Chemometrics and Intelligent Laboratory Systems 135 (2014) 110–125

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemolab

Visualizing Big data with Compressed Score Plots: Approach and research challenges

José Camacho *

Departamento de Teoría de la Señal, Telemática y Comunicaciones, Universidad de Granada, 18071 Granada, Spain

ARTICLE INFO

Article history:
Received 18 February 2014
Received in revised form 12 April 2014
Accepted 15 April 2014
Available online 24 April 2014

Keywords:
Exploratory Data Analysis
Big data
Principal Component Analysis
Partial Least Squares
Score plots

ABSTRACT

Exploratory Data Analysis (EDA) can be defined as the initial exploration of a data set with the aim of generating a hypothesis of interest. Projection models based on latent structures and associated visualization techniques are valuable tools within EDA. In particular, score plots are a main tool to discover patterns in the observations. This paper addresses the extension of score plots to very large data sets, with an unlimited number of observations. The proposed solution, based on clustering and approximation techniques, is referred to as the Compressed Score Plots (CSPs). The approach is presented to deal with high volume data sets and high velocity data streams. The objective is to retain the visualization capabilities of traditional score plots while making the user-supervised analysis of huge data sets affordable in a similar time scale to that of low size data sets. Efficient processing and updating approaches, visualization techniques, performance measures and challenges for future research are identified throughout the paper. The approach is illustrated with several data sets, including a data set of five million observations and more than one hundred variables.

1. Introduction

According to [1], *Exploratory Data Analysis (EDA) is an approach to data analysis that postpones the usual assumptions about what kind of model the data follow with the more direct approach of allowing the data itself to reveal its underlying structure and model*. EDA has been employed for decades in many research fields, including social sciences, psychology, education, medicine, chemometrics and related fields [2,3]. EDA is both a data analysis philosophy and a set of visualization tools [4]. Nevertheless, while the philosophy has essentially remained the same, the tools are in constant evolution, as numerous recent references suggest [5–8]. This is the direct consequence of the increasing complexity of the problems tackled with data analysis methods thanks to increasing computers capabilities.

The advances in technology in the last decade have led to the so-called Big data era, where exabytes[1] of data are daily generated by humans and, more importantly, machines [9]. This has drawn the attention of the scientific and technological community, driving initiatives for Big data analysis like Hadoop (http://hadoop.apache.org/). Also, extensions of modeling, classification and data mining techniques to Big data, like the Mahout project (http://mahout.apache.org/), have been developed. Unfortunately, the application of the EDA philosophy, which relies so much on visualizations, to Big data problems is challenging due to the large scale of the data sets involved. However, this application deserves attention, since both EDA and modelling applications are complementary, with EDA a suggested first step prior to data modelling [10,11]. Omitting EDA has the risk of misinterpreting the modelling results, as illustrated in [12] with a number of real examples. Therefore, there is the need for developing EDA methods that are suitable to manage the data scales aforementioned, while taking advantage of *the basic importance of simply looking at data* [4].

Big data are commonly defined by the so-called 4 Vs [13]:

- Variety: Data are varied in nature. Different sources, including unstructured and structured information, need to be properly combined in order to make the most of the analysis.
- Veracity: The search for valuable information in large data sets is very much like the problem of finding the needle in a haystack. Big data present low signal to noise ratio, and exploratory and data mining techniques are needed to find patterns or trends of practical use, which are more reliable than punctual measures.
- Volume: The amount of data that needs to be handled simultaneously makes processing parallelism a must. Exabytes, zettabytes, and even higher amounts of data are described in Big data applications.
- Velocity: In Big data problems, a high rate of sampling is common. This further complicates the analysis and makes parallelism even more necessary.

In data sets with a large number of variables, collinear data and missing values, projection models based on latent structures are valuable tools within EDA. Standard well-known projection models are Principal Component Analysis (PCA) [14,15,2] and Partial Least Squares (PLS)

* Tel.: +34 958 248898; fax: +34 958 240831.
E-mail address: josecamacho@ugr.es.

[1] one exabyte corresponds to 2^{18} bytes or one million terabytes.

http://dx.doi.org/10.1016/j.chemolab.2014.04.011
0169-7439/

[16–18]. These models and the set of tools used in combination [19–22] simplify the visual analysis of complex data sets. While the calibration of projection models from large scale data has already been studied [23, 24], the extension of the visual tools in this context has not been treated. One of the most used visualization tools in the context of projection models is the score plot. The score plot is a very useful tool to discover the distribution of the observations, including special observations (outliers) and clusters of related observations. All this information may be of paramount importance to improve data knowledge, and a valuable starting point for the selection and proper application of modeling and data mining tools.

Projection models have outstanding capabilities for combining data from different sources and for handling uncertain data. This covers two of the aforementioned 4 Vs of Big data. Addressing the other two, Volume and Velocity, is the focus of this paper. For this, an approach to extend the score plots to Big data, the compressed score plots (CSPs), is introduced. It is argued that this extension is of interest not only for the chemometric community, but for the general community involved in Big data analytics. For this reason, the examples used to illustrate the approach are not limited to chemometric data. Limitations and research challenges in the proposed approach are pointed out throughout the document.

The paper is organized as follows. Section 2 introduces PCA and PLS and their iterative computation. Section 3 discusses the limitations of traditional score plots to visualize a large number of observations. This is illustrated using a data set collected from a continuous process. Section 4 introduces the proposed solution to that limitation: the Compressed Score Plots. Section 5 introduces a methodology to update Compressed Score Plots when the subspace of interest changes. Section 6 illustrates the complete approach with two additional cases studies. Section 7 presents conclusions and future research challenges.

2. Projection subspaces

Both PCA and PLS provide a similar solution to the same problem: data collinearity. The approach of these methods to overcome the problems derived from collinearity is to identify a reduced number of new variables, referred to as latent variables (LVs) or specifically in PCA as principal components (PCs). These LVs are obtained as a combination of the original variables in the data. In standard PCA and PLS, the LVs are linear combinations of the original variables, but non-linear extensions also exist [25]. For a given data set, the LVs are found by maximizing a given quadratic function, variance in the case of PCA and covariance for PLS. The operation to obtain the LVs from the original variables can be geometrically interpreted as a projection operation. Thus, projection models can be understood as projection subspaces of the original variables space.

PCA follows the expression:

$$\mathbf{X} = \mathbf{T}_A \cdot \mathbf{P}_A^t + \mathbf{E}_A, \tag{1}$$

where $\mathbf{T}_A$ is the $N \times A$ score matrix containing the projection of the observations in the A PCs sub-space, $\mathbf{P}_A$ is the $M \times A$ loading matrix containing the A eigenvectors of $\mathbf{X}^T \cdot \mathbf{X}$ with highest associated eigenvalues and $\mathbf{E}_A$ is the $N \times M$ matrix of residuals. The number of PCs retained in a PCA model, A, is a principal choice [15,26], which in general can be regarded as an application dependent decision [27,28].

PLS performs a biased solution of the linear regression problem to the least squares solution. The linear regression problem is defined by the following expression:

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{B} + \mathbf{F} \tag{2}$$

where $\mathbf{Y}$ is the $N \times K$ matrix of variables that are to be estimated, $\mathbf{X}$ is the $N \times M$ matrix of variables available to estimate $\mathbf{Y}$, $\mathbf{B}$ is the $M \times K$ matrix of regression coefficients and $\mathbf{F}$ is the $N \times K$ matrix of residuals. A possible way to interpret $\mathbf{B}$ is as a model of $\mathbf{Y}$, with $\mathbf{X}$ being the input to the model.

The aim of PLS regression is to estimate $\mathbf{Y}$ from the subspace of $\mathbf{X}$ that maximizes its covariance with $\mathbf{Y}$. The partial linear regression problem between normalized matrices $\mathbf{X}$ and $\mathbf{Y}$ can be stated as:

$$\begin{aligned} \mathbf{X} &= \mathbf{T}_A \cdot \mathbf{P}_A^T + \mathbf{E}_A \\ \mathbf{Y} &= \mathbf{T}_A \cdot \mathbf{Q}_A^T + \mathbf{F}_A \end{aligned} \tag{3}$$

where $\mathbf{T}_A$ is the $N \times A$ score matrix which contains the projections of $\mathbf{X}$ to the latent A-dimensional subspace, $\mathbf{P}_A$ and $\mathbf{Q}_A$ are the $M \times A$ and $K \times A$ regressor matrices, also called loading matrices, and $\mathbf{E}_A$ and $\mathbf{F}_A$ are the $N \times M$ and $N \times K$ matrices of residuals of $\mathbf{X}$ and $\mathbf{Y}$, respectively. Eq. (3) can be rearranged in the following form:

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{R}_A \cdot \mathbf{Q}_A^T + \mathbf{F}_A \tag{4}$$

with:

$$\mathbf{R}_A = \mathbf{W}_A \cdot \left(\mathbf{P}_A^T \cdot \mathbf{W}_A\right)^{-1} \tag{5}$$

where $\mathbf{W}_A$ is a $M \times A$ matrix of weights. Thus, a PLS model is represented by matrices $\mathbf{P}_A$, $\mathbf{W}_A$ and $\mathbf{Q}_A$.

A useful variant of PLS for supervised classification in EDA is PLS-Discriminant Analysis (PLS-DA) [29]. In PLS-DA, matrix $\mathbf{Y}$ is artificially generated with dummy variables, which codify the different classes in the data set. Typically, $\mathbf{Y}$ is constructed with as many variables as classes. All the observations belonging to a class have value 1 for the corresponding variable, and -1 (or 0) for the rest.

In the following subsections, some problems arisen to fit projection models from Big Data are discussed. Since the main goal of the present paper is to provide a solution to data visualization, model fitting problems are only briefly reviewed and a practical solution based on the iterative computation of cross-product matrices is adopted.

2.1. Computation of projection models from large volumes of data

There are two problems of interest to the scientific community in the calibration of projection models from very large data sets. The main problem is to compute the model out-of-core [30], that is, without maintaining the whole data set in the main memory of the computer. A second problem is the development of fast, approximate and computationally efficient calibration algorithms [31,32]. Model fitting does not represent the most computational intensive step in the proposal of this paper. For this reason, this section is only devoted to the first problem.

Several algorithms for PCA or PLS model fitting take the calibration data set $\mathbf{X}$ (and optionally $\mathbf{Y}$), with N observations, as input. Due to limited computer resources, in particular computer memory, this approach is infeasible when N grows beyond a certain number, as is the case for large volume data sets. In those cases, cross-product matrices can be used for model fitting. The loading vectors of PCA can be identified using the eigendecomposition (ED) of the cross-product matrix $\mathbf{XX} = \mathbf{X}^T \cdot \mathbf{X}$. Similarly, the loadings and weights in PLS regression can be identified from matrices $\mathbf{XX}$ and $\mathbf{XY} = \mathbf{X}^T \cdot \mathbf{Y}$ using the kernel algorithm [33–35].

The computation of these cross-product matrices can be performed in an iterative manner. This procedure assumes the data have been previously preprocessed, which can also be performed in an iterative fashion. For the sake of generality and to reduce computational overhead, the cross-product matrices are updated for each batch of observations of size B, instead of for each single observation. The observation-wise iterative computation can be derived for $B = 1$. The iterative computation follows:

$$\mathbf{XX}_t = \mathbf{XX}_{t-1} + \mathbf{X}_t^T \cdot \mathbf{X}_t, \tag{6}$$