CH 5440 Multivariate Data Analysis in Process Monitoring and Diagnosis

Assignment 5

1. Consider the problem of developing a correlation between pressure and saturated temperature (boiling point). For pure components, the Antoine equation given below generally fits the data well

$$\ln P^{sat} = A - \frac{B}{T + C}$$

For n-hexane, the values of the constants are A = 14.0568, B = 2825.42, and C = 230.44 where P^{sat} is given in kPa and T in deg C. Using this correlation a data set consisting of 100 samples have been generated in the temperature range 10 - 70 deg C. Gaussian measurements errors to both the true temperature and saturated pressures with standard deviations of 0.18 deg C and 2 kPa, respectively, have been added to generate the measurements (available in *vpdata.mat*)

(a) Apply Kernel PCA to obtain a nonlinear correlation between saturated pressure and temperature using Gaussian Kernels. Use the first 70 samples for developing the model and the remaining 30 samples for cross validation. Choose the optimal hyper-parameters (Kernel width as well as the number of PC's in feature space to be chosen) using PRESS on the cross validation samples. It is better to first shift the temperature measurements using the mean and scale using the maximum range or the standard deviation of the temperature before applying the nonlinear transformation. Note that you should shift and scale the temperature as you do for the training set before using the KPCR model for predicting the saturated pressures for the test data set.

(b) Test the accuracy of predicting the saturated pressures using the above KPCA model for a temperature of 55 deg C and temperature of 100 deg C and compare with the values obtained using the Antoine equation. What are your observations from the predicted saturated pressures? Provide reasons for the same.

Note: Matlab function eig for computing eigenvalues and eigenvectors of a square matrix arranges the eigenvalues from smallest to largest (unlike svd)

2. An autoregressive dynamic model of order *p* with exogenous inputs (ARX) for a single-input single-output (SISO) process is given by

$$y_k = a_1 y_{k-1} + a_2 y_{k-2} + \mathrm{K} + a_p y_{k-p} + b u_{k-1}$$

where y_k is the measured value of output at time instant *k* and u_k is the measured value of manipulated input at time *k* and a_i, *b* are the unknown coefficients of the ARX model. The data file arx.mat contains 1000 measurements of input (variable umeas) and output (variable ymeas) made at 1000 successive time instants of a dynamic process.

(a) Assuming a first order ARX model to describe the dynamic process, apply OLS to estimate the ARX model parameters. For this purpose, construct an input data matrix from the time series data consisting of $[y_{k-1} \;\; u_{k-1}]$ corresponding to the output $[y_k]$ at time k. Thus, an input matrix of size 999x2 and a corresponding output data matrix of size 999x1 can be constructed from the given time series data. Report the parameters of the ARX model obtained.

(b) Since both inputs and output measurements contain errors, it is better to apply TLS to obtain the ARX model parameters. (i) Rewrite Eq. (2) for a first order ARX process as an implicit equation relating the variables y_k, y_{k-1} and u_{k-1} and estimate the constraint by applying PCA to the data matrix containing both inputs and outputs (*Hint: use the input and output data matrices constructed in part (a)*). Assume that the errors in the input and output variables have same variances and, therefore, there is no need to scale the data before applying PCA. (ii) Derive the ARX model from the constraint model estimated using PCA and report the ARX model parameters.

(c) In part *(b)* it is implicitly assumed that we know the true order of the dynamic process and we have therefore appropriately constructed the data matrix to ensure that there will be only one constraint among the variables of the data matrix. In general the true order is unknown and therefore we stack several past instances of the output and input variable in the data matrix. That is the data matrix may contain for every instant k a stacked vector corresponding to $\begin{bmatrix} y_k & y_{k-1} & \mathrm{K} & y_{k-m} & u_{k-1} & u_{k-2} & \mathrm{K} & u_{k-m} \end{bmatrix}$. (i) If the true process order is 1, how many constraints are present among the stacked vector of variables corresponding to a stacking order m? (ii) If the true process order is p derive a relation between number of constraints, ARX model order p and stacking order m (assuming m is greater than p).

(d) Apply PCA to a data matrix obtained using stacking order 10 and obtain an estimate of the model order. Obtain a stacked data matrix using stacking order equal to model order and use last PCA to obtain an estimate of the dynamic model

(e) If the errors in input and output variables are different and unknown, describe a procedure for using IPCA to simultaneously estimate error variances and model order and model parameter estimates. Describe the additional modifications that you need to make to IPCA to estimate noise variances and model order. What is the minimum stacking order required so that IPCA can be used to estimate noise variances?

3. Apply Sparse PCA to the microarray data set given in problem 2 of assignment 4. For this purpose, use the sparse PCA matlab function provided (the code was developed by Matthias Hein and Thomas Bühler of Department of Mathematics and Computer Science - Saarland University, Denmark). Count the number of zeros in each column of the structure matrix and provide this as input to the sparse PCA code to determine the sparse loadings vector. (Note that sparse PCA will not produce a loadings vector with the position of the zeros as given in the structure matrix). Assess the performance of sparse PCA by computing the fraction of non-zero positions correctly estimated by sparse PCA for each column of the structure matrix.

4. The data audiomixture.mat contains five noisy mixtures created by mixing of two sources (an excerpt from Kennedy's famous speech and an excerpt from a popular old song) and adding

Gaussian noise. The original sources are also included as wav files which can be read by MATLAB using function audioread (function sound can be used to play the clip after reading it). Apply ICA to separate the source signals. Compare how well the source signals have been extracted based on correlation between extracted and original signals. (Note like NMF, ICA also has a permutation ambiguity). For this purpose use the FASTICA toolbox developed at Helsinki University of Technology.