F-046

Biological Data Analysis based on Kolmogorov Complexity

Kimihito Ito[†] Thomas Zeugmann[‡] Yu Zhu [§]

Abstract. In this paper, we focus on one simple data mining method called Normalized Compression Distance (NCD) which has been suggested by Cilibrasi Vitányi. By this method, we analyzed the HA sequences of virus data for the HA gene based on the available compressors. The built-in compressors `zlib` and `bzip` are compared by using the Complearn Toolkit. And a comparison is made with respect to hierarchical and spectral clustering. Our results shows that one can obtain an (almost) perfect clustering. It turned out that the `zlib` compressor allowed for better results than the `bzip` compressor and, and the hierarchical clustering is a bit better than spectral clustering if all data are concerned.

1. Introduction

The similarity between objects is a fundamental notion in everyday life. Usually the similarity between objects is measured by a domain-specific distance measure based on features of the objects. For example, the distance between pieces of music can be measured by using features like rhythm, pitch, or melody, i.e., features that do not make sense in any other domain. If one is pursuing the approach to design data mining algorithms based on domain knowledge, then the resulting algorithms tend to have many parameters. Expressing the differently, one has to tune the algorithms which is requiring domain knowledge and a larger amount of experience. Furthermore, it may be expensive, error prune and time consuming to arrive at a suitable tuning.

So the approach of parameter-free data mining is aiming at scenarios where we are not interested in a certain similarity measure but in the similarity between the objects themselves. The most promising approach to this paradigm uses Kolmogorov complexity theory [10] as its basis. The key ingredient to this approach is the so-called *normalized information distance* (NID) which was developed by various researchers during the past decade in a series of steps (cf., e.g., [3, 9, 6]).

More formally the *normalized information distance* between two strings $\mathtt{x}$ and $\mathtt{y}$ is defined as

$$NID(\mathtt{x},\mathtt{y}) = \frac{\max\{\mathsf{K}(\mathtt{x}|\mathtt{y}),\, \mathsf{K}(\mathtt{y}|\mathtt{x})\}}{\max\{\mathsf{K}(\mathtt{x}),\, \mathsf{K}(\mathtt{y})\}}\ , \qquad (1)$$

where $\mathsf{K}(\mathtt{x}|\mathtt{y})$ is the length of the shortest program that outputs $\mathtt{x}$ on input $\mathtt{y}$, and $\mathsf{K}(\mathtt{x})$ is the length of the shortest program that outputs $\mathtt{x}$ on the empty input. It is beyond the scope of the present paper to discuss the technical details of the definition of the NID. We refer the reader to Vitányi *et al.* [11].

Since its definition involves the Kolmogorov complexity $\mathsf{K}(\,\cdot\,)$, the NID cannot be computed. Therefore, to apply this idea to real-world data mining tasks, standard compression algorithms, such as `gzip`, `bzip2`, or `PPMZ`, have been used as approximations of the Kolmogorov complexity. This yields the *normalized compression distance* (NCD) as approximation of the NID (cf. Definition 1).

The main goal of the present paper is a detailed analysis of the general method outlined above in the domain of influenza viruses. More specifically, we are interested in learning whether or not specific gene data for the hemagglutinin of influenza viruses are *correctly* classifiable by using the concept of the NCD. For this purpose we have chosen a set of 106 gene sequences from the National Center for Biotechnology Information for which the correct classification of the hemagglutinin is known. As explained in Section 3., there are 16 subtypes commonly called H1, ..., H16. For these 106 gene sequences (or subsets thereof) we then compute the NCD by using the CompLearn Toolkit (cf. [4]) as done in [5].

This computation returns a symmetric matrix D such that $\mathtt{d}_{\mathtt{ij}}$ is the NCD between the data entries $\mathtt{i}$ and $\mathtt{j}$ (henceforth called distance matrix). The next step is the clustering. As the first clustering algorithm, we tried the *hierarchical clustering* algorithm from the R package (called `hclust`) with the average option. The second clustering algorithm used is *spectral clustering* via kLines (cf. Fischer and Poland [7]).

The results obtained are generally very promising. Quite often, we obtained a *perfect* clustering independently of the method used. On the other hand, when including all data or a rather large subset thereof, the clustering obtained is not perfect but the number of errors made is still sufficiently small to make the results interesting. Without going into details here, it can be

[†]Research Center for Zoonosis Control, Hokkaido University.
[‡]Division of Computer Science,Graduate School of Information Science and Technology, Hokkaido University.
[§]Division of Computer Science,Graduate School of Information Science and Technology, Hokkaido University.

said that the zlib compressor seems more suitable in this setting than the bzip2 compressor.

2. Background and Theory

As explained in the Introduction,The definition of the NID depends on the function K which is *uncomputable*. Thus, the NID is *uncomputable*, too. Using a real-word compressor, one can approximate the NID by the NCD (cf. Definition 1). Again, we omit details and refer the reader to [11].

Definition 1. *The* normalized compression distance *between two strings* x *and* y *is defined as*

$$NCD(x,y) = \frac{\{C(xy) - \min\{C(x),\ C(y)\}}{\max\{C(x),\ C(y)\}} \ ,$$

where C *is any given data compressor.*

Common data compressors are gzip, bzip2, zlib, etc. Note that the compressor C has to be computable and *normal* in order to make the NCD a useful approximation. This can be stated as follows.

Definition 2 ([11]). *A compressor* C *is said to be* normal *if it satisfies the following axioms for all strings* x, y, z *and the empty string* λ.

(1) $C(xx) = C(x)$ *and* $C(\lambda) = 0$; (identity)

(2) $C(xy) \geqslant C(x)$; (monotonicity)

(3) $C(xy) = C(yx)$; (symmetry)

(4) $C(xy) + C(z) \leqslant C(xz) + C(yz)$; (distributivity)

up to an additive $O(\log n)$ *term, with* n *the maximal binary length of a string involved in the (in)equality concerned.*

These axioms are in various degrees satisfied by good real-world compressors like bzip2, PPMZ and gzip, where the latter did not perform so well, as informal experiments have shown (cf. [6]). For our investigations we used the built-in compressors bzip2 and zlib and the ncd function from the CompLearn Toolkit (cf. [4]). After having done this step, we have a distance matrix $D = \big(d(x,y)\big)_{x,y\in X}$, where $X = (x_1, \ldots, x_n)$ is the relevant data list.

Next, we turn our attention to clustering. First, we shortly outline the hierarchical clustering as provided by the R package, Input is the $(n \times n)$ distance matrix D. The program uses a measure of dissimilarity for the objects to be clustered. Initially, each object is assigned to its own cluster and the program proceeds iteratively. In each iteration the two most similar clusters are joint, and the process is repeated until only a single cluster is left. Furthermore, in every iteration the distances between clusters are recomputed by using the Lance–Williams dissimilarity update formula for the particular method used.

The methods differ in the way in which the distances between clusters are recomputed. Provided are the *complete linkage method*, the *single linkage method*, and the *average linkage clustering*. The *average linkage clustering* defines the distance between any two clusters to be the average of distances between all pairs of objects from any member of one cluster to any member of the other cluster. As a result, the average pairwise distance within the newly formed cluster, is minimum.

Next, the spectral *spectral clustering* algorithm used is shortly explained. The transformation of the distance matrix into a similarity matrix is done by using a suitable kernel function. In our experiments we have used the Gaussian kernel function. So, the remaining problem is a suitable choice for σ. Unfortunately, the performance of spectral clustering heavily depends on this σ. In the experiments, we compute the mean value of the entries of the distance matrix D and then set $\sigma = \mathtt{mean}(D)/\sqrt{2}$. The final spectral clustering algorithm for a known number of clusters k is stated below.

Algorithm Spectral clustering of a data list
Input: data list $X = (x_1, x_2, \ldots, x_n)$, number of clusters k
Output: clustering $c \in \{1 \ldots k\}^n$
1. for $x, y \in X$, compute the distance matrix $D = \big(d(x,y)\big)_{x,y\in X}$
2. compute $\sigma = \mathtt{mean}(D)/\sqrt{2}$
3. compute the similarity matrix $A = \big(\exp(-\frac{1}{2}d(x,y)^2/(2\cdot\sigma^2))\big)$
4. compute the Laplacian $L = S^{-\frac{1}{2}}AS^{-\frac{1}{2}}$, where $S_{ii} = \sum_j A_{ij}$ and $S_{ij} = 0$ for $i \neq j$
5. compute top k eigenvectors $V \in^{n\times k}$
6. cluster V using kLines

3. Experiments and Results

In this section we describe the data used, the experiments performed and the results obtained.

3.1 Influenza Viruses - The Data Set

We shortly describe the data set used. Influenza viruses were probably a major cause of morbidity and mortality world wide. Biologists classify influenza A viruses primarily by their hemagglutinin type. Hemagglutinin (HA) is an antigenic glycoprotein found on the surface of the virus. This protein is responsible for binding the virus to the cell it infects. So far, 16 subtypes of HA are known and commonly denoted by H1, ..., H16. Influenza A viruses of all 16 hemagglutinin (H1-H16) subtypes are maintained in their nature host, i.e., the duck. HA is a the major target of an-

H1	H2	H3	H4	H5	H6	H7	H8
8	8	8	8	8	8	8	7
H9	H10	H11	H12	H13	H14	H15	H16
8	8	8	8	2	4	4	1

Figure 1: Number of sequences for each subtype

tibodies that neutralize viral infectivity. Therefore, in the experiments performed we have exclusively selected data of influenza viruses that have been obtained from viruses hosted by the duck.

In order to cluster the sequences we collected from each subtype up to 8 examples. The reason for choosing at most 8 sequences from each type has been caused by their availability. While for some subtypes there are many sequences, there are also subtypes for which only very few sequences are available. The extreme case is the subtype H16 for which only one sequence is in the data base. Table 1 shows the number of sequences chosen.

For a complete list of the data description we refer the reader to

http://www-alg.ist.hokudai.ac.jp/datasets.html .

For the ease of presentation, in the following we use the following abbreviation for the data entries. Instead of giving the full description, e.g.,
>gi|113531192|gb|AB271117|/Avian/4(HA)/H10N1/ Hong Kong/1980/// Influenza A virus(A/duck/Hong Kong/938/80(H10N1)) HA gene for hemagglutinin, complete cds.

we refer to this datum as H10N1AB271117 for short.

3.2 Results

All experiments have been performed under SuSE Linux. As already mentioned, for the hierarchical clustering we used the open source R package (cf. [2]).

The Algorithm (Spectral clustering of a data list) has been realized by performing Step 1 via the CompLearn function `ncd` (cf. [4]). Steps 2 through 6 have been implemented in `GNU Octave`, version 2.1.72 (cf. [1]). It should be noted that `ncd` assigns 0.000000 to all elements on the main diagonal of the distance matrix (Version 1.1.5).

Our results show that one can obtain an (almost) perfect clustering. Details can be found in [8].

4. Conclusions

The usefulness of the normalized compression distance for clustering the HA type of virus data for the HA gene for it (segment 4) has been demonstrated. Though we just used the built-in compressors `zlib` and `bzip` the results are (almost) correct when clustering the resulting distance matrix for the whole data set with `hclust` or spectral clustering via kLines.

References

[1] Gnu octave. http://www.gnu.org/software/octave/.

[2] The R project for statistical computing. http://www.r-project.org/.

[3] C. H. Bennett, P. Gács, M. Li, P. M. B. Vitányi, and W. H. Zurek. Information distance. *IEEE Transactions on Information Theory*, 44(4):1407–1423, 1998.

[4] R. Cilibrasi. The CompLearn Toolkit, 2003-. http://www.complearn.org/.

[5] R. Cilibrasi and P. M. Vitányi. A new quartet tree heuristic for hierarchical clustering. In D. V. Arnold, T. Jansen, M. D. Vose, and J. E. Rowe, editors, *Theory of Evolutionary Algorithms*, number 06061 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2006.

[6] R. Cilibrasi and P. M. B. Vitányi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.

[7] I. Fischer and J. Poland. New methods for spectral clustering. Technical Report IDSIA-12-04, IDSIA / USI-SUPSI, Manno, Switzerland, 2004.

[8] K. Ito, T. Zegumann, and Y. Zhu. Clustering the normalized compression distance for virus data. In *proceedings of Sixth Workshop on Learning with Logics and Logics for Learning*, 2009.

[9] M. Li, X. Chen, X. Li, B. Ma, and P. M. Vitányi. The similarity metric. *IEEE Transactions on Information Theory*, 50(12):3250–3264, 2004.

[10] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer, 3rd edition, 2008.

[11] P. M. B. Vitányi, F. J. Balbach, R. L. Cilibrasi, and M. Li. Normalized information distance. In *Information Theory and Statistical Learning*, pages 45–82. Springer, New York, 2008.