SOME ASPECTS OF DEEP REPRESENTATION LEARNING ON TRANSFORMED EEG DATA

Victor IAPĂSCURTĂ[1,2]

[1]*Department of Software Engineering and Automation, Doctoral School, Technical University of Moldova, Chisinau, Republic of Moldova*
[2]*N. Testemitanu State University of Medicine and Pharmacy, Chisinau, Republic of Moldova*

Corresponding author: Victor Iapăscurtă, victor.iapascurta@doctorat.utm.md

Abstract. *Visualizing high-dimensional datasets can be challenging. While it is possible to plot data in two or three dimensions to reveal the data's innate structure, analogous high-dimensional representations are significantly less understandable. A dataset's structure must be shown to some extent, hence the dimension must be decreased. Principal component analysis (PCA) and linear discriminant analysis (LDA) were the two historically the first methods. Several nonlinear techniques were afterwards developed, including locally linear embedding (LLE), multi-dimensional scaling (MDS), isometric feature mapping (Isomap), stochastic neighborhood embedding (t-SNE), etc. In the current study, several nonlinear representation learning techniques are used for electroencephalography (EEG) data with the ultimate objective of categorizing the EEG signal.*

Keywords: *manifold learning, algorithmic complexity, EEG signal, machine learning.*

Introduction

Multidimensional data sets are common in current times. It may be challenging to view and comprehend data due to the vast number of characteristics. Applications commonly use high-dimensional vectors as data representations (gene expression data, drug discovery data, etc.). While working with such data, one may run into the so-called "curse of dimensionality" (a term coined by Richard Bellman in 1957), which describes how high-dimensional algorithms are more difficult to build and frequently have running times that are exponentially related to the dimension. Many dimensionality reduction/representation learning techniques may be used to this issue. Traditionally, there are two categories that these techniques fall under (a) *Linear Representation Learning* (e.g., LDA and PCA), and (b) *Nonlinear Representation Learning* (e.g., LLE, MDS, Isomap, t-SNE, etc.). The Manifold learning hypothesis is one of the fundamental ideas behind representation learning, and it is discussed in more detail below.

Multidimensional data include, for example, multichannel EEG data. In the current study, these data are pre-processed before being utilized for representation learning by calculating their algorithmic complexity (AC) over time.

1. Manifold learning. Formulation of the manifold learning problem

According to the manifold hypothesis, although datasets might be very highly dimensional when they are acquired, the real linkages between the data are found in much smaller dimensional areas (embedded in the high dimensional space). In light of this, it may be said that data analysis essentially looks for this lower-dimensional space (e.g., through dimensionality reduction).

According to [1], the issue is formulated as having a manifold $\mathcal{M}$ of dimension d embedded in an m-dimensional Euclidian space $\mathcal{R}^m$, $d < m$, and $\mathcal{M} = f(\Omega)$ with a mapping $f: \Omega \to \mathcal{R}^m$. Assume we have a set of points $x_1, \dots, x_N$ that were sampled from the manifold $\mathcal{M}$ with noise.

$$x_i = f(\tau_i) + \varepsilon_i, \quad i = 1, \dots, N, \tag{1}$$

where $\{\varepsilon_i\}$ denotes noise, and $\{\tau_i\}$ and/or the mapping $f(\cdot)$ are sought for outcomes from the noisy data $\{x_i\}$. Generally speaking, this issue is referred to as nonlinear dimension reduction or manifold learning. In order to estimate the local structures around each sample point x_i, a class of local techniques for manifold learning first estimates the local structures in question, then aligns them to derive estimates for $\{\tau_i\}$.

2. Algorithmic complexity

The estimation of *algorithmic (Kolmogorov-Chaitin) complexity* carried out using the *Block Decomposition Method (BDM)* from the field of algorithmic information dynamics (AID) [2] is credited with playing a significant role in the data processing flow in this study. The concept of algorithmic complexity (Kolmogorov-Chaitin or program-size) is crucial in this context [3]:

$$K_T(s) = min\{|p|, T(p) = s\}, \quad (2)$$

where K_T is the length of the shortest program p that produces the string s when executed on a universal Turing machine T.

The online algorithmic complexity calculator (OACC), which employs the BDM approach and is based on algorithmic probability specified by the coding theorem method (CTM), is a specific tool included in the AID toolkit for delivering accurate estimates to uncomputable functions [3]:

$$BDM = \sum_{i=1}^{n} CTM(block_i) + log_2(|block_i|). \quad (3)$$

3. The dataset

The data used throughout this paper include a set consisting of algorithmic complexity (AC) time series coming from 36 healthy human volunteers who perform an intensive mental task [4]. The AC is estimated on paired EEG signals obtained before and during the task. The final goal of the research is to investigate the possibility of classifying the EEG signals as recorded before and during the task using transformed (by BDM) EEG signals. The set hereinafter is named "Arithmetic test set"

4. Non-linear representation learning

A system can automatically find the representations required for feature detection or classification from row data using non-linear approaches. Contrary to linear representations, which are supervised learning techniques, non-linear methods have two main types: (a) unsupervised (based on unlabeled data by examining the relationship between points in the data set), and (b) semi-supervised, in which features are learned using unlabeled data, but input-label pairs are constructed from each data point, allowing learning the structure of the data through supervised methods.

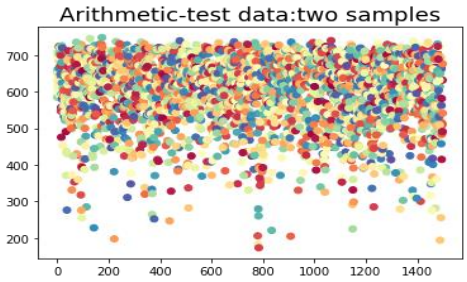

Figure 1. A scatter plot of two samples from Arithmetic test set. X-axis is time and Y-axis is for algorithmic complexity. The spectral color map is used.

The following is the description of a number of non-linear representation learning approaches applied to Arithmetic test data.

4.1. Locally linear embedding

The *Locally linear embedding* (LLE) [5] method computes low-dimensional, neighborhood-preserving embeddings of high-dimensional data to solve the nonlinear dimensionality reduction issue. A data collection of dimensions *n* is mapped onto a single global coordinate system of lower dimensionality, *d*, where it is believed to sit on or close to a smooth nonlinear manifold of dimensionality $d < n$. Locally linear fits are used to reconstruct the global nonlinear structure. The

global internal coordinates on the manifold may be obtained by linearly mapping the high-dimensional coordinates of each neighborhood. So, by computing the locally linear patches in step one and the linear mapping to the coordinate system on the manifold in step two, it is possible to identify the nonlinear structure of the data.

After locating the closest neighbors, a local geometry is generated for each locally linear patch. Linear coefficients that rebuild each data point from its neighbors define this geometry:

$$\min_{w} \sum_{i=1}^{n} \left\| x_i - \sum_{j=1}^{k} w_{ij} x_{N_i(j)} \right\|^2, \tag{4}$$

where *k* is the number of neighbors, *w* are weights, and $N_i(j)$ is the index of the j^{th} neighbor of the i^{th} point. Finally, by completing a similar job for *y*, estimated vectors are created in order to maintain the reconstruction weights.

4.2.Multi-dimensional scaling

Using knowledge of the separations between the *n* patterns, *Multi-dimensional scaling* (MDS) [5] tackles the issue of creating a configuration of *n* points in Euclidean space. If a $n \times n$ matrix *D* is symmetric, has $d_{ii} = 0, \text{and } d_{ij} > 0, i \neq j$, it is referred to as a distance or affinity matrix. Given a distance matrix *D*, the MDS searches for *n* data points in *d* dimensions with $y_1, \dots, y_n$ such that if $\widehat{d_{\iota J}}$ signifies the Euclidean distance between y_i and y_j, then $\widehat{D}$ is identical to *D*. In doing so, MDS reduces

$$\min_{Y} \sum_{i=1}^{n} \sum_{i=1}^{n} \left(d_{ij}^{(X)} - d_{ij}^{(Y)} \right)^2, \tag{5}$$

where $d_{ij}^{(X)} = \left\| x_i - x_j \right\|^2 \; and \; d_{ij}^{(Y)} = \left\| y_i - y_j \right\|^2$.

4.3. Isometric mapping

A nonlinear extension of traditional MDS is called *Isometric mapping* (Isomap) [5]. Similar to LLE, the first step can be carried out by selecting all locations within a certain radius or by determining the *k* nearest neighbors. A graph *G* that connects each data point to its closest neighbors via edges with weights $d_X(i,j)$ represents these neighborhood relations. Then, between each pair of points on the manifold $\mathcal{M}$, the geodesic distances $d_G(i,j)$ are computed.

The data are then generated into an embedding in a *d*-dimensional Euclidean space *Y* by applying traditional MDS to D^G in the last phase of the process. Setting the coordinates of y_i to the top *d* eigenvectors of the inner-product matrix *B* produced by D^G results in the global minimum of the cost function.

4.4. Stochastic neighbor embedding

t-Distributed stochastic neighbor embedding (t-SNE) [6] is suited for the simultaneous preservation of data, recollecting both the local and global framework of the data, in contrast to the techniques mentioned above. Given a collection of high-dimensional objects $N = \{x_1, x_2, \dots, x_N\}$ and the function $d(x_i, x_j)$, which stands for the Euclidean distance, then $d(x_i, x_j) = \left\| x_i - x_j \right\|$. The conditional probabilities $P_{j|i}$ between the comparable objects of data points x_i and x_j are first calculated via t-SNE as follows:

$$P_{j|i} = \frac{\exp(-\left\| x_i - x_j \right\|^2 / 2 * \sigma_i^2)}{\sum_{k \neq i} \exp(-\left\| x_i - x_j \right\|^2 / 2 * \sigma_i^2)}. \tag{6}$$

For the low-dimensional collection of data y_i and y_j assigned to the high-dimensional set of data x_i and x_j, a conditional probability $q_{j|i}$ is determined in a similar way:

$$q_{j|i} = \frac{exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} exp\left(\|y_i - y_j\|^2\right)^2}. \quad (7)$$

When two sets of data are comparable, we set $q_{j|i} = 0$. When t-SNE is used to minimize Kullback-Leibler divergences between *P* and *Q*, the size of the cost function *C* is given by,

$$C = KL(P\|Q) = \sum_{i \neq j} p_{ij} * \log \frac{p_{ij}}{q_{ij}}. \quad (8)$$

4.5. Spectral clustering/embedding

The two primary processes of *Spectral embedding* (SE) [7] are to first embed the data points in a space where clusters are more "apparent," and then to use a conventional clustering method like *k*-means. A data-independent kernel, such as the Gaussian kernel, is initially applied to each pair of instances to create the spectral clustering affinity matrix *K*: $\widetilde{K}_{ij} = \tilde{k}(x_i, x_j)$. The matrix $\widetilde{K}$ is then "divisively" normalized as follows:

$$K_{ij} = \frac{\widetilde{K}_{ij}}{\sqrt{S_i S_j}}, \quad (9)$$

where the S_i are the row sums of $\widetilde{K}$:

$$S_i = \sum_{j=1}^{m} \widetilde{K}_{ij}. \quad (10)$$

After normalizing each embedding vector to have a unit norm, the first *N* main eigenvectors of *K* are calculated in order to produce *N* clusters: the r^{th} coordinate of the i^{th} example is $v_{r,i} / \sqrt{\sum_{l=1}^{N} v_{l,i}^2}$.

Finally, we require a kernel that might have produced that matrix *K* in order to extend spectral clustering to out-of-sample points:

$$k_m(x, y) = \frac{1}{m} \frac{\tilde{k}(x, y)}{\sqrt{\hat{E}_{x'}\left[\tilde{k}(x', y)\right] \hat{E}_{y'}\left[\tilde{k}(x, y')\right]}}. \quad (11)$$

5. Results

The figures below denote the results of applying the nonlinear representation methods described before on the data. Green color denotes pretest algorithmic complexity and red color denotes during-the-test complexity of the EEG signal.

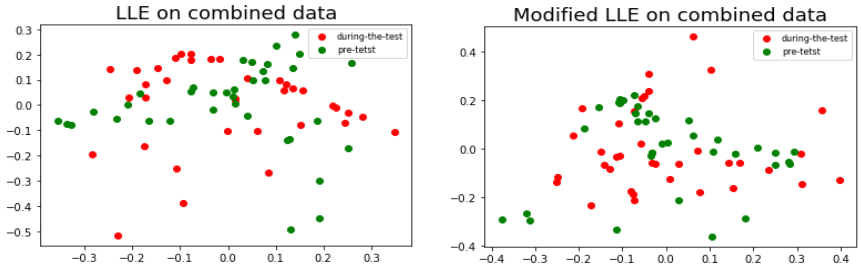

Figure 2. LLE on arithmetic test data; On the left is LLE and on the right is modified LLE.

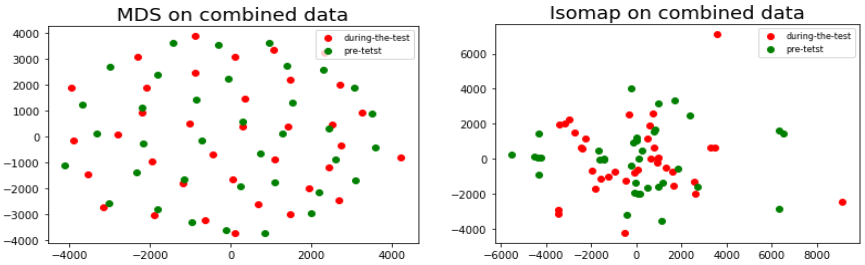

Figure 3. MDS (on the left) and Isomap (on the right) on arithmetic test data.

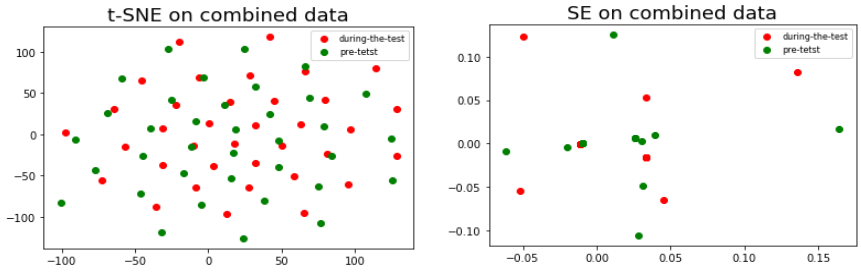

Figure 4. t-SNE (on the left) and SE (on the right) and on arithmetic test data.

6. Conclusions

Since the final goal of the research, the current work is part of, is a classification one, using methods that contribute to "decluttering" the data and providing more comprehensive view on them, is expected to translate into better classification accuracy. And the differences between Figure 1 (i.e., difficult to separate data points) and Figures 2 to 4 (i.e., almost no overlap present) can provide some explanation for a potential increase in accuracy.

Acknowledgments.

The author of this paper would like to thank for valuable guidance Mr. Nistor Grozavu, CY Cergy Paris University**,** who was the holder of the course on Deep Representation Learning attended by the author and Mr. Ion Fiodorov, TUM who is the supervisor of the doctoral research project.

References

1. ZHA, H., ZHANG, Z. Spectral Properties of the Alignment Matrices in Manifold Learning. In: *Review of the Society for Industrial and Applied Mathematics*, 51(3), 2009, pp. 545-566
2. ZENIL, H. A Review of Methods for Estimating Algorithmic Complexity: Options, Challenges, and New Directions. In: *Entropy* 22(6), 2020, 612. doi.org/10.3390/e22060612
3. ZENIL, H. et al. A Decomposition Method for Global Evaluation of Shannon Entropy and Local Estimations of Algorithmic Complexity. In: *Entropy*, 20(8), 2018 p. 605. doi:10.3390/e20080605.
4. ZYMA, I. et al. Electroencephalograms during Mental Arithmetic Task Performance. In: *Data*, 2019, 4(1):14, doi.org/10.3390/data4010014
5. GHODI, A. *Dimensionality Reduction. A Short Tutorial*. Department of Statistics and actuarial science, University of Waterloo, Ontario, Canada, 2006
6. SAKIB, S. et al. Performance Evaluation of t-SNE and MDS Dimensionality Reduction Techniques with KNN, ENN and SVM Classifiers, 2020, arXiv:2007.13487
7. BENGIO Y. et al. Spectral Dimensionality Reduction. In: Guyon, I., Nikravesh, M., Gunn, S., Zadeh, L.A. eds *Feature Extraction. Studies in Fuzziness and Soft Computing*, vol 207. Springer, Berlin, Heidelberg, 2006 https://doi.org/10.1007/978-3-540-35488-8_28