DATA DISTILLATION, ANALYTICS, AND MACHINE LEARNING

S. Joe Qin[1,2] and Yining Dong[2]
[1] The Mork Family Department of Chemical Engineering and Materials Science
[2] Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089 USA
Email: sqin@usc.edu

Abstract

In this paper we first provide a brief overview on latent variables modeling methods for process data analytics and the related objectives to distill desirable components or features from a mixture of measured variables. These methods are then extended to modeling high dimensional time series data to extract the most dynamic latent variables one after another, which are referred to as *principal time series,* with the current values being best predicted from the past values of themselves or a different set of variables. We show how real process data are efficiently and effectively modeled using these dynamic methods to extract features in process operations and control, leading to new perspectives on how industrial process data can be indispensable for data-driven process operations and control.

1 Introduction

The available massive amount of data has prompted many disciplines and industries to reexamine their traditional paradigms and views, such as statistics, management science, econometrics, computer science, and engineering. As a result, a new discipline known as *data science* is forming to derive knowledge and information from massive data. Several examples have shown that the possession of huge amount of data has tremendous advantage when combined with effective analytics and superior computing power to distill knowledge from data. The Google's flu prediction is such an example (Ginsberg et al., 2009), which could predict the spread of the winter flu outbreak in 2009 in the United States and down to the states level. Google took 50 million most common searches and compared them to the Center for Disease Control (CDC) data on the spread of winter flu from 2003 to 2008. Google's data processing power screened through 150 million models to discover 45 features with a mathematical model that had high correlation to the data from CDC. In addition, Google could predict nearly in real time, while CDC's data took weeks to compile. While this data analytic approach is entirely new to chemical engineers, the functionality of the models is known as *inferential sensors* and practiced in process systems engineering (Tham, 1991; Qin and McAvoy, 1992).

Process operation data are usually massive and high dimensional due to the complexity of the process and control. The process measurement and process analytical technologies (PAT) range from conventional process sensors such as temperature and flow-rate to concentrations, spectra, and images. Although process operations data are high dimensional, the measurement vector space is far from being fully excited due to process operation requirements and physical constraints. For these data traditional regression methods such as least squares fail to yield reliable answers due to

- High colinearity among the measurement data that leads to ill-conditioning or numerical problems.
- Even though the numerical problems can be circumvented by using techniques like pseudo-inverse, the statistical properties of the models are poor such as inflated variance.

- Regularization methods such as ridge regression can be used and tuned to achieve reliable prediction models by trading-off bias and variance, which basically shrinks the magnitudes of the model parameters. However, these models are not easily interpretable, whereas an important purpose of data modeling is interpretability.

The high dimensional data, whether normal or abnormal, are often driven or excited by a few dominant factors that propagate to all measurements via the process units, controls, and operations. To analyze these data effectively, latent variables methods (LVM), including principal component analysis (PCA), projection to latent structures (PLS), and canonical correlation analysis (CCA) are preferred. For brevity of the paper, we will not provide a historical perspective of the latent variable methods in process applications. Interested readers should refer to the work of MacGregor and Koutoudi (1995), Wise and Gallagher (1996), and Qin (2003).

In this section we review the traditional latent variable methods that are the basis for extending to dynamic and nonlinear analytics tools. First we give the context in which the process and quality data are collected and monitored. Then we illustrate the objectives of each LVM and comment on their advantages and shortcomings. Lastly we give an analogy of the latent-variable modeling that extract component by component to that of a distillation process that separates chemical components from a mixture of solutions.

In the remainder of this paper we offer a brief introduction to the essence of latent variable analytics in Section 2. We then present dynamic latent variable methods for the modeling of time series data for prediction, decision-making, and feature analysis in Section 3. The methods are demonstrated in Section 4 on a real process data set to extract principal time series that are best predicted by its past and are easily used to visualize features hidden in the original data. In the end of the paper we encourage an open mindset towards embracing the power of new machine learning techniques that have enjoyed tremendous development in the last 20 years.

2 Data Analytics Using Latent Variables

2.1 Process, Data, and Monitoring

The process and quality data considered for process data analytics can be illustrated Figure 1, where the hierarchical data structure is shown. At the bottom level are the equipment sensor measurements that can be in milliseconds. At the process level are regularly sampled process control data. The product quality measurements come in all forms and often irregularly sampled. The top level is the customer feedback data that can go from customer service channels to social network complaints. The advantages of the latent structure modeling methods, such as PCA and PLS, are that they can be used to detect abnormal changes in process operations from real time data due to the dimension-reduction capability, ease of visualization, and ease of interpretation. The related fault diagnosis methods have been intensively studied and applied successfully in many industrial processes, e.g. chemicals, iron and steel, polymers, and semiconductor manufacturing.

Process data are often categorized into process input, process output, quality output, and indirect (e.g., vibration signals and images) types of data, as shown in Figure 1. The typical procedure of the multivariate process data analytics is

- Collection of (clean) normal data with good coverage of the operating regions
- Fault data cases can be useful, but not required *a priori*
- Latent structure methods (PCA, PLS, etc.) to model the data
- Fault detection indices and control limits, such as the Hotelling T-square and the squared prediction error indices

- Fault diagnosis and troubleshooting, such as reconstruction-based fault identification and contribution analysis.

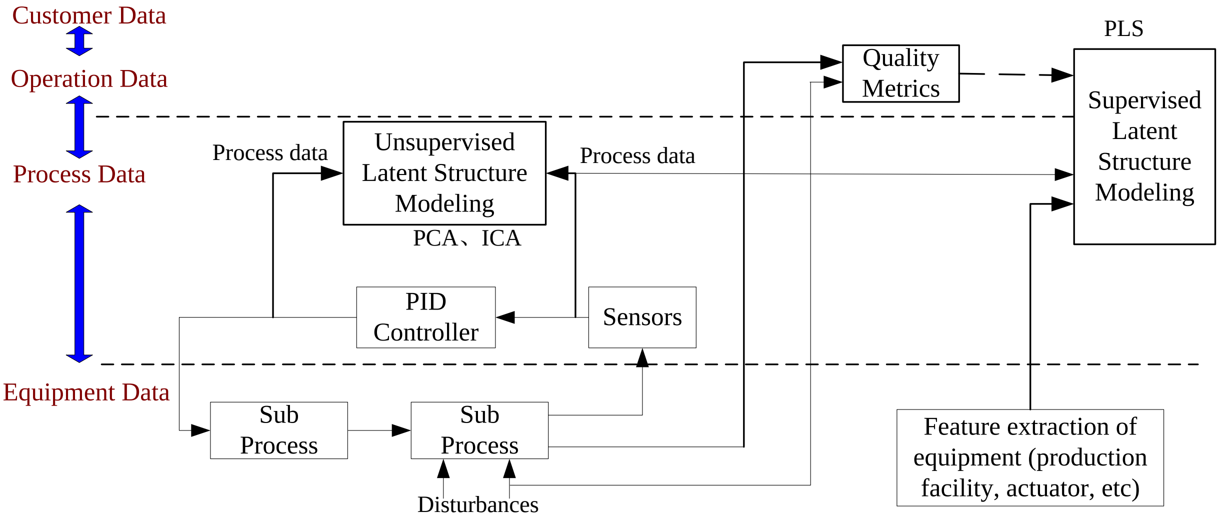


Figure 1. Process and quality data collected under a process and control hierarchy.

2.2 Latent Variable Methods

The objective of PCA is, from a number of data observations, to represent a number of typically correlated variables with a reduced number of latent variables that are most representative for the original variables. Without any prior requirement it is natural to extract the latent variables such that they capture the largest variation in the original data and, therefore, the residuals will be minimal. The extracted latent variables (LV) or principal components (PC) can be easily visualized with low dimensional plots or interpreted with physical understanding of the process behind the observed data. From the latent variables point of view, the measured data are merely various observations that are driven by the underlying latent variables which are not directly measured.

Let *x* denote a sample vector of M variables. Assuming that there are N samples for each variable, a data matrix **X** is composed with N rows (observations) and M columns (variables) as follows

$$\mathbf{X} = \begin{bmatrix} \boldsymbol{x}^T(1) \\ \boldsymbol{x}^T(2) \\ \cdots \\ \boldsymbol{x}^T(N) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_M \end{bmatrix}$$

For convenience the variables are usually scaled to zero mean and unit variance. Principal component analysis extracts a direction or subspace of the largest variance in the M dimensional measurement space. For an arbitrary vector direction $\mathbf{p} \in \Re^M$, such that $\|\mathbf{p}\| = 1$, the projection of **X** on to this direction is $\mathbf{t} = \mathbf{X}\mathbf{p}$. The PCA objective is to maximize the variance along this direction, that is,

$$\max\ \mathbf{t}^T\mathbf{t} = \mathbf{p}^T\mathbf{X}^T\mathbf{X}\mathbf{p}$$

The solution to the above problem with $\|\mathbf{p}\| = 1$ as a constraint can be obtained using a Lagrange multiplier as follows

$$\mathbf{X}^T\mathbf{X}\mathbf{p} = \lambda\mathbf{p}$$

which implies that **p** is the eigenvector corresponding to the largest eigenvalue of the covariance matrix of **X**. The vector **p** is known as the loading vector for the first principal component. After the first component is extracted and removed from the data matrix, the same eigen-decomposition procedure is iterated on the residual,

$$\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i \mathbf{p}_i^T$$

The data matrix, being the first one in the iteration, is decomposed as follows.

$$\mathbf{X} = \sum_{i=1}^{l} \mathbf{t}_i \mathbf{p}_i^T + \mathbf{X}_{l+1}$$

If the data matrix **X** contains highly correlated columns, it will take fewer components than *M* to leave little variance in the residuals. The variance of the extracted PC scores, $\mathbf{t}_i$., corresponding to the eigenvalues in descending order of magnitude. This is analogous to separating a more volatile chemical component from a mixture of less volatile ones in a distillation process.

Partial least-squares methods find a latent structure between data matrices, **X** and **Y**, collected from input variables and output variables, such that the respective score vectors

$$\begin{aligned} \mathbf{t} &= \mathbf{X}\mathbf{w} \\ \mathbf{u} &= \mathbf{Y}\mathbf{q} \end{aligned}$$

have maximized covariance. Mathematically this is expressed as

$$\max_{\mathbf{t},\mathbf{u}} J = \mathbf{t}^T \mathbf{u}$$

subject to the constraint that the weighting vectors **w** and **q** have unit norm,

$$\begin{aligned} \|\mathbf{w}\|^2 &= 1 \\ \|\mathbf{q}\|^2 &= 1 \end{aligned}$$

The solution to this problem can also be achieved by using Lagrange multipliers, which lead to an eigen-problem related to the two data matrices. Deflations and iterations are necessary to extract all significant latent variables one after another.

Due to the use of a covariance objective function in PLS, it usually requires multiple latent variables even to predict a single output variable in **Y**. One arguable advantage of needing multiple LVs is that the method exploits the variance of the input while trying to predict the output. This is, nevertheless, trying to achieve two objectives at once, which can sometimes compromise both objectives. For instance, there is usually a significant portion of the latent variable subspace that is orthogonal or irrelevant to the output, although that subspace contains significant variability of the input data. This is the motivation of several subsequent efforts to develop orthogonalized PLS (Sun et al., 2009) and concurrent PLS methods (Qin and Zheng, 2013).

An alternative objective is the canonical correlation analysis (CCA) objective developed by Hotelling (1936) to maximize the *correlation* between two sets of latent vectors **t** and **u**,

$$\max_{\mathbf{t},\mathbf{u}} J = \frac{\mathbf{t}^T \mathbf{u}}{\|\mathbf{t}\| \|\mathbf{u}\|}$$

which is also the cosine of the angle between the latent vectors. The solution to this problem is an eigen-vector solution of $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{X}$. An advantage of the CCA method is that it has maximized efficiency in predicting the output **Y** using variations in **X**. For the single output case, CCA requires only one latent variable to extract all variations in the input data to predict or interpret the output. However, due to the inverses of the covariance matrices involved in the CCA solution, it is sensitive to colinearity among the variables. Some form of regularization is necessary to make the method insensitive to collinear data. Another issue is that CCA has no attention to the input variances, other than the portion that is useful in predicting the output. This makes CCA incapable of exploiting the input variance structure. The recently developed concurrent CCA (Zhu et al., 2016) combines CCA and PCA to achieve two objectives concurrently, this is, to exploit the variance structure of the input while predicting the output efficiently.

The aforementioned methods all exploit latent structured relations among the variables that are linear and static. They form the foundation for extensions to nonlinear or dynamic latent structure modeling. Since all methods have clear objectives factor by factor, they are analogous and can be interpreted as distilling needed components from data one after another, with respective objectives and intentions.

3 Dynamic Data Distillation Using Latent Variables

The vast amount of process data are collected in the form of *time series* with regular sampling intervals. These data are often collected at the process level and the equipment level, making the sampling intervals very high, from seconds to milliseconds. Dynamics or time correlations are inevitable among the data and they are useful for prediction and interpretation. Given the fact that the large dimensional time series data are both cross-correlated and auto-correlated over time, it is necessary to develop dynamic versions of the latent variables methods such as PCA, PLS and CCA, such that the variables' current data are best predicted by the past data of themselves or other variables, using a reduced number of dynamic latent variables. The extracted data for these dynamic latent variables are referred to as *principal time series*, with reduced dimensions, which can be best predicted from the past data of themselves or another set of variables.

3.1 PCA with Dynamic Latent Variables

In this subsection, dynamic-inner principal component analysis (DiPCA) is presented to build most dynamic relations of the inner latent variables. DiPCA extracts one or more latent variables that are linear combinations of the original variables and have maximized auto-covariance. In other words, the current values of these latent variables are in a sense best predictable from their past values. In the complement, the residuals after extracting the most predictable latent variables from the data will be least predictable and, in the limiting case, tend to be white noise. The method overcomes drawbacks of existing dynamic PCA methods that perform static PCA on simply augments time lagged data (e.g., Ku et al. 1995).

The advantages of the DiPCA algorithm that extracts principal time series are that i) the dynamic components can be predicted from their past data as known information, so that the uncertainty is the prediction errors only; ii) the extracted dynamic components can reveal useful dynamic features for data interpretation and diagnosis, which are otherwise difficult to observe from the original data; and iii) the prediction errors after all dynamics are effectively extracted, can be further modeled as static data with the static PCA method (Dong and Qin, 2016).

In general, we wish to extract dynamics in a latent variable t_k so that the current value can be predicted from its past, for instance, as follows,

$$t_k = \beta_1 t_{k-1} + \cdots + \beta_s t_{k-s} + r_k$$

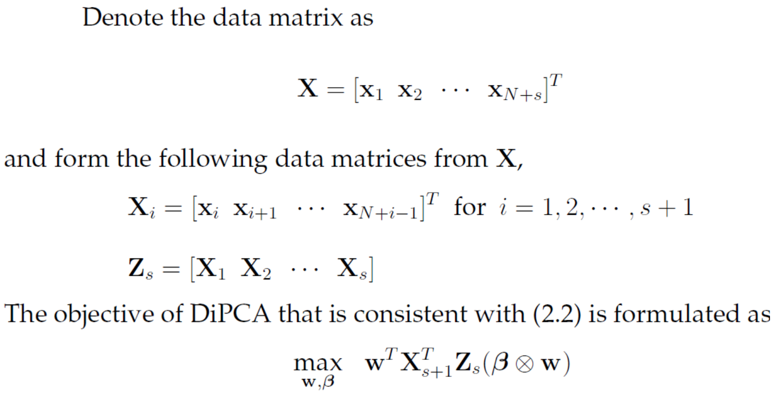
with the latent variable as a linear combination of the original variables $t_k = \mathbf{x}_k^T \mathbf{w}$. The prediction from the dynamic inner model is

$$\begin{aligned}\hat{t}_k &= \mathbf{x}_{k-1}^T\mathbf{w}\beta_1 + \cdots + \mathbf{x}_{k-s}^T\mathbf{w}\beta_s \\ &= [\mathbf{x}_{k-1}^T \cdots \mathbf{x}_{k-s}^T](\boldsymbol{\beta} \otimes \mathbf{w})\end{aligned}$$

where $\boldsymbol{\beta} = [\beta_1 \quad \beta_2 \cdots \beta_s]^T$ and **w** are constrained to be unit norm without loss of generality. The objective of the dynamic inner PCA algorithm is to maximize the covariance between the extracted data and the prediction, that is

$$\frac{1}{N}\sum_{k=s+1}^{s+N} \mathbf{w}^T\mathbf{x}_k[\mathbf{x}_{k-1}^T \cdots \mathbf{x}_{k-s}^T](\boldsymbol{\beta} \otimes \mathbf{w})$$

For a number of observations, Dong and Qin (2016) reformulate the above objective in matrix notation as follows.



Denote the data matrix as

$$\mathbf{X} = [\mathbf{x}_1 \;\; \mathbf{x}_2 \;\; \cdots \;\; \mathbf{x}_{N+s}]^T$$

and form the following data matrices from $\mathbf{X}$,

$$\mathbf{X}_i = [\mathbf{x}_i \;\; \mathbf{x}_{i+1} \;\; \cdots \;\; \mathbf{x}_{N+i-1}]^T \;\text{ for }\; i = 1, 2, \cdots, s+1$$

$$\mathbf{Z}_s = [\mathbf{X}_1 \;\; \mathbf{X}_2 \;\; \cdots \;\; \mathbf{X}_s]$$

The objective of DiPCA that is consistent with (2.2) is formulated as

$$\begin{aligned}&\max_{\mathbf{w},\boldsymbol{\beta}} \quad \mathbf{w}^T\mathbf{X}_{s+1}^T\mathbf{Z}_s(\boldsymbol{\beta} \otimes \mathbf{w}) \\ &\text{s.t.} \quad \|\mathbf{w}\| = 1, \|\boldsymbol{\beta}\| = 1\end{aligned}$$

The complete DiPCA algorithm is given in Appendix A, while more detail about the DiPCA properties can be found in Dong and Qin (2016). With the objective of maximizing the covariance between the latent variable and its prediction from the past, DiPCA performs dynamic data distillation from all measured data such that the extracted dynamic components co-varies the most with its past. The prediction errors of the data after the first predicted component are then used to extract the second most co-varying latent component, until all significant dynamic components are extracted. This procedure is analogous to a multi-stage binary distillation process, with each stage separating a most dynamic co-varying component from the rest. After all components are extracted, the prediction errors are essentially un-autocorrelated. Figure 2 illustrates how DiPCA distills dynamic latent components one after another, with the objective to maximize the covariance of the component with the prediction from its past. High dimensional time series data are considered as a mixture of a number of dynamic latent components, which are not measured directly, and static variations. DiPCA distills the multi-dimensional data into dynamic components in descending order of covariance.

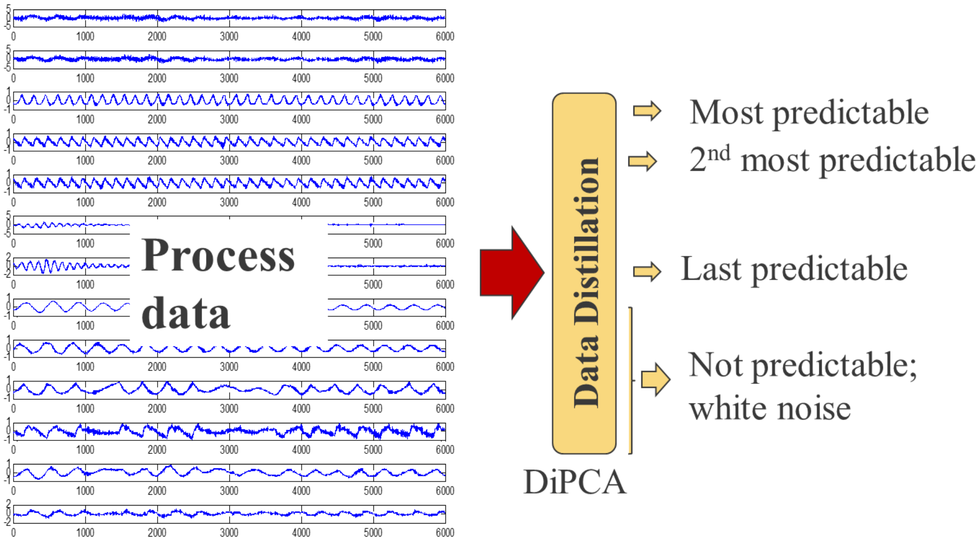


Figure 2. DiPCA is a process of distilling dynamic latent components one after another, with the objective to maximize the covariance of the component with the prediction from its past.

This DiPCA can also be viewed as a whitening filter applied to the data. After all DiPCA components are extracted, the prediction errors are essentially white as virtually all the dynamic relationships in data are extracted. An important notion of this whitening filter is that it has a reduced number of latent variables compared to the number of variables in that data, and is appropriate for modeling the common case of highly collinear data from real world problems. This solution is different from a full dimensional vectored autoregressive model that requires to invert a covariance matrix that can be ill-conditioned with highly correlated data. Furthermore, the DiPCA latent variables have a clear objective and can provide useful features for data based interpretation, visualization, and diagnosis.

3.2 PLS with Dynamic Latent Variables

The PLS algorithm performs regression with inter-related variables by projecting to a lower dimensional latent space one dimension at a time. This resembles a version of conjugate gradient methods for the linear regression problem. This approach not only avoids direct inversion of a potentially ill-conditioned matrix in ordinary least squares, it also provides a way to tradeoff between the model prediction variance and bias by selecting an appropriate number of latent variables less than the number of input variables.

The objective of PLS only focuses on static relations in the input and output data. In the case that dynamic relationships exist between the input and output data, PLS will leave the dynamics unmodeled. To build a dynamic PLS model, the objective should be changed to aim at extracting a dynamic latent relation as follows,

$$u_k = \beta_0 t_k + \beta_1 t_{k-1} + \cdots + \beta_s t_{k-s} + r_k$$

with the latent variables related to original variables as follows

$$u_k = \mathbf{y}_k^T \mathbf{q}$$

$$t_k = \mathbf{x}_k^T \mathbf{w}$$

For each factor, the inner model prediction should be

$$\begin{aligned}\hat{u}_k &= \mathbf{x}_k^T\mathbf{w}\beta_0 + \mathbf{x}_{k-1}^T\mathbf{w}\beta_1 + \cdots + \mathbf{x}_{k-s}^T\mathbf{w}\beta_s \\ &= [\mathbf{x}_k^T \quad \mathbf{x}_{k-1}^T \cdots \mathbf{x}_{k-s}^T](\boldsymbol{\beta} \otimes \mathbf{w})\end{aligned}$$

The dynamic inner PLS (DiPLS) algorithm from Dong and Qin (2015) maximizes the covariance between the latent scores u_k and its prediction as follows,

$$\frac{1}{N}\sum_{k=s}^{N+s} \mathbf{q}^T\mathbf{y}_k[\mathbf{x}_k^T \quad \mathbf{x}_{k-1}^T \cdots \mathbf{x}_{k-s}^T](\boldsymbol{\beta} \otimes \mathbf{w})$$

This objective contains clearly latent dynamics, while remaining outer projections of the input and output data to the latent variable dimension. For the special case of s = 0, DiPLS reduces to the static PLS.

For a number of observations of input and output data we form the following data matrices

$$\mathbf{X}_i = [\mathbf{x}_i \quad \mathbf{x}_{i+1} \quad \cdots \quad \mathbf{x}_{i+N}]^T, \text{for } i = 0, 1, 2, \cdots, s$$

$$\mathbf{Z}_s = [\mathbf{X}_s \quad \mathbf{X}_{s-1} \quad \cdots \mathbf{X}_0]$$

$$\mathbf{Y}_s = [\mathbf{y}_s \quad \mathbf{y}_{s+1} \quad \cdots \quad \mathbf{y}_{s+N}]^T$$

The objective of DiPLS can be represented as

$$\begin{aligned}&\max \quad \mathbf{q}^T\mathbf{Y}_s^T\mathbf{Z}_s(\boldsymbol{\beta} \otimes \mathbf{w}) \\ &\text{s.t.} \quad \|\mathbf{w}\| = 1, \|\mathbf{q}\| = 1, \|\boldsymbol{\beta}\| = 1\end{aligned}$$

Lagrange multipliers are used to solve this optimization problem, which yields the DiPLS algorithm (Dong and Qin, 2015) as given in Appendix B.

3.3 DiCCA with Dynamic Latent Variables

The DiPCA and DiPLS algorithms build inherent dynamics in the latent variables and give explicit projections from the data space to the latent space. However, the objective functions that maximize the covariance do not necessarily lead to a principal time series that can be best predicted by its past values. To obtain a principal time series that can be best predicted from its past values, some form of least squares objective should be minimized, such as,

$$\min \quad J = ||\mathbf{u} - b\mathbf{t}||^2 = ||\mathbf{Yq} - b\mathbf{Xw}||^2$$

This objective, of course, does not have a minimum unless the weights are somehow restricted in the norm. By restricting $||\mathbf{Yq}||^2 = 1$ and $||\mathbf{Xw}||^2 = 1$, we have the following Theorem.

[Theorem 1] The least squares objective, min $\quad J = ||\mathbf{Yq} - b\mathbf{Xw}||^2$ reduces to the CCA objective, max $\quad J = \frac{\mathbf{q}^T\mathbf{Y}^T\mathbf{Xw}}{||\mathbf{Yq}||||\mathbf{Xw}||}$, if $||\mathbf{Yq}||^2 = 1$ and $||\mathbf{Xw}||^2 = 1$.

The proof of the theorem is straightforward by using Lagrange multipliers, which is omitted here. Therefore, to achieve a truly most predictive time series from the past data of itself or another latent variable, DiPCA and DiPLS should use the CCA objective that maximizes the correlation instead of the covariance. This modification leads to a dynamic inner CCA (DiCCA) algorithm, which simply replaces the covariance objective in DiPCA and DiPLS with a correlation objective. between The principal time series is best predicted from the past values of itself or a latent variable derived from another set of variables. It is straightforward to solve the maximization problem by using Lagrange multipliers to derive the DiCCA algorithm, which applies to both dynamic PCA and dynamic PLS problems.

4 Case Demonstration Using Real Process Data

Figure 3 shows a process schematic diagram from the Eastman Chemical Company, USA. Eastman Chemical has identified a need to diagnose a common oscillation with a period around two hours (320 samples/cycle). Five process variables are selected that have strong oscillations (Yuan and Qin, 2014), which are used here to demonstrate how the dynamic data and features can be modeled using DiPCA and DiCCA and compared to PCA.

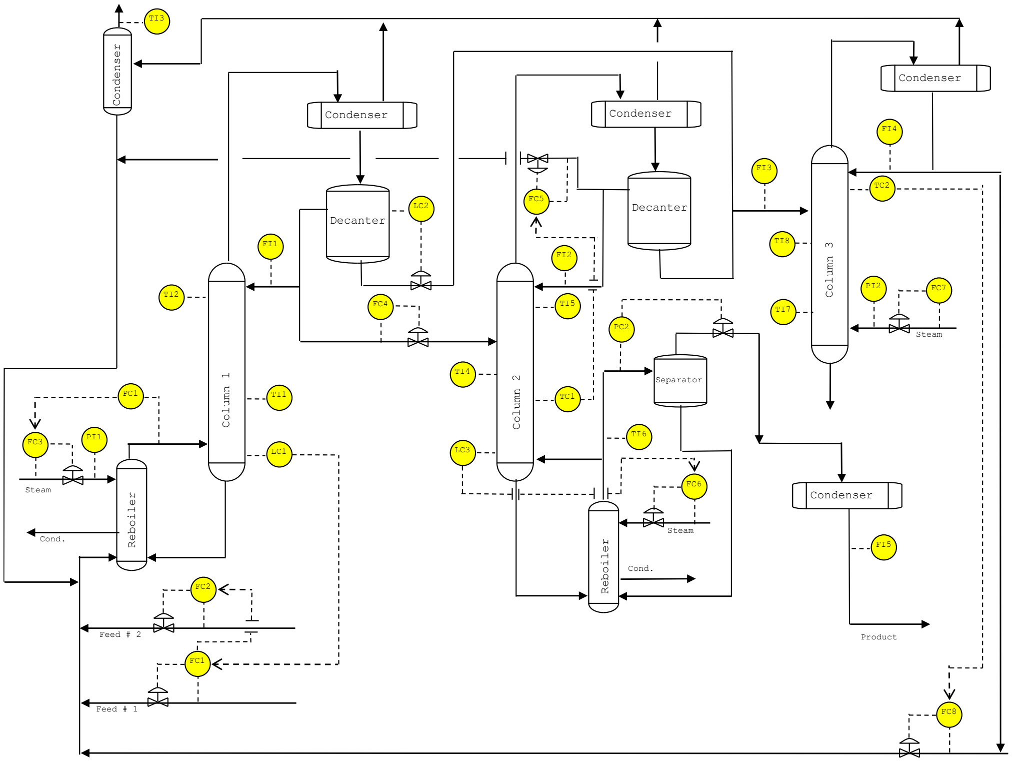


Figure 3. Process schematic diagram from the Eastman Chemical Company

4.1 DiPCA Results

Using DiPCA on the five process variables leads to five dynamic PCs as shown in Figure 4. The auto-regression order of the dynamics is chosen to be 21, which makes the prediction errors of the dynamic principal components essentially white. Figure 5 depicts the auto-correlation and cross-autocorrelation for the five dynamic PCs. It is clear that the first two PCs are very oscillatory, while the third one is still somewhat oscillatory and co-varies with the first two PCs. To visualize how the DiPCA model predicts the PCs, the first two DiPCA PCs and the predictions from their past scores are shown in Figure 6. The circular shape shows the co-varying oscillations at the same frequency.

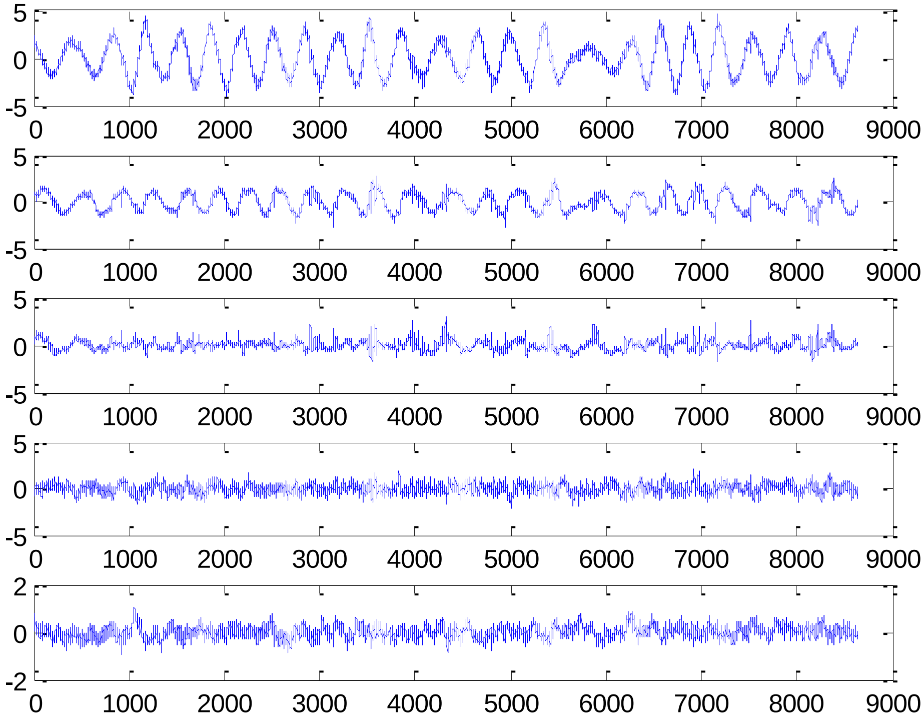


Figure 4. Plots of five dynamic principal components using DiPCA

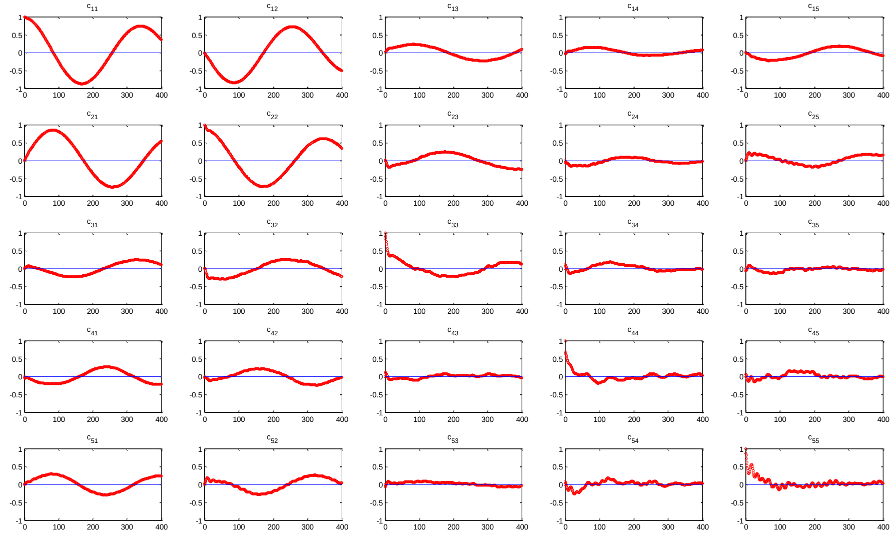


Figure 5. Auto-correlation and cross-autocorrelation for five DiPCA PCs.

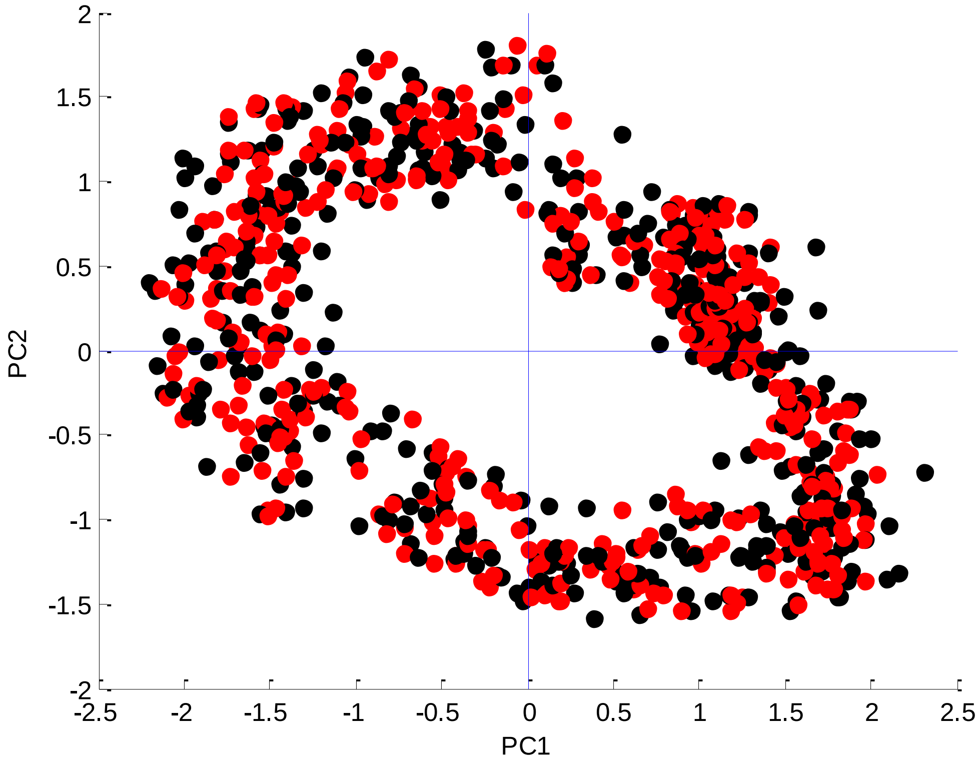


Figure 6. The first two DiPCA PCs and the predictions from their past scores using DiPCA. The circular shape shows covarying oscillations at the same frequency.

4.2 DiCCA Results

Next, DiCCA is used to model the five process variables, which leads to five dynamic PCs as shown in Figure 7. The order of the dynamics is chosen as 22, which is chosen such that the errors predicted with the dynamic PCs are essentially white. Figure 8 depicts the auto-correlation and cross-autocorrelation for the five DiCCA PCs. It is clear that the first two PCs are very oscillatory, while the third one is little correlated to the first two PCs.

To visualize how the DiCCA model predicts the PCs, the first two DiCCA PCs and the predictions from their past scores are shown in Figure 9. While the big circular shape shows co-varying oscillations at the same frequency, there is a clear smaller oscillation with higher frequency that is best captured by the second PC. This feature is **not** observed at all using DiPCA analysis. The DiCCA scatterplot has clear ups and downs on top of the circular shape, indicating that there is another high frequency oscillation component. This frequency is more likely caused by the valve stiction, since the bigger oscillation of 320 points per cycle (about two hours) seems to be too large a period to be caused by a valve stiction. The fact that DiCCA detects a new feature makes it better than DiPCA in extracting dynamic features.

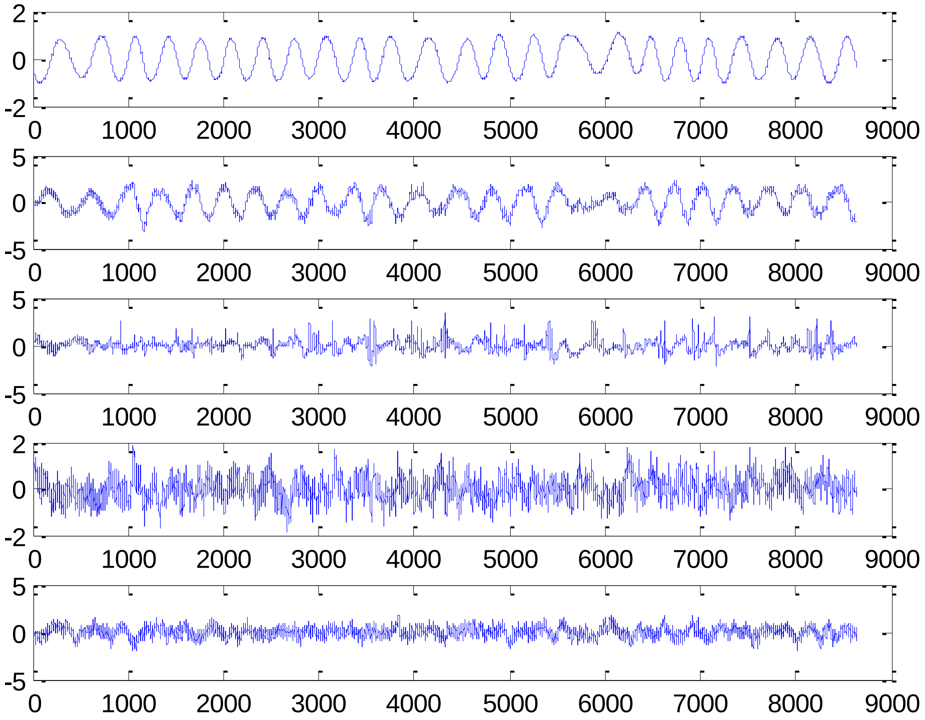


Figure 7. Plots of five dynamic principal components using DiCCA

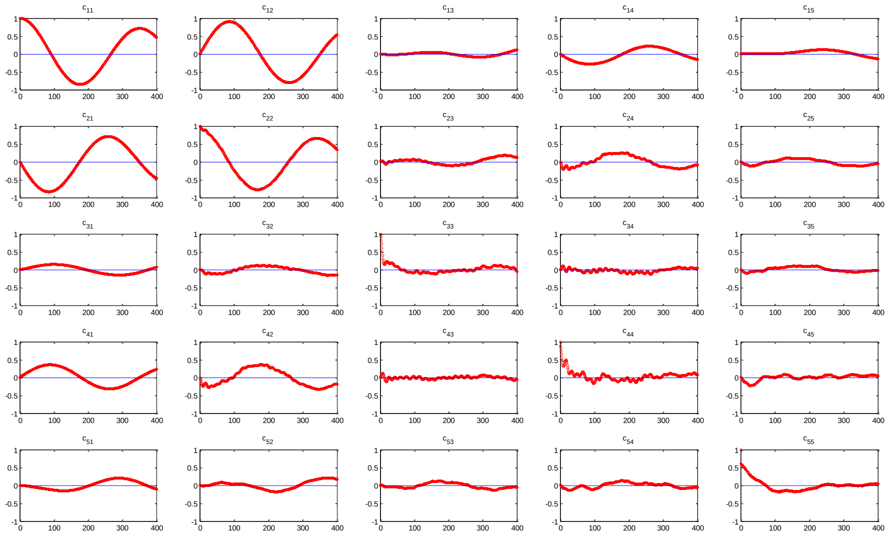


Figure 8. Auto-correlation and cross-autocorrelation for five DiCCA PCs. The third PC is little correlated to the first two PCs

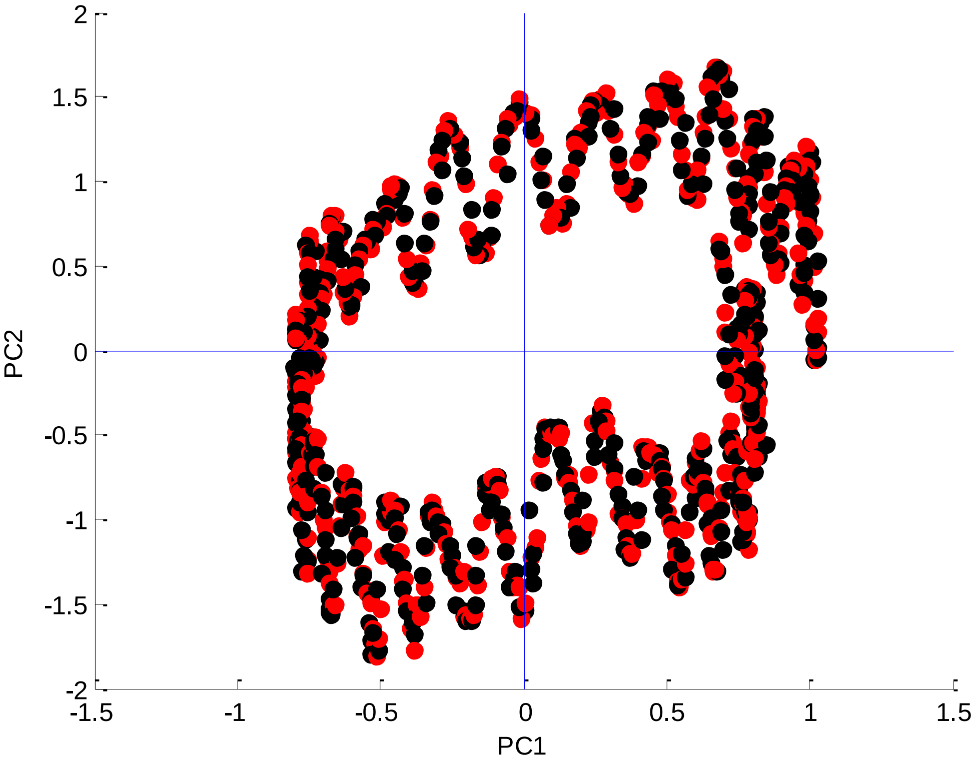


Figure 9. The first two DiCCA PCs and the predictions from their past scores. The big circular shape shows co-varying oscillations at the same frequency. In addition, there is a smaller oscillation with higher frequency due to the second PC.

4.3 Comparison of DiPCA, DiCCA and PCA Results

To illustrate the efficiency and effectiveness of DiPCA and DiCCA in extracting dynamics in the data, we compare their models to the results from standard PCA. Figure 10 depicts the predicted R^2 values of each LV of the DiPCA and DiCCA models in the top row, while the percent variances captured by each LV of DiPCA, DiCCA, and PCA are shown in the bottom row. PCA is not a predictive model so it does not have predicted R^2 values to show. As can be seen, the predicted R^2 values for the first two LVs of DiPCA and DiCCA model are very close to one, showing that the periodic latent variables are nearly perfectly predicted by their past values. The predicted R^2 values for the third to fifth LVs of DiPCA and DiCCA are different; the predicted R^2 values from DiCCA model are higher than those from the DiPCA model. Furthermore, the predicted R^2 values from DiCCA have clearly a descending order, while those from DiPCA do not. The results are the natural outcome of the DiCCA objective to minimize the prediction errors in a least squares sense.

The percent variances captured by each LV of DiPCA, DiCCA, and PCA are also different. The DiCCA model ranks the LVs in the order of descending predictability from their past values, as shown by the predicted R^2 values, and it does not rank the LVs by the percent variance captured. The PCA focuses on maximizing the variance captured each LV only and has no attention to predicting its values using the past. In this case study, due to the high predicted R^2 values of the first two DiPCA LVs, the t_k and its prediction $\hat{t}_k$ are nearly identical, making the DiPCA objective close to the PCA objective for these LVs. Therefore, the percent variance captured by these two LVs of DiPCA and PCA are similar. However, this is *not* generally true, and the two models are intrinsically different.

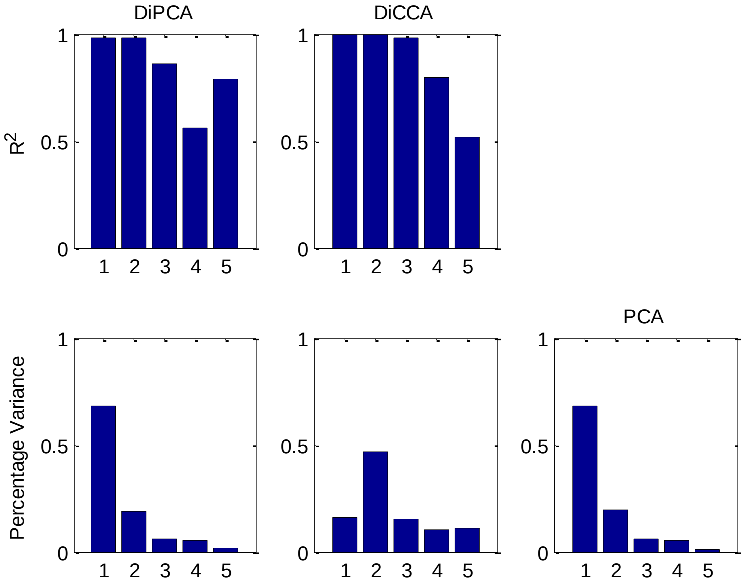


Figure 10. The predicted R^2 values of the DiPCA and DiCCA models and the percent variances captured by DiPCA, DiCCA, and PCA vs. the number of LVs.

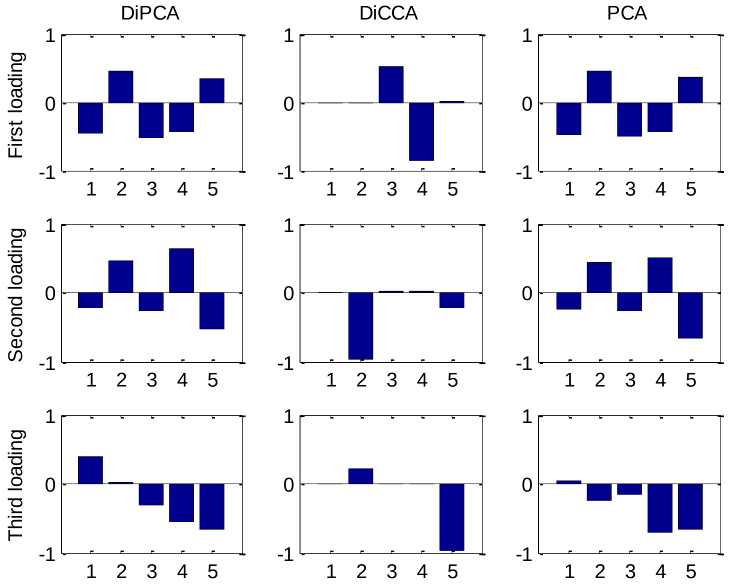


Figure 11. The loadings of the DiPCA, DiCCA, and PCA models vs. the LVs.

To further examine the difference among the DiPCA, DiCCA, and PCA models, their loadings vs. the LVs are shown in Figure 11. The DiCCA loadings clearly point out that Variables 4 and 3 dominate the first LV, while Variables 2 and 5 dominate the second LV, which has superimposed higher frequency oscillations.

The first two loadings from the DiCCA and PCA models are similar in this specific example, but they do not give the selectivity of variables as DiCCA does.

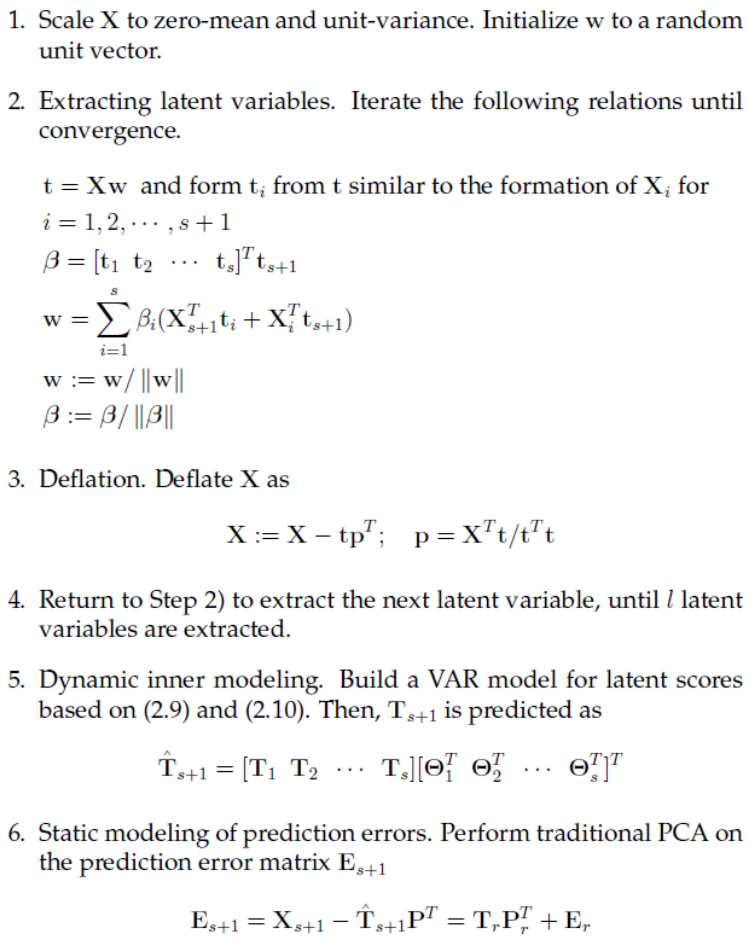
The fact that DiCCA and DiPCA has explicit loadings specific to each variable is another advantage of these methods over the traditional dynamic PCA method that simply augments a time-lagged matrix from the original data matrix (Ku et al., 1995). This feature makes the proposed new methods superior in interpretability.

5 Summary

Process data analytics have been applied in chemical process operations for decades. However, with the development of advanced analytics in other sectors of industries and business operations, there appears to be much more room to grow. While physical and chemical sciences develop principles based on which mechanistic models are established for process understanding, data analytics provide real and up-to-date information that reflects changes in the operation, and provide a reliable source of information to characterize uncertainty and diagnose emerging situations.

Prediction, visualization, and interpretation using latent variables are powerful to deal with massive, high dimensional and highly correlated data. The goal of data analytics is to turn data into knowledge and support effective decision-making. Nonlinear and robust methods in statistical machine learning are new ways to use messy and complex data, which goes beyond a traditional mindset. To make best use of machine learning to extract knowledge from massive data, the practitioners should familiarize themselves with data science tools (e.g., Hinton, G. E. and R. Salakhutdinov, 2006; Jordan, M.I. et al., 2013; Keogh, E., and S. Kasetty, 2002) and merge them into existing tools that are proven effective.

Appendix A. DiPCA Algorithm.



1. Scale X to zero-mean and unit-variance. Initialize w to a random unit vector.

2. Extracting latent variables. Iterate the following relations until convergence.

   $\mathrm{t} = \mathrm{X}\mathrm{w}$ and form t_i from t similar to the formation of X_i for $i = 1, 2, \cdots, s+1$

   $\beta = [\mathrm{t}_1 \;\; \mathrm{t}_2 \;\; \cdots \;\; \mathrm{t}_s]^T \mathrm{t}_{s+1}$

   $\mathrm{w} = \sum_{i=1}^{s} \beta_i (\mathrm{X}_{s+1}^T \mathrm{t}_i + \mathrm{X}_i^T \mathrm{t}_{s+1})$

   $\mathrm{w} := \mathrm{w} / \|\mathrm{w}\|$

   $\beta := \beta / \|\beta\|$

3. Deflation. Deflate X as

$$\mathrm{X} := \mathrm{X} - \mathrm{t}\mathrm{p}^T; \quad \mathrm{p} = \mathrm{X}^T\mathrm{t}/\mathrm{t}^T\mathrm{t}$$

4. Return to Step 2) to extract the next latent variable, until l latent variables are extracted.

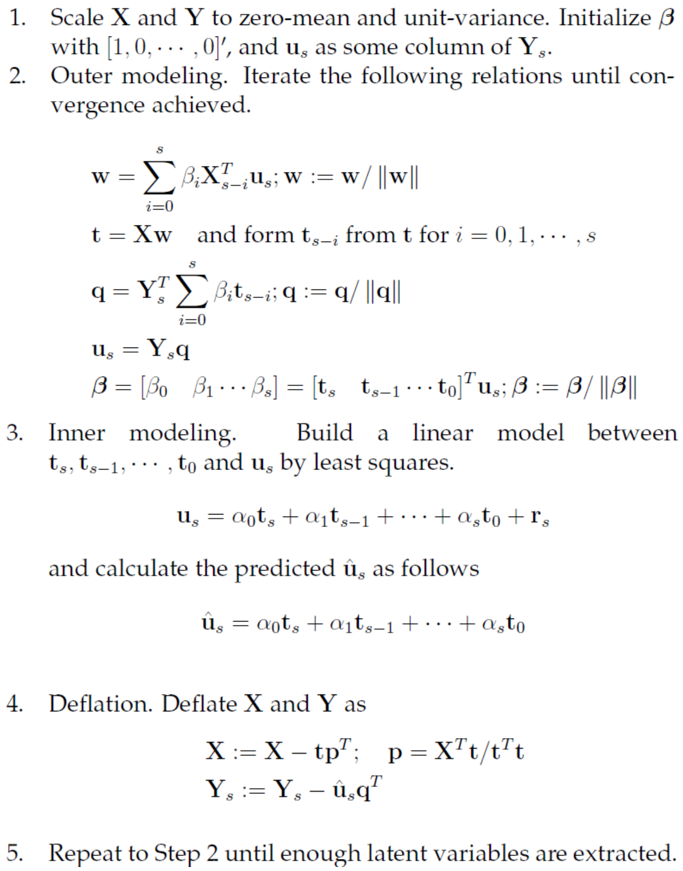
5. Dynamic inner modeling. Build a VAR model for latent scores based on (2.9) and (2.10). Then, T_{s+1} is predicted as

$$\hat{\mathrm{T}}_{s+1} = [\mathrm{T}_1 \;\; \mathrm{T}_2 \;\; \cdots \;\; \mathrm{T}_s][\Theta_1^T \;\; \Theta_2^T \;\; \cdots \;\; \Theta_s^T]^T$$

6. Static modeling of prediction errors. Perform traditional PCA on the prediction error matrix E_{s+1}

$$\mathrm{E}_{s+1} = \mathrm{X}_{s+1} - \hat{\mathrm{T}}_{s+1}\mathrm{P}^T = \mathrm{T}_r\mathrm{P}_r^T + \mathrm{E}_r$$

Appendix B. DiPLS Algorithm.



1. Scale $\mathbf{X}$ and $\mathbf{Y}$ to zero-mean and unit-variance. Initialize $\boldsymbol{\beta}$ with $[1, 0, \cdots, 0]'$, and $\mathbf{u}_s$ as some column of $\mathbf{Y}_s$.
2. Outer modeling. Iterate the following relations until convergence achieved.

$$\begin{aligned}
&\mathbf{w} = \sum_{i=0}^{s} \beta_i \mathbf{X}_{s-i}^T \mathbf{u}_s; \, \mathbf{w} := \mathbf{w} / \left\| \mathbf{w} \right\| \\
&\mathbf{t} = \mathbf{X}\mathbf{w} \quad \text{and form } \mathbf{t}_{s-i} \text{ from } \mathbf{t} \text{ for } i = 0, 1, \cdots, s \\
&\mathbf{q} = \mathbf{Y}_s^T \sum_{i=0}^{s} \beta_i \mathbf{t}_{s-i}; \, \mathbf{q} := \mathbf{q} / \left\| \mathbf{q} \right\| \\
&\mathbf{u}_s = \mathbf{Y}_s \mathbf{q} \\
&\boldsymbol{\beta} = [\beta_0 \quad \beta_1 \cdots \beta_s] = [\mathbf{t}_s \quad \mathbf{t}_{s-1} \cdots \mathbf{t}_0]^T \mathbf{u}_s; \, \boldsymbol{\beta} := \boldsymbol{\beta} / \left\| \boldsymbol{\beta} \right\|
\end{aligned}$$

3. Inner modeling. Build a linear model between $\mathbf{t}_s, \mathbf{t}_{s-1}, \cdots, \mathbf{t}_0$ and $\mathbf{u}_s$ by least squares.

$$\mathbf{u}_s = \alpha_0 \mathbf{t}_s + \alpha_1 \mathbf{t}_{s-1} + \cdots + \alpha_s \mathbf{t}_0 + \mathbf{r}_s$$

and calculate the predicted $\hat{\mathbf{u}}_s$ as follows

$$\hat{\mathbf{u}}_s = \alpha_0 \mathbf{t}_s + \alpha_1 \mathbf{t}_{s-1} + \cdots + \alpha_s \mathbf{t}_0$$

4. Deflation. Deflate $\mathbf{X}$ and $\mathbf{Y}$ as

$$\begin{aligned}
&\mathbf{X} := \mathbf{X} - \mathbf{t}\mathbf{p}^T; \quad \mathbf{p} = \mathbf{X}^T \mathbf{t} / \mathbf{t}^T \mathbf{t} \\
&\mathbf{Y}_s := \mathbf{Y}_s - \hat{\mathbf{u}}_s \mathbf{q}^T
\end{aligned}$$

5. Repeat to Step 2 until enough latent variables are extracted.

Literature Cited

Dong, Y., & Qin, S. J. (2015). "Dynamic-Inner Partial Least Squares for Dynamic Data Modeling," ***IFAC-PapersOnLine,*** **48**(8), 117-122.

Dong, Yining, and S. Joe Qin (2016). A Novel Dynamic PCA Algorithm for Dynamic Data Modeling and Process Monitoring. Submitted to ***Journal of Process Control***.

Ginsberg, J., M.H. Mohebbi, R.S. Patel, and L. Brammer L, "Smolinski1 MS, Brilliant L. (2009). Detecting Influenza Epidemics Using Search Engine Query Data," ***Nature***, 457, 1012-1014.

Hinton, G. E. and R. Salakhutdinov (2006). "Reducing the Dimensionality of Data with Neural Networks," ***Science***, **313**(5786), 504–507.

Jordan, M.I. et al. (2013). "Frontiers in Massive Data Analysis," The National Academies Press.

Keogh, E., and S. Kasetty (2002)., "On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration," ***Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining,*** 102-111.

Ku, W., R. H. Storer, C. Georgakis (1995). Disturbance detection and isolation by dynamic principal component analysis, ***Chemometrics and intelligent laboratory systems***, 30 (1) 179-196.

MacGregor J. F., M. Koutoudi (1995), Statistical process control of multivariate processes, Control Engineering Practice 3, 403-414.

Qin, S. J. (2003), Statistical process monitoring: Basics and beyond, J. Chemometrics, 17, 480-502

Qin, S. J. and T.J. McAvoy (1992), "A Data-Based Process Modeling Approach and Its Applications, ***Proceedings of the 3rd IFAC DYCORD Symposium***, 321-326, College Park, Maryland.

Qin, S. J., Y.Y. Zheng (2013). "Quality-Relevant and Process-Relevant Fault Monitoring with Concurrent Projection to Latent Structures," ***AIChE Journal***, **59**, 496-504.

Sun, L., Shuiwang Ji, Shipeng Yu, Jieping Ye (2009). On the Equivalence between Canonical Correlation Analysis and Orthonormalized Partial Least Squares, Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09).

Tham, M. (1991). "Soft-Sensors for Process Estimation and Inferential Control," ***Journal of Process Control***, **1**(1), 3-14. DOI:10.1016/0959-1524(91)87002-F.

Wise, B. and N. Gallagher (1996). The process chemometrics approach to process monitoring and fault detection, J. Proc. Cont. 6, 329-348.

Yuan, T., and S.J. Qin (2014). "Root Cause Diagnosis of Plant-wide Oscillations Using Granger Causality," ***Journal of Process Control***, 24, Pages 450–459.

Zhu, Q., Q. Liu, and S. J. Qin (2016). Concurrent canonical correlation analysis modeling for quality-relevant monitoring, Proc. of IFAC Symposium on ***DYCOPS-CAB***, IFAC-PapersOnLine, vol. 49, no. 7, pp. 1044–1049.