Image Analysis in a Parameter-Free Setting

Yu Zhu and Thomas Zeugmann

Division of Computer Science, Hokkaido University, N-14, W-9, Sapporo 060-0814, Japan
{thomas,zhuyu07}@ist.hokudai.ac.jp

Summary. The paper proposes a new method to approximate the normalized information distance by a compression method that is particularly suited for image data. The new method is based on a video compressor. The new method is used to compute the distance matrix of all the images in the data sets considered. Moreover, the hierarchical clustering method from the R package is used to cluster the distance matrix obtained. Two different datasets are considered to demonstrate the usefulness of our new image analysis method. The results are very promising and show that one can obtain a very good clustering of the image data.

1 Introduction

To measure the similarity between objects is a fundamental notion in everyday life. For many data mining and machine learning algorithms the task of measuring the similarity between objects is also fundamental. Usually, the similarity between objects is measured by a domain-specific 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., these features do not make sense in any other domain. For defining the right domain-specific distance measure one needs special knowledge about the application domain for extracting the relevant features beforehand. By using these parameters, one can then control the algorithms' sensitivity to certain features. Determining how relevant particular features are is often difficult and may require a certain amount of guessing. Expressing this differently, one has to tune the algorithms, which is requiring domain knowledge and a larger amount of experience. Such an approach does not only cause difficulties, but includes a certain danger or risk of being biased. Furthermore, it may be expensive, error prune, and time consuming to arrive at a suitable tuning.

However, as a radically different approach; i.e., the paradigm of *parameter-free data mining*, has emerged (cf. Keogh *et al.* [4]). The main idea of parameter-free data mining is the design of algorithms that have no parameters and that are universally applicable in all areas.

The problem is whether or not such an approach can be realized at all. It is only natural to ask how an algorithm can perform well if it is not based on extracting the important features of the data, and if we are not allowed to adjust its parameters until it is doing the right thing. As expressed by Vitányi *et al.* [12], *if we a priori know the features, how to extract them, and how to combine them into exactly the distance measure we want, we should do just that. For example, if we have a list of cars with their color, motor rating, etc. and want to cluster them by color, we can easily do that in a straightforward way*.

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 is based on Kolmogorov complexity theory. The intuitive idea can be described as follows: If two objects x and y are similar then it should be possible to obtain a short description of how to transform object x into y and object y into object x. Conversely, if two objects have (almost) nothing in common, then obtaining x from y and y from x is (almost) as complex as describing x and y, respectively, from scratch. Note that we need both directions here. For example, if we are given a blue image and a beautiful flower image (of the same size) the one can easily obtain the blue image from the flower image by assigning to each pixel the color blue. But the converse is not true; i.e., obtaining the flower from the blue image is as complex as describing the flower from the empty image.

The key ingredient to this approach is the so-called *normalized information distance* (NID) which was developed by various researchers during the past decades in a series of steps (cf., e.g., [1, 5, 2]).

More formally the *normalized information distance* between two strings x and y is defined as

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

where $K(x|y)$ is the length of the shortest program that outputs x on input y, and $K(x)$ is the length of the shortest program that outputs 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.* [12].

The NID has nice theoretical properties, the most important of which is universality. The NID is called *universal*, since it accounts for the dominant difference between two objects (cf. Li *et al.* [5] and Vitányi *et al.* [12] and the references therein).

In a sense, the NID captures all computational ways in which the features needed in the traditional approach could be defined. Since its definition involves the Kolmogorov complexity $K(\cdot)$, 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 to approximate the Kolmogorov complexity and resulted in the *normalized compression distance* (NCD). We modify this approach by using a video compressor. This yields the *normalized MPEG distance* (NMD) as an approximation of the NID (cf. Definition 1). In [3] we have shown that the NCD based method is useful for text strings, such as influenza virus datasets.

But the NCD method could not find the real similarity in image datasets, even if we get a good compression ratio for images by the standard compression algorithms.

In this paper, we focus our attention to image datasets and provide the NMD to calculate the distance between images. We show our method to create an efficient and robust image distance measure. Using the NMD we compute a symmetric matrix D such that d_{ij} is the NMD between the data entries i and j (henceforth called distance matrix). The next step is the clustering. Here of course the variety of possible algorithms is large. We have decided to try the *hierarchical clustering* algorithm from the R package (called `hclust`) with the average option. In this way we obtain a rooted tree showing the relations among the input data. The results obtained are generally very promising. For the butterflies dataset, we obtained a perfect clustering result, which nicely coincides with a clustering obtained by human intuition. For the spider dataset, we still could get accuracy of 83.59% without any occasional human intervention.

2 Background and Theory

2.1 Background

As explained in the Introduction, the theoretical basis for computing the distance matrix is deeply based in Kolmogorov complexity theory. Since the definition of the NID depends on the function K and since K is *uncomputable*, the NID is *uncomputable*, too. Thus it must be approximated. Experience showed that universal compression algorithms yield good approximations of the NID.

In our previous work we also demonstrated that these compressors gave unexpected good results for text based datasets such as virus datasets (cf. [3]).

However for the images, these ”universal” compressors could not find the real similarity between images but only reduced the size of the data, even if they are lossless compressors. The main reason is that image data are usually large and thus the standard compressors are *not normal* when applied to image data. For example, if C is compressor then it should satisfy the condition $C(xx) = C(x)$, it should be symmetric and distributive and obey a monotonicity condition (cf. [3] for the formal definition).

The usual way to deal with images is to extract their features and then to compare the features extracted. Various methods have been proposed to perform this feature extraction.

Here we aim at an approach that maintains the benefits of parameter-free data mining. The idea is to use a video compressor to approximate the NID in an appropriate manner. It is known that widely used video compressors work as a tool to reduce the amount of data needed for subsequent frames. This means if the two images are similar, then the compressor can reduce more data and returns a file of smaller size. If the two images have almost nothing in common, then the video compressor basically just concatenates the two images and returns a file of bigger size.

However, using a video compressor poses some technical difficulties, since it requires the input of at least two images. Therefore, the original definition of the normalized compression distance has to be modified appropriately. This is done in the following subsection.

2.2 Our new modified measure method

Looking at the definition of the NID, we see that it takes two inputs and then determines the length of the shortest program to produce x and y (expressed by $K(x)$ and $K(y)$, respectively). Furthermore, determining $K(x|y)$ means to figure out how much information about x is already contained in y and similarly for $K(y|x)$.

Since the video compressor m requires at least two images, this poses the problem how to relate $m(xy)$ and $m(yx)$ which maybe considered as approximations of $K(y|x)$ and $K(x|y)$, respectively, to $m(xx)$ and $m(yy)$. The result is provided in Definition 1 below; i.e., we compare the resulting compressions $m(xy)-m(xx)$ and $m(yx)-m(yy)$ and normalize it by dividing by the maximum of $m(xx)$ and $m(yy)$.

We choose to realize this idea by using the *MPEG* encoder provided by MathWorks in Matlab for its simplicity and availability [9]. Of course, all of the input images should be in the same format, it could be *jpg*, *jpeg* or *png* any uniform files. The first step will transform the two individual images to two frames, and then transform them to one movie. The only thing we care about here is the size of the movie, as this indicate how similar the two images are. We perform a pairwise transformation for all of the images in the dataset. Then we can calculate the distance between each pair of images as defined in our measure method. More precisely, we have the following.

Definition 1. *The distance between two images x and y is defined as*

$$MD(x,y) = \frac{\max\{m(xy)-m(xx),\, m(yx)-m(yy)\}}{\max\{m(xx),\, m(yy)\}}$$

where m is the given video compressor.

Having this definition we can put all images in a list $X(x_1,\ldots,x_n)$ and compute the distance matrix $MD = \big(md(x,y)\big)_{x,y\in X}$. Here, the MD stands for "*MPEG* Distance." Note that the distance matrix $MD(x,y)$ returned is positive and symmetric.

Next, we turn our attention to clustering. We shortly outline the hierarchical clustering as provided by the R package, i.e., by the program `hclust` (cf. [10]). The input is the $(n\times n)$ distance matrix MD. 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*. In the first case, the distance between any two clusters is equal to the greatest similarity from any member of one cluster to any member of the other cluster. This method works well for compact clusters but causes sensitivity to outliers. The second method pays attention solely to the area where the two clusters come closest to one another. The more distant parts of the clusters and the overall structure of the clusters is not taken into account. If the total number of clusters is large, a messy clustering may result.

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.

Heuristically, the average linkage clustering should give the best results in our setting, and thus we have chosen it (see also Manning *et al.* [6] for a thorough exposition).

Hierarchical clustering builds clusters within clusters, and does not require a prespecified number of clusters like k-means and k-medoids do. A hierarchical clustering can be thought of as a tree and displayed as a dendrogram; at the top there is just one cluster consisting of all the observations, and at the bottom each observation is an entire cluster. In between are varying levels of clustering.

As shown below, our algorithm works in this way.

```
Algorithm MPEG distance clustering for a data list

Input: image data list $X = (x_1, x_2, \dots, x_n)$;
Output: clustering tree displayed as a dendrogram;
Step 1. for $x, y \in X$, use $MPEG$ compressor to get the video size $m(xy)$ for pairwise
    images;
Step 2. compute the distance matrix $MD = \big(md(x,y)\big)_{x,y \in X}$;
Step 3. cluster the matrix $MD$ by using hierarchical clustering from the R package
    (cf. [10]).
```

3 Experiments and Results

In this section we describe the two datasets used and the results obtained. The clusterings obtained provide evidence for the usefulness of our new method for image classification.

3.1 Butterflies dataset

First, we applied our new method to a set of butterfly images in order to judge its effectiveness intuitively. These images were extracted from various websites devoted to entomology. The result obtained is displayed in Figure 1. As we see the butterflies belong to six clusters based on their color and shapes.

In order to obtain this clustering, first we performed an alignment of the images; i.e., we ensured that all images have the same size. Here we kept the resolution ratio as its original ratio and added a frame to ensure that all images have the same size. This preprocessing is necessary, since in a movie all images are assumed to be of the

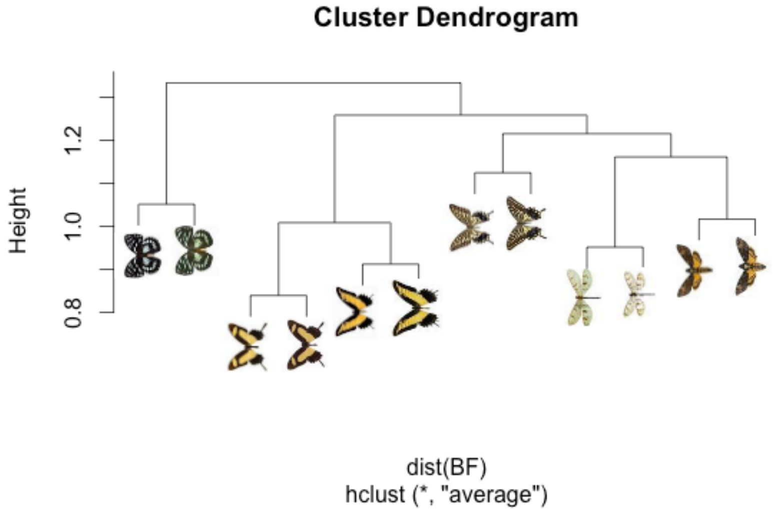

Fig. 1. The butterflies dataset clustered with our new distance measure method (hierarchical clustering method as provided by R package).

same size. If an image did not fit in the frame, a white background was added to fill the blank part.

After this step, we calculated the symmetrical pairwise distance matrix. Then we applied the hierarchical clustering as described above to judge the *similarity* between these images. As Figure 1 shows, the obtained result is really satisfying.

Note that we clustered this set of images, because it was used to test the utility of a color and shape distance measures in [13]. We found that the parameter-free method's result is even better than the one obtained by the carefully tuned color or shape measure.

3.2 Spider dataset

To show that the MD measure method could be widely used, we chose the Australasia ground spiders of the family Trochanteriidae as our dataset. This selection was made primarily because all species were already available and the size of the family is 121 species in 14 genera, this seems a reasonable dataset and we can use it to judge the effectiveness of our method [11].

Species discrimination in spiders is based primarily on the shape of the male and female genitalia. If we try to identify the species, or to systematically describing new species, we need to examine these structures. Epigyna is the reproductive structures for female spider. The epigynum is found on the ventral side of an adult female and is visible without dissection. (cf. Figure 2 and cf. Figure 3)

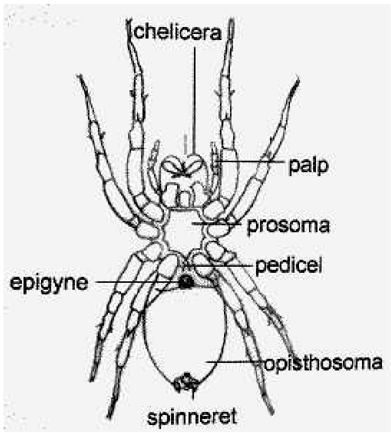

Fig. 2. spider ventral view

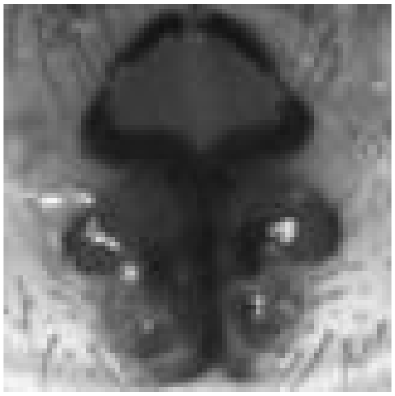

Fig. 3. One example of Epigynum for spider

In order to allow the reader to check the results obtained, we shortly describe the data set used. We chose 128 epigyna images from four species; i.e., the first two belong to the Boolathana species (displayed in yellow), the third to the 35th belong to the Fissarena species (shown in green), the 36th to 92nd are Hewicloeina species (displayed in red) and the remaining ones (93rd to 128th) are Longrita species (drawn in blue) (cf. Figure 4).

All experiments were performed under Mac OS X Lion. We calculated the distance matrix by using MPEG in Matlab. From the 128 spider images 107 are correctly clustered. Thus, the accuracy of the cluster for the spider data is 83.59%.

By performing our experiments we demonstrated the usefulness of our new measure method for image data. We could easily justify the clustering result for butterflies even we do not have background knowledge in this area. For the spider dataset, results of informal surveys of archaeologists suggest that acceptable cutoffs for accuracy vary widely and often depend on the background of the respondents. Systematists or taxonomic specialists demand on the highest accuracy level – 95 percent minimum for such a system to be useful for them. Ecologist and conservationists would be happy with 80-90 percent if it meant they could have a species list to work with [8]. Our method gave an acceptable result without any trained personal who are able to identify known species correctly. Especially for the Fissarena species (in green) and Longrita species (in blue) we only got 7 individual mixed in the clustering result, this part gave an 91.67% accuracy result. We have not reported the speed of the *MPEG* distance method, one of the reasons is we wanted to show the utility first. Another reason is that optimizing speed may be irrelevant in many domains. As in some medical application it may take over an hour to produce an image, and wait another hour to find matches in a database [7].

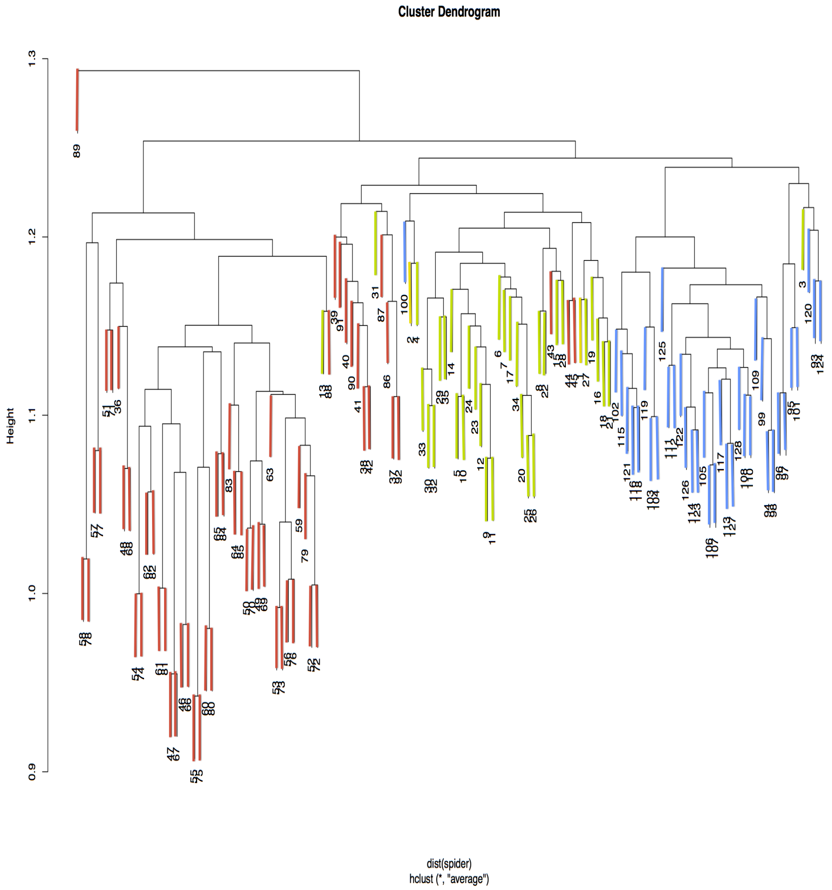

Fig. 4. The spider dataset clustered with our new distance measure method (hierarchical clustering method as provided by R package).

4 Conclusion and Future Work

The usefulness of our new measure method, *MPEG* distance, for clustering the butterflies images and spider's epigyna images has been demonstrated. For the butterflies dataset, we got a perfect clustering result. And for the spider subset, we got an accuracy of 83.59%. We are not claiming the *MPEG* distance is the best measure possible for image analysis problems. We have not reported the running time here, which it is still acceptable, since the process of identification and description of new species usually takes months or even years. For specialized application areas, there may be better measures, which include domain specific constraints and features. However, the *MPEG* distance measure offers a useful simple way when we do not have so much background about the application domain.

For the future work, we also would like to combine our measure method with the multisets to find the similarity between a pair of finite objects based on compression. Exploiting some other video compressors maybe increase the speed of our method. Of course, to apply the *MPEG* distance to other fields is the first task we will pursue next.

5 Acknowledgments

We would like to thank to the program committee and the anonymous referees for their valuable comments.

References

1. Charles H. Bennett, Péter Gács, Ming Li, Paul M. B. Vitányi, and Wojciech H. Zurek. Information distance. *IEEE Transactions on Information Theory*, 44(4):1407–1423, 1998.
2. Rudi Cilibrasi and Paul M. B. Vitányi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.
3. Kimihito Ito, Thomas Zeugmann, and Yu Zhu. Clustering the normalized compression distance for influenza virus data. In *Algorithms and Applications*, volume 6060 of *Lecture Notes in Computer Science*, pages 130–146. Springer, 2010.
4. Eamonn Keogh, Stefano Lonardi, and Chotirat Ann Ratanamahatana. Towards parameter-free data mining. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 206–215. ACM Press, 2004.
5. Ming Li, Xin Chen, Xin Li, Bin Ma, and Paul M.B. Vitányi. The similarity metric. *IEEE Transactions on Information Theory*, 50(12):3250–3264, 2004.
6. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
7. Theo Pavlidis. Limitations of content-based image retrieval, 2008. unpublished manuscript: http://www.theopavlidis.com/technology/CBIR/PaperB/vers3.htm.
8. Kimberly N. Russell, Martin T. Do, Jeremy C. Huff, and Norman I. Platnick. Introducing spida-web: Wavelets, neural networks and internet accessibility in an image-based automated identification system. In Norman MacLeod, editor, *Automated Taxon Identification in Systematics: Theory, Approaches and Applications*, pages 131–152. CRC Press, 2007.
9. S. Sumathi and Surekha Paneerselvam. *Computational Intelligence Paradigms Theory and Applications using MATLAB*. CRC Press, 2010.
10. The R project for statistical computing. http://www.r-project.org/.
11. Jaime R. Ticay-Rivas, Marcos del Pozo-Baños, William G. Eberhard, Jesús B. Alonso, and Carlos M. Travieso. Spider specie identification and verification based on pattern recognition of it cobweb. *Expert Systems with Applications*, 40(10):4213–4225, 2013.
12. Paul M. B. Vitányi, Frank J. Balbach, Rudi L. Cilibrasi, and Ming Li. Normalized information distance. In *Information Theory and Statistical Learning*, pages 45–82. Springer, New York, 2008.
13. Xiaoyue Wang, Lexiang Ye, Eamonn Keogh, and Christian Shelton. Annotating historical archives of images. In *Joint Conference on Digital Libraries*, pages 341–350, 2008.