, 0 if the same, 1 or 2 if 1 or 2 names different
6. **Carlos Daddit**, 0 if the same, 1 or 2 if 1 or 2 names different
7. **Dalan Fairchild** 0 if the same, 1 or 2 if 1 or 2 names different
8. **a young woman in California**, 0 if both letters have or don't have this part, 5 if one does and one doesn't

You will be assigned one of the 11 chain letters, which are labeled **A** through **K**.

Compute the distance of your chain letter to all the other chain letters. Do this by computing the pairwise score of your chain letter against the others.

When you are done, you will have computed 11 numbers, which represent one row of the distance matrix. When I have collected all the rows from you, I can do a single linkage clustering.

This assignment is due by Monday, 21 September. You can submit it to Scholar or turn it in on Monday.

Chain Letter Clustering Results

I collected the estimates for the distances between pairs of chain letters into a single distance matrix. Some rows were estimated by more than one person, and row A was *very* popular.

As I was entering the data, more was submitted, so I decided that I needed to set this calculation up in a way that would be easy to correct and modify...you guess it, more MATLAB!

I needed a **dist** matrix, and a **count** vector which would keep track of how many people estimated the same row, so I could average.

I thought it would also be good to average the matrix and its transpose, so that it was symmetric.

```
a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;
dist = zeros(11,11);
count = zeros(11,1);
count(g) = count(g) + 1;
dist(g,1:11) = dist(g,1:11) ...
  + [ 12, 18, 14, 12, 16, 11,  0, 16, 11,  9, 12 ];
count(a) = count(a) + 1;
dist(a,1:11) = dist(a,1:11) ...
  + [  0,  9,  9, 10,  9,  7, 16,  9, 14, 13, 11 ];
count(c) = count(c) + 1;
dist(c,1:11) = dist(c,1:11) ...
  + [ 12,  9,  0, 11,  9,  7, 17, 11, 14, 15, 11 ];
 (and so on)
```

```
% Average the rows
  for row = a : k
    if ( 0 < count(row) )
      dist(row,a:k) = dist(row,a:k) / count(row);
    end
  end
% Make the matrix symmetric
% (I could have done this with one command.  What is it?)
  for row = a : k
    for col = a : row - 1
      rc = dist(row,col);
      cr = dist(col,row);
      dist(row,col) = ( rc + cr ) / 2.0;
      dist(col,row) = ( rc + cr ) / 2.0;
    end
  end
```

Chain Letter Clustering Results: The Table

   The result of our comparisons is similar to a city-to-city mileage chart, that suggests which chain letters are close, based on the markers we chose.

	A	B	C	D	E	F	G	H	I	J	K
A	0	9	9	9	7	7	13	10	14	12	9
B	9	0	9	7	10	6	18	11	15	12	8
C	9	9	0	8	9	7	15	11	14	14	9
D	9	7	8	0	8	7	14	9	14	13	0
E	7	10	9	8	0	9	16	11	13	13	8
F	7	6	7	7	9	0	14	11	12	12	8
G	13	18	15	14	16	14	0	16	12	11	13
H	10	11	11	9	11	11	16	0	15	16	9
I	14	15	14	14	13	12	12	15	0	9	12
J	12	12	14	13	13	12	11	16	9	0	11
K	9	8	9	0	8	8	13	9	12	11	0

By chance, the chain letters are grouped in such a way that the distance matrix suggest that G, H, I and J are "far away" from the rest of the letters.

Notice that D and K are apparently 0 units apart, and their rows are almost (but not quite) identical.

Given any table that is a distance map, or suggestive of one, we can try to make sense of the information by treating the objects as cities, and placing them on a map so that we approximate the distance table. It's usually impossible to do this perfectly, but we can always get an answer that minimizes the sum of squares of the errors.

For instance, suppose we have 4 objects and an estimated distance table:

```
     A    B    C    D
  +--------------
A | 00   AB   AC   AD
B | BA   00   BC   BD
C | CA   CB   00   CD
D | DA   DB   DC   00
```

We'll assume the matrix is symmetric, so there are six known distances. Assuming our map is two dimensional, we have 4 unknown X coordinates and 4 unknown Y coordinates.

This is an unusual case, since we have more unknowns than data. In the typical case of larger N, the number of unknowns, $2*N$, will be much smaller than the number of equations we are trying to satisfy, $nchoosekN, 2.$ constraints

Chain Letter Clustering Results: A Pseudomap

If we knew the locations, we could compute the distances:

$$AB^2 = (A_x - B_x)^2 + (A_y - B_y)^2$$
$$AC^2 = (A_x - C_x)^2 + (A_y - C_y)^2$$
$$AD^2 = (A_x - D_x)^2 + (A_y - D_y)^2$$
$$BC^2 = (B_x - C_x)^2 + (B_y - C_y)^2$$
$$BD^2 = (B_x - D_x)^2 + (B_y - D_y)^2$$
$$CD^2 = (C_x - D_x)^2 + (C_y - D_y)^2$$

We don't know the locations, but we want to choose their values so that they product this set of distances, or as close as possible. We think of the 8 variables $(A_x, A_y, B_x, ..., D_y)$ as our degrees of freedom and the 6 equations as a vector-valued function $f()$, and we are trying to solve

$$f(A_x, A_y, B_x, ..., D_y) = 0$$

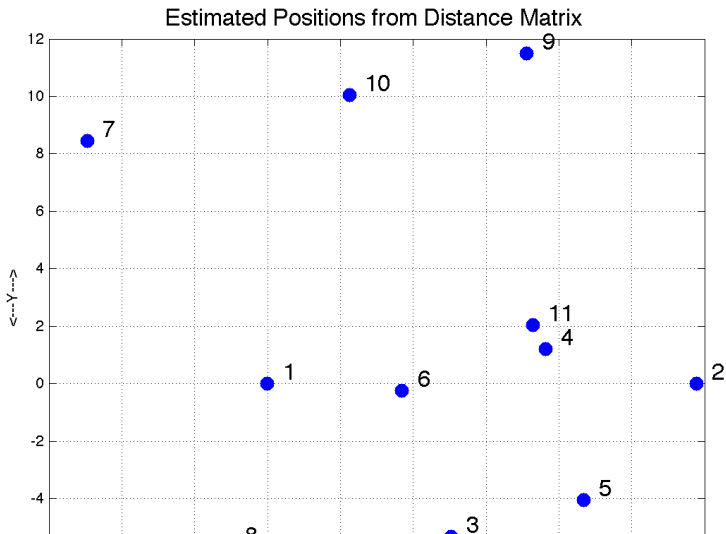or else to find the 8 variables that minimize the size of the error.

MATLAB's ??? function can minimize such a function...

Estimated Positions from Distance Matrix

The pseudomap certainly reminds us that chain letters 4 and 11 are very close, and it seems to put 7, 10, and 9 in a region further away from the others, but it's really not as enlightening as we might have hoped.
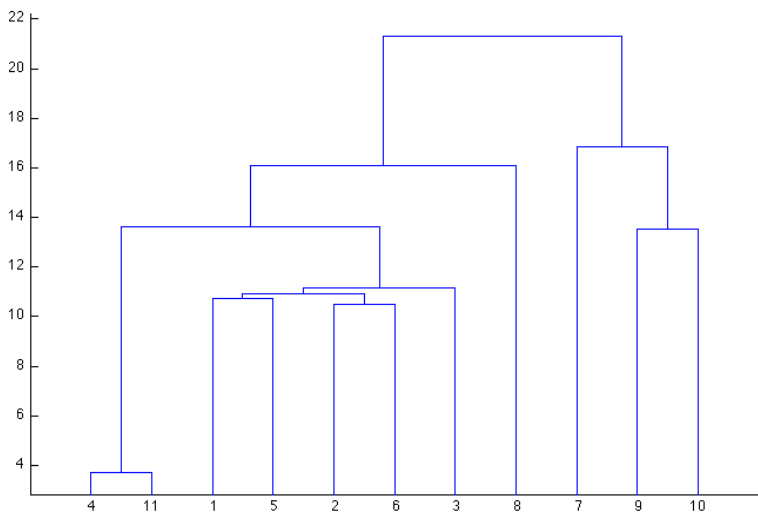
In fact, however, we actually have a feeling that the data we are observing represents an evolution, that is, growth from a single parent with divergence. In that case, there is another way of looking at the data that is much more revealing, a dendrogram, or, for those who don't speak Greek, literally a "tree diagram".

The following MATLAB commands will analyze our data:

```
dv = pdist ( dist );
dl = linkage ( dv, 'single' );
dendrogram ( dl )
```