The Application of Hierarchical Clustering to Homophonic Ciphers

Anna Lehofer
Department of Philosophy and History of Science
Budapest University of Technology and Economics
Budapest H-1111, Egry József u. 1. E 610, Hungary
lehofer.anna@gmail.com

Abstract

In this work in progress study I examined whether the method of hierarchical clustering could be used efficiently on Hungarian homophonic ciphers from the early modern age. First I have tested the methodology on artificial homophonic ciphers. The original corpora of these artificial codes were appropriate to ascertain the effectiveness of the method: knowing the plaintext I could control the outcome. In connection with text length I have identified the limits of the applicability of hierarchical clustering. In a second part, the investigation of eight original letters from the early modern age followed. The testing of original manuscripts shows whether the results based on the artificial ciphers are applicable to original historical documents as well.

1 Homophonic Ciphers of the Early Modern Age

In a homophonic substitution cipher single plaintext letters can be replaced with several code characters. In simpler cases only the vowels and the most frequent letters are replaced with more code characters, but in an advanced, complex cipher key, each of the plaintext letters receive several code characters, so-called homophones. I call these ciphers pure homophonic ciphers. But in many cases, early modern homophonic ciphers used separate tokens for syllables, logograms (characters representing frequent words or names) and nulls (meaningless tokens to confuse the cryptanalysis) beyond the homophones. I call these types of ciphers advanced homophonic systems.

Both pure homophonic ciphers and advanced homophonic systems were part of the early modern practice, even the simple monoalphabetic substitution was in use in some cases. Breaking these monoalphabetic codes can even be an easy task. The frequency analysis of the code characters, recurring character lines, vowel-consonant analysis can bring us closer to find the plaintext letters of the ciphers.

The same cannot be said about homophonic ciphers. Speaking of advanced homophonic systems of the 16th century, the few pages long character tables consisted of two or three homophones for each plaintext letter, about 10 symbols for nulls, 10 for bigraphs, 100-150 characters for syllables and even 300 characters for logograms (Láng, 2015, 37). For such codes, a properly composed and correctly used cipher-key can result in an almost even distribution in the frequency of the code characters, making the task of the codebreakers much harder. So the tools that can lead us to the decryption of monoalphabetic codes give us no help for decrypting homophonic systems.

In the practice, using homophonic substitution meant a higher security compared to simple monoalphabetic ciphering, but it also had its drawbacks. The complexity of the cipher-keys made the usage of this encrypting method slower and more complicated.

2 Hierarchical Clustering

"Cluster analysis groups data objects based only on information found in the data that describes the objects and their relationships. The goal is that the objects within a group be similar (or related) to one another and different from (or unrelated to) the objects in other groups. The greater the similarity (or homogeneity) within a group and the greater the

Proceedings of the 1st Conference on Historical Cryptology, pages 133– 136,
Uppsala, Sweden, 18-20 June, 2018

difference between groups, the better or more distinct the clustering (Kumar et al., 2005, 490)."

Speaking of homophonic ciphers, the base set of these data objects is the multitude of the code characters. The aim of the clustering process is to ascertain with which right and left neighbors the particular code characters appear in the text.

To illustrate the operation of hierarchical clustering to homophonic codes let's suppose that we have a homophonic cipher using 100 different code characters. The aim of the method is to investigate which code characters are likely to appear together. Based on the 100 code characters of the text, we prepare two 100x100 matrices. Both the rows and columns of the matrix represent the code characters of the cipher. If we point at a number in this matrix, it indicates the occurrence-frequency, how often the concerning two code characters (indicated by the row and the column) appear together. One matrix shows the occurrence frequency with the left neighbors, the other shows the occurrence frequency with the right neighbors. To create one attribution from the left and right neighborhood, we combine these two matrices and use the new 100x200 matrix in the following step. In this matrix, each line is a 200-dimensional vector, representing one code character. Depending on the neighbors of these code characters, each vector points to different directions. Similar vectors point almost to the same direction, vectors that differ from each other point into different directions.

From the upper vectors, on the basis of cosine distance function we generate a 100x100 distance matrix with values from 0 to 1. In the diagonal of this matrix (where the distance of the vectors from themselves appears, namely the distance of two equal vectors) the function gets a value of 1. The other values – depending on the angle locked together – will get values between 0 and 1. The more similar these vector pairs, the more they point to the same direction (the closer this value is to 1).

To display hierarchical clustering graphically I have used the open source Cran R software. It uses a tree-like diagram called a dendrogram to visualize these relationships. It draws these dendrograms on the basis of the distance-matrix.

Exactly the same method was efficiently used by the decryption process of the famous Copiale code (Kevin et al., 2011).

3 Artificial Ciphers

Hereupon I have created artificial homophonic codes from a Hungarian corpus (Géza Gárdonyi, Eclipse of the Crescent Moon) to be able to tell a bit more about the criteria for the optimal application. These artificial codes are pure homophonic codes which were created by Cran R that randomly assigned the desired number of homophones to the plaintext letters.

In the testing process I have investigated two things. First I have gradually increased the number of homophones assigned to a plaintext letter (starting with a monoalphabetic set of code characters) to see how long hierarchical clustering is able to detect the homophone groups. Secondly I have gradually reduced the length of the examined part of the text to find the point where hierarchical clustering loses its efficiency in finding the vowel and consonant groups and the groups of homophones.

3.1 Full Text Codes

Based on the artificial codes created from the full text of the novel I have faced with the followings. The first, monoalphabetic code has immediately brought in an interesting result. The software separated two bigger clusters on the dendrogram: one big cluster showed only vowels, the other bigger group contained only consonants. So the method can be used on monoalphabetic ciphers as well: it can almost perfectly separate vowels and consonants in a monoalphabetic ciphertext.

By tripling the number of the homophones clustering can also find the vowel and consonant groups, furthermore it can correctly recognize the three-element homophone groups belonging to the particular plaintext letters.

I was surprised when the program could even identify the vowel and consonant groups and the homophone groups when 20 homophones were assigned to a plaintext letter. It seems that in case of a 400-page corpus hierarchical clustering can identify the homophone groups belonging to the particular plaintext letters, even if we have far more homophones than the early modern practice shows (early modern cipher keys usually have 2-3 or 5-6 code characters for one plaintext letter at most).

Of course, in reality, codebreakers do not have book-lengthy texts. Most often they have a paragraph or at most a few pages written with encrypted characters. In the following, I have examined how the method worked when I started to reduce the length of the examined text.

3.2 Unicity Point

According to the writings of Elliot Fischer and James Reeds the limits of using hierarchical clustering efficiently will be discussed here with the concepts of text redundancy and unicity point. The unicity point of a cipher is U=H(k)/D where H(k) is

the logarithm of the number of possible keys of the ciphers and D is the redundancy of the language. The unicity point is the message length beyond which decipherment using a known system becomes a unique process. From the given formula it is clear that the lower the redundancy of a language, the greater the unicity point for a given cipher (Fischer, 1979 and Reeds, 1977).

I examined the original corpus in two ways. The first table shows how entropy – thus redundancy – and the unicity point changes when increasing the number of homophones gradually from 1 to 5 on the 700000-character-long corpus. Despite of the indicated infinite limit of the 5th case, all of the related five dendrograms have identified the vowel and consonant groups correctly and clustering could even find the 1-2-3-4-5 element homophone groups of the ciphers.

Number of homophones	Number of used code characters	H_{max}	H_{real}	Redundancy	Unicity point
1	35	5.129	4.58	0.107	1240
2	70	6.129	5.579	0.09	3706
3	105	6.714	6.164	0.082	6816
4	138	7.109	6.579	0.074	10569
5	174	7.443	6.901	0.073	∞

Table **1**: Increasing the number of homophones in the full text

The second table shows how unicity point changes when decreasing text length assuming 2 homophones for each plaintext letters. The first value (around 700000 characters) shows the full length of the text, 100%. Than follows 10%, 1%, 0.5% and finally 0.1%.

Text length (number of characters)	Number of used code characters	Unicity point
700934	70	3706
70093	66	3986
7009	66	4059
3504	65	4203
700	62	3515

Table **2**: How unicity point changes when reducing text length using 2 homophones per letter

We can see that in the given artificial code, speaking of pure homophonic substitution, using two homophones for each plaintext letter, the efficiency of hierarchical clustering falls down around the text length of 3500 characters. Here the unicity point is around 4200 characters, so a longer text is needed for a safe codebreaking than the examined one. The dendrograms of these cases also corroborate this statement: while the dendrogram of the 3500-character-long text can still separate a big cluster for vowels and another one for consonants almost perfectly, the dendrogram of the 700-character-long text (of which unicity point value is already much lower than the real text length) falls into smaller clusters. These small clusters may still support the individual codebreaking process but neither separate vowels and consonants, nor identify the homophone pairs of the ciphertext in a proper way.

4 Early Modern Letters

In this section, I will investigate encrypted letters[1] from the early modern age. The cipher keys of these letters were also available (in an archive or reconstructed form), thus the keys offered help and control when examining the efficiency of clustering.

The first letter I have examined – C.Bay.01 – was a 419-character-long almost fully encrypted letter that uses a very complex cipher key: beyond the homophonic set of code characters it also indicates syllables, logograms and nulls with separate signs. The dendrogram outlined as a result of clustering proved that this letter was too short, the cipher key was too complex to give any help in the decoding process.

After the Bay letter I looked for a letter with a less complex cipher key than the first one, and examined C.Wes.03.a. It was a 2359-character-long letter using an all-in-all 43-element cipher key, assigning more (5-6) code characters only to the vowels.

The cluster map of this cipher looked more promising. The software separated two bigger clusters: one showed only consonants, the other bigger cluster contained almost exclusively vowels. The program identified homophone pairs in five cases. The remaining smaller groups and the characters that were not grouped to other ones were mostly logograms, so they were "outranked" correctly from the homophones.

So far I have examined 6 more early modern ciphers to find out where the limits of applicability are. All of the scrutinized letters come from the period 1664-1706 and have their cipher keys in an available form as well.

To describe applicability, two outcomes were tested: 1) whether the clustering process could identify the vowels and consonants in different

1 Up to now I have investigated 8 early modern Hungarian letters. Since this is a work in progress, this outcome will be better grounded, after I will have transcribed and analyzed several other manuscripts in the near future.

clusters, and 2) whether the clustering process could identify the homophone groups belonging to the particular plaintext letters. In cases where clustering can show up any of these two identifications, hierarchical clustering can be stated effective. In these cases hierarchical clustering can support the codebreaking process.

The outcomes of the examined letters are summarized in the following table. The first column shows the name of the letters following the notation of Benedek Láng (Láng, 2015, 233). The column of *text length* shows how many code characters the concrete letters are made of; *number of used code characters* shows how many characters were actually used in the concrete letters. H_{max} shows the maximum value of entropy, H_{real} stands for the actual values of entropy. *Redundancy* shows the text redundancy of the letters, the column of *unicity point* indicates the required text length. *Vowel-consonant groups* shows whether the method of hierarchical clustering could separate the vowels and the consonants in different clusters; and homophone groups shows if the clustering process could identify the *homophone groups* belonging to the particular plaintext letters.

Letters[2]	Text length	Number of used code characters	H_{max}	H_{real}	Redundancy	Unicity point	Vowel-consonant groups	Homophone groups
C.Bay.01	419	113	6.82	6.113	0.104	5905	no	no
C.Bay.02	494	130	7.022	6.338	0.097	7490	no	no
C.Kov.02	1537	189	7.562	6.689	0.115	∞	no	no
C.Wess.03.a	2359	64	6	4.92	0.18	1643	vowels	in 6 cases
C.Wess.03.b	828	61	5.931	4.939	0.167	1662	vowels	in 5 cases
C.Wes.04	1525	77	6.267	4.994	0.203	1850	vowels	in 5 cases
C.Wes.05	749	26,	4.7	4.029	0.143	618	partly	-
C.Wes.06	417	37	5.209	4.231	0.188	763	no	no

Table **3**: Features of the examined early modern letters

5 Summary

In this paper I have first tested hierarchical clustering on artificial codes by modifying two parameters: increasing the number of homophones assigned to a plaintext letter and decreasing the text length. It can be stated that in case of a 400-page corpus hierarchical clustering could identify the homophone groups successfully, even if we had far more homophones (20) than the early modern practice showed (2-6). Investigating the unicity points of ciphertexts it can be stated that hierarchical clustering was still efficient when text length was under the unicity point, but near to it. In cases when text length was much lower than the unicity point, the dendrograms could not give any help for the codebreaking process.

In a second part I have processed original early modern ciphers with the upper methodology. I have stated that hierarchical clustering was efficient if it could clearly identify the vowels and consonants in separate clusters on the dendrogram and/or if it could find the homophone groups belonging to the particular plaintext letters. The features and outcomes of the eight early modern letters showed that when the unicity point was under or near the text length the dendrograms could help the codebreaking process. Hierarchical clustering could not bring any results in case of letters that were much shorter than the unicity point.

Consequently, speaking of homophonic substitution ciphers we can state that the longer an encrypted letter, or the less symbols its cipher key uses, the more probable the cipher can be solved with the help of hierarchical clustering. Since the historical manuscripts of the early modern age do involve such encrypted letters – we can find ciphers with thousands of code characters, or cipher keys that have only 30-40 symbols – hierarchical clustering offers significant contribution to the codebreaking process of historical homophonic substitution ciphers.

2 These letters can be found in the Hungarian National Arcives, G 15 Caps. D. Fasc 81. and G 15 Caps. C. Fasc 36. fol. 3-4. and in the ÖStA HHStA Ungarische Akten Specialia Verschwörerakten VII. Varia (Pressburger Kommission etc.) Fasc. 327. Konv. D. Chiffres 1664-1668, fol 35-37, 40-41, 62, 63.

References

Benedek Láng. 2015. *Titkosírás a Kora Újkori Magyarországon.* Balassi Kiadó, Budapest.

Elliot Fischer. 1979. Language Redundancy and Cryptanalysis. In *Cryptologia*, volume 3, pages 233-235.

James Reeds. 1977. Entropy Calculations and Particular Methods of Cryptanalysis. In *Cryptologia*, volume 1, pages 235-254.

Kevin Knight, Beáta Megyesi, Christiane Schaefer. 2011. The Copiale Cipher. Presented at the *ACL Workshop on Building and Using Comparable Corpora*.

Vipin Kumar, Michael Steinbach, Pang-Ning Tan. 2005. *Introduction to Data Mining*. Pearson (Education Inc.), Boston.