Computers and Chemical Engineering 98 (2017) 113–127

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

On Generative Topographic Mapping and Graph Theory combined approach for unsupervised non-linear data visualization and fault identification

Matheus S. Escobar, Hiromasa Kaneko, Kimito Funatsu *

Chemical System Engineering Department, The University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8656, Japan

ARTICLE INFO

Article history:
Received 18 October 2016
Received in revised form 5 December 2016
Accepted 12 December 2016
Available online 18 December 2016

Keywords:
Generative Topographic Mapping
Graph clustering
Fault detection
Fault identification
Process monitoring

ABSTRACT

Process monitoring of chemical plants relies on two steps: discriminating anomalies (fault detection) and characterizing them (fault identification). This work proposes a combined Generative Topographic Mapping (GTM) and Graph Theory (GT) approach. GTM highlights system features, reducing variable dimensionality and providing a strategy for calculating similarity between samples. GT then clusters them using networks, discriminating normal and anomalous entries. Because of biased normal and anomalous labeling, however, the methodology proposed is unsupervised, meaning that labels are inexistent. Three case studies were considered: a simulation data set, Tennessee Eastman process and an industrial data set. Principal Component Analysis (PCA), dynamic PCA and kernel PCA indexes (Q and T^2) alongside GTM and GT independent monitoring methodologies were used for comparison, considering supervised and unsupervised approaches. For the industrial scenario, soft sensors were used for assessing discrimination performance. The proposed method, while unsupervised, discriminated normal states similarly to supervised strategies, justifying its development.

1. Introduction

Highlighting important characteristics and features of a process is fundamental for all sorts of applications. Machine learning techniques are particularly interesting when tackling those issues, due to their non-phenomenological approach, which relies on the process data available. For this work, the main focus is process monitoring, more specifically fault identification and process data visualization. It is important not only to distinguish between normal and abnormal states in the plant, but also to be able to visualize how process data can be reconciled to reveal different characteristics of the process.

Evidently, anomalies have several sources, such as hidden plant states, disturbances and equipment malfunction, to name a few. Different faults lead to different approaches and applications. Database maintenance, for example, aims to only store relevant data by creating from a training data set a detection model that can identify normal and abnormal samples for future online evaluation. Such technique can, then, improve soft sensors' (Kaneko et al., 2011) accuracy, by only indicating reliable samples for model generation. Process control applications also benefit greatly from it, where developments in alarm technologies and hierarchical control systems can ensure better response to anomalous scenarios.

This work relies mainly on Multivariate Statistical Process Control (MSPC) and Monitoring (MSPM) (Kourti, 2005; Bersimis et al., 2007; Prasad et al., 1995; Nicolotti and Carotti, 2006) aspects. Our concern is to evaluate and visualize how variables and samples interact with the process and with each other. By doing so, we hope to achieve a more complete understanding of the process, where data discrimination is more objective and meaningful. One could expect us to consider a more conventional supervised tactic, where a previously known stable state is used as reference, with samples flagged as normal or anomaly (Chiang et al., 2000; Russell et al., 2000). In many cases, though, labels might not be openly available, reliable or even existent. Biases naturally arise in any analysis, leading to labeling errors. Outdated methodologies might not be representing the process in its entirety, or data labeling might have been poorly assessed. This leads to considering unsupervised methodologies for monitoring, which would provide a fresh perspective on data. By not relying on any labels, only the relationship between variables and their evolution over time is used for data discrimination.

When developing unsupervised methodologies, however, several factors have to be taken into account. Primarily, the quality of

* Corresponding author.
E-mail address: funatsu@chemsys.t.u-tokyo.ac.jp (K. Funatsu).

http://dx.doi.org/10.1016/j.compchemeng.2016.12.009
0098-1354/

the information available is fundamental for the development of trustworthy models. Real data sets struggle with redundant information and noise, which might hide the true relation between different features and, therefore, different samples. Dimensionality reduction, thus, identifies regions with similar characteristics and filters redundant information from data. One of the most widespread methods for process monitoring is Principal Component Analysis (PCA) (Jolliffe, 2002), which assesses linear correlation between different process variables, so to reduce the dimensionality of highly correlated variables. Its use is so widespread that numerous PCA-based MSPMs were developed, such as dynamic PCA (DPCA) (Russell et al., 2000), recursive PCA (Li et al., 2000), distributed PCA (Ge and Song, 2013) and maximum-likelihood PCA (Choi et al., 2005). Extensions were developed to overcome its linear nature and to deal with non-linear systems, such as kernel PCA (Lee et al., 2004). Other methods also tackle non-linearity from scratch, such as Support Vector Machines (SVM) (Kittiwachana et al., 2010), Gaussian Mixture Models (GMM) (Yu and Qin, 2008), Generative Topographic Mapping (GTM) (Bishop et al., 1998), and even the use of inferential models (Masuda et al., 2014).

Focusing on strategies developed so far for fault identification, the main element explored in this work builds upon a previous development (Escobar et al., 2015), whose focus is on GTM and Graph Theory. GTM's non-linear and probabilistic nature leads to a better handling of complex and realistic scenarios. When it comes to unsupervised fault identification, one key aspect is how to assess data similarity, so that similar samples, and therefore similar states, can be clustered. Each sample plotted in GTM's latent space has a unique probability distribution (PD), a fingerprint, associated to each latent grid point. By assuming that samples with correlated PD profiles represent data with similar characteristics, GTM can be used for fault identification and dimensionality reduction simultaneously, including discrimination of normal and anomalous data. Clustering is performed by Graph Theory (GT) (Harary, 1994), where similarity information is used for establishing a network. Then, its density and number of connections unravel clusters with different characteristics. This methodology, called GTM+GT (Escobar et al., 2015), can successfully discriminate normal and anomalous data, however there is much room for improvement. This work aims to explore GTM+GT even more, revealing different features of this combined approach, allowing better refinement of normal clusters and revealing a myriad of different interpretations for the networks established.

Three case studies are defined for performance comparison. Initially, a simulation data set with multiple anomaly scenarios is created. Secondly, Tennessee Eastman process (TEP) (Downs and Vogel, 1993) is considered for validation of the methodology. Finally, an industrial case study is used for final comparison and validation. The proposed method (GTM+GT) is compared against unsupervised and supervised PCA, DPCA, KPCA, GTM. Section 2 presents a review on dimensionality reduction and GT. Section 3 presents all fault identification methods considered for comparison in this work. Section 4 describes in detail the proposed method. Section 5 presents several results, discussing the impact of different methodologies on anomaly detection. Section 6 presents final remarks and future work.

2. Dimensionality reduction and Graph Theory

2.1. Principal Component Analysis

Visualization of the relationship between distinct variables can be rather complex, especially if the system possesses many variables or is non-linear. PCA is the most straightforward linear approach known, relying on variables being converted into linearly uncorrelated variables called Principal Components (PC), through an orthogonal transformation (Jolliffe, 2002; Colliandre et al., 2012). The basic logic behind PCA can be seen in Eq. (1).

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathrm{E} \tag{1}$$

X is the original data set matrix, **T** is the score matrix, **P** is the loading matrix and **E** is the residual matrix. **P** establishes the relation between **X** and **T**, resulting in the projection of **X** values onto the transformed space **T**, where the PCs are its column vectors. Each PC contributes to the original data contained in **X** proportionally to their eigenvalue. Such effect can be expressed as the equation described in Eq. (2), considering data has been auto-scaled.

$$C_{t_i} = \frac{\sigma^2(\mathrm{t_i})}{M} \tag{2}$$

C_{t_i} is the component contribution for PC $\mathbf{t}_i$ and M is the number of input variables. The main goal is to select only those PCs that contain relevant information, excluding the rest. The heuristics considered in this work keeps only those components whose accumulated component contribution is just below 99%.

Since one of its main limitations is its inherent linear nature, which limits its application for more complex, non-linear systems, other techniques were developed to cope with that, such as kernel PCA (Choi et al., 2005), one of the most popular PCA extensions. In this work, we are focused on the original PCA, DPCA (Lin et al., 2000) and kernel PCA, to be explained in details in the next subsections.

2.2. Dynamic Principal Component Analysis

DPCA extends the regular PCA concept by introducing dynamics to better understand and represent non-linear time series processes. The methodology itself is remarkably simple. Time shifted variables are added as extra features, establishing a relation between current and past samples (Russell et al., 2000). Eq. (3) shows how to represent this new variable set.

$$\mathrm{X_{Dyn}} = [\mathrm{X_1}, \mathrm{X_2}, \ldots, \mathrm{X_d}] = \begin{bmatrix} \mathrm{x_{d+1}} & \mathrm{x_d} & \cdots & \mathrm{x_1} \\ \mathrm{x_{d+2}} & \mathrm{x_{d+1}} & \cdots & \mathrm{x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{x_N} & \mathrm{x_{N-1}} & \cdots & \mathrm{x_{N-d}} \end{bmatrix} \tag{3}$$

$\mathbf{X}_i$ is the original data set being delayed, N is the total number of samples and d is the sample delay. $\mathbf{x}_n$ is a row vector with all variables for the nth sample. DPCA has the same approach as PCA, but with extra time shifted column vectors. Thus, all analysis related to PCA, such as determining the optimal number of principal components, apply to DPCA as well.

2.3. Kernel Principal Component Analysis

Kernel Principal Component Analysis (KPCA) relies on efficiently computing principal components in high-dimensional feature spaces, using integral operators and non-linear kernel functions (Lee et al., 2004). The concept behind KPCA is simple, where linearly inseparable data are transformed (projected) onto a new feature space, providing better discrimination. The mapping of a sample $\mathbf{x}_i$ can be written as $\mathbf{x}_i \rightarrow \boldsymbol{\phi}(\mathbf{x}_i)$, where $\boldsymbol{\phi}$ is called kernel function.

Instead of applying PCA on the original data, a kernel matrix **K** is used, where each matrix element $k(\mathbf{x}_i, \mathbf{x}_j)$ is defined by the dot products shown in Eq. (4)

$$k(\mathrm{x_i}, \mathrm{x_j}) = \langle \phi(\mathrm{x_i}), \phi(\mathrm{x_j}) \rangle \tag{4}$$

Kernels can map non-linear data in distinct ways. Sigmoidal, Polynomial and Gaussian kernels (Shawe-Taylor and Cristianini,