# Compression-Based Data Mining

**Eamonn Keogh**
*University of California - Riverside, USA*

**Li Wei**
*Google, Inc., USA*

**John C. Handley**
*Xerox Innovation Group, USA*

## INTRODUCTION

Compression-based data mining is a universal approach to clustering, classification, dimensionality reduction, and anomaly detection. It is motivated by results in bioinformatics, learning, and computational theory that are not well known outside those communities. It is based on an easily computed compression dissimilarity measure (CDM) between objects obtained by compression. The basic concept is easy to understand, but its foundations are rigorously formalized in information theory. The similarity between any two objects (XML files, time series, text strings, molecules, etc.) can be obtained using a universal lossless compressor. The compression dissimilarity measure is the size of the compressed concatenation of the two objects divided by the sum of the compressed sizes of each of the objects. The intuition is that if two objects are similar, lossless compressor will remove the redundancy between them and the resulting size of the concatenated object should be close the size of the larger of the two compressed constituent objects. The larger the CDM between two objects, the more dissimilar they are.

Classification, clustering and anomaly detection algorithms can then use this dissimilarity measure in a wide variety of applications. Many of these are described in (Keogh et al., 2004), (Keogh et al. 2007), and references therein. This approach works well when (1) objects are large and it is computationally expensive to compute other distances (e.g., very long strings); or (2) there are no natural distances between the objects or none that are reasonable from first principles. CDM is "parameter-free" and thus avoids over-fitting the data or relying upon assumptions that may be incorrect (Keogh et al., 2004).

CDM enjoys the following properties:

1. Because it makes no distributional or modeling assumptions about the data, it allows true exploratory data mining.
2. The accuracy of CDM is often greatly superior to those of parameter-laden or model-based algorithms, even if we allow these algorithms to search exhaustively over their parameter spaces.
3. CDM uses compression algorithms which are typically space and time efficient. As a consequence, CDM can be much more efficient than other algorithms, in some cases by three or four orders of magnitude.
4. CDM makes no assumption about the format of the data, nor does it require extensive data cleaning to be effective.

## BACKGROUND

The use of data compression to classify sequences is also closely related to the Minimum Description Length (MDL) and Minimum Message Length (MML) principles (Grünwald, 2007), (Wallace, 2005). See keyword definitions at the end of the article. TheMDL/MMLprinciple has generated an extensive body of literature in the data mining community. CDM is a related concept, but it requires no probabilistic concepts and can be universally applied.

CDM is based on the concept of Kolmogorov complexity, a measure of randomness of strings based on their information content (Li & Vitanyi, 1997). It was proposed by Kolmogorov in 1965 to quantify the randomness of strings and other objects in an objective and absolute manner. The Kolmogorov complexity K(x) of a string x is defined as the length of the shortest program capable of producing x on a universal computer — such as a Turing machine. Different programming languages will give rise to distinct values of K(x), but one can prove that the differences are only up to a fixed additive constant. Intuitively, K(x) is the minimal quantity of information required to generate x by an algorithm. The conditional Kolmogorov complexity K(x|y) of x to y is defined as the length of the shortest program that computes x when y is given as an auxiliary input to the program. The function K(xy) is the length of the shortest program that outputs y concatenated to x. In Li et al. (2001), the authors consider the distance between two strings, x and y, defined as

$$d_k(x, y) = \frac{K(x \mid y) + K(y \mid x)}{K(xy)} \tag{1}$$

which satisfies the triangle inequality, up to a small error term. A more mathematically precise distance was proposed in Li et al. (2003). Kolmogorov complexity is without a doubt the ultimate lower bound among all measures of information content. Unfortunately, it cannot be computed in the general case (Li and Vitanyi, 1997). As a consequence, one must approximate this distance. It is easy to realize that universal compression algorithms give an upper bound to the Kolmogorov complexity. In fact, K(x) is the best compression that one could possibly achieve for the text string x. Given a data compression algorithm, we define C(x) as the size of the compressed size of x, C(xy) as the size of the compressed size of the concatenation of x and y and C(x|y) as the compression achieved by first training the compression on y, and then compressing x. For example, if the compressor is based on a textual substitution method, one could build the dictionary on y, and then use that dictionary to compress x. We can approximate (1) by the following distance measure

$$d_c(x, y) = \frac{C(x \mid y) + C(y \mid x)}{C(xy)} \tag{2}$$

The better the compression algorithm, the better the approximation of $d_c$ for $d_k$ is. Li et al., (2003) have shown that $d_c$ is a similarity metric and can be successfully applied to clustering DNA and text. However, the measure would require hacking the chosen compression algorithm in order to obtain C(x|y) and C(y|x). CDM simplifies this distance even further. In the next section, we will show that a simpler measure can be just as effective.

A comparative analysis of several compression-based distances has been recently carried out in Sculley and Brodley (2006). The idea of using data compression to classify sequences over finite alphabets is not new. For example, in the early days of computational biology, lossless compression was routinely used to classify and analyze DNA sequences. Refer to, e.g., Allison et al. (2000), Baronchelli et al. (2005), Farach et al. (1995), Frank et al. (2000), Gatlin (1972), Kennel (2004), Loewenstern and Yianilos (1999), Needham and Dowe (2001), Segen (1990), Teahan et al. (2000), Ferragina et al. (2007), Melville et al. (2007) and references therein for a sampler of the rich literature existing on this subject. More recently, Benedetto et al. (2002) have shown how to use a compression-based measure to classify fifty languages. The paper was featured in several scientific (and less-scientific) journals, including Nature, Science, and Wired.

## MAIN FOCUS

CDM is quite easy to implement in just about any scripting language such as Matlab, Perl, or R. All that is required is the ability to programmatically execute a lossless compressor, such as gzip, bzip2, compress, WinZip and the like and store the results in an array. Table 1 shows the complete Matlab code for the compression-based dissimilarity measure.

Once pairwise dissimilarities have been computed between objects, the dissimilarity matrix can be used for clustering (e.g. hierarchical agglomerative clustering), classification (e.g., k-nearest neighbors), dimensionality reduction (e.g., multidimensional scaling), or anomaly detection.

The best compressor to capture the similarities and differences between objects is the compressor that compresses the data most. In practice, one of a few easily obtained lossless compressors works well and the best one can be determined by experimentation. In some specialized cases, a lossless compressor designed for the data type provides better results (e.g., DNA clustering, Benedetto et al. 2002). The theoretical relationship between optimal compression and features for clustering is the subject of future research.