Brazilian Journal
of Chemical
Engineering

ISSN 0104-6632
Printed in Brazil
www.abeq.org.br/bjche

Vol. 36, No. 01, pp. 409 - 419, January - March, 2019
dx.doi.org/10.1590/0104-6632.20190361s20170455

K-RANK: AN EVOLUTION OF Y-RANK FOR MULTIPLE SOLUTIONS PROBLEM

Pedro V. J. L. Santos[1], Lucas Ranzan[1*], Marcelo Farenzena[1] and Jorge O. Trierweiler[1]

[1] Universidade Federal do Rio Grande do Sul, Departamento de Engenharia Química, Grupo de Intensificação, Modelagem, Simulação, Controle e Otimização de Processos, Porto Alegre, RS, Brazil. E-mail: lcsranzan@gmail.com, ORCID: 0000-0002-9407-4320

(Submitted: August 25, 2017 ; Revised: August 2, 2018 ; Accepted: August 6, 2018)

Abstract - Y-rank can present faults when dealing with non-linear problems. A methodology is proposed to improve the selection of data in situations where y-rank is fragile. The proposed alternative, called k-rank, consists of splitting the data set into clusters using the k-means algorithm, and then apply y-rank to the generated clusters. Models were calibrated and tested with subsets split by y-rank and k-rank. For the Heating Tank case study, in 59% of the simulations, models calibrated with k-rank subsets achieved better results. For the Propylene / Propane Separation Unit case, when dealing with a small number of sample points, the y-rank models had errors almost three times higher than the k-rank models for the test subset, meaning that the fitted model could not deal properly with new unseen data. The proposed methodology was successful in splitting the data, especially in cases with a limited amount of samples.
Keywords: Splitting data; K-means; Systematic sampling; Multiple solutions.

INTRODUCTION

The explosion of data is a reality in all scientific areas. In chemical engineering, oil processing plants (Chandra Srivastava, 2012; Baliño, 2014), chemometrics studies (Ranzan et al., 2014), and process control strategies (Storkaas and Skogestad, 2007; Chi et al., 2014; Boullosa et al., 2017) can accumulate so much raw data that it is difficult to analyze it all to extract useful information. Learning from data is part of important fields in computer science called machine learning and data mining. There are many applications for these techniques, such as in bioinformatics, where large genome datasets need to be analyzed for detecting diseases and for new drug development, or in economics, where the analysis of large market datasets can help improve planning and decision-making strategies (Kramer, 2016; Massaron and Boschetti, 2016).

Learning from data means that new knowledge is extracted from a large amount of information. Often, large datasets cannot be studied directly in their raw form, so several techniques of data mining and machine learning have been proposed and developed in the last decades. Predictive models and inferences can be developed through supervised learning, which uses information from the system response to calibrate a predictive function that will estimate a given variable. Data can also be grouped and qualitatively analyzed using unsupervised techniques, that is, without system output information (Raschka, 2015; Kramer, 2016; Massaron and Boschetti, 2016).

In a prediction or inference problem, we search for a model $y_i = f(x_i)$ that could represent an estimate of a real, but unknown, model. Therefore, machine learning techniques are used to develop the model $f(x_i)$ from observed data, where y_i is the desired value. The adjusted model has parameters that can be tuned during a training process (Kramer, 2016).

To fit the model, the data can be split into training, validation and test sets. The training set should contain enough information for the model to achieve a good fit that could represent the whole system satisfactorily. The validation set tests the model and is used to make

* Corresponding author: Lucas Ranzan - E-mail: lcsranzan@gmail.com

update decisions. With the test set, the reliability of the definitive model is evaluated with new data (not used during training). One of the problems that can occur during modeling is called overfitting, which means that a model fits well the training data, but fails to achieve the same accuracy in an independent test dataset. However, a good model should be able to generalize data that was not used in its training (Kramer, 2016).

BACKGROUNDS: Y-RANK AND K-MEANS

A simple method for splitting data is the systematic sampling approach (Kao et al., 2011). This procedure is also known as 'Y-rank', and one simple implementation is described in a MatLab toolbox developed by FABI (1997). The y-rank algorithm ranks the data based on the outputs values *y*, followed by selecting every *k*th sample of a population of *N* units. From a predetermined splitting ratio, the algorithm selects the pace between *k*s (the selected samples) and distributes the data into training/validation/test subsets (Fleck et al., 2012; Schultz, 2015). The work of Leu and Kao (2006) presents a review on many modifications of the systematic methodology, such as multi-start sampling (Gautschi, 1957), Markov systematic sampling (Sampath and Uthayakumaran, 1999) and circular systematic sampling. All the modification try to solve problems in the split, especially in populations where the simple y-rank cannot capture linear or parabolic trends efficiently.

Other methodologies for data splitting such as cross-validation (Browne, 2000) and Leave-P-out (Celisse and Robin, 2008) can be very useful for some cases, providing randomness to the selection and usually yielding more robust models. However, the use of this kind of methodology that demands training in various groups greatly increases the computational time, being sometimes prohibitive.

In the implementation of the y-rank algorithm used in this work, the data is sorted in ascending order of the system output *y*. Then, the proportions of each subset are determined (e.g., 50% for calibration and 50% for test), and y-rank adopts a pattern corresponding to the sampling of every *k*th sample, which is repeated over the whole dataset splitting it into subsets (i.e., calibration-calibration-test or calibration-validation-test and so on). It is worth mentioning that the extremes are fixed as calibration so that extrapolation does not occur (Fleck et al., 2012; Schultz, 2015). An example of the splitting can be seen in Figure 1, representing the first ten numbers of the Fibonacci sequence.

If *y* grows linearly as a function of the independent variables, this will extinguish a possible unequal separation of the set, in which each subset would have small and distinct intervals (Fleck et al., 2012). The problem with y-rank occurs when there are multiplicities

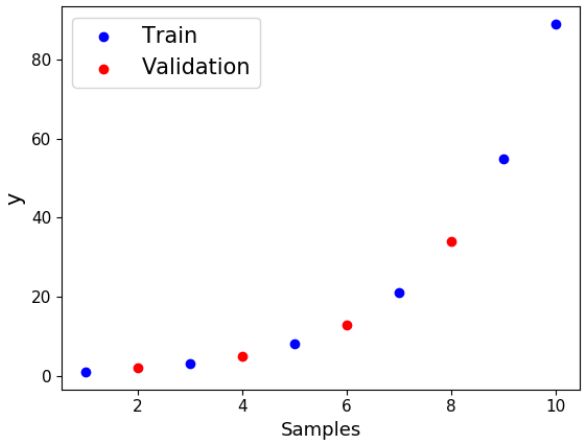

Figure 1. Example of the first 10 numbers of the Fibonacci sequence separated by y-rank.

of solutions, what can make the y-rank algorithm fail, because, for the same value of y, solutions may exist in different regions. To solve this problem, we propose first a division of the sample set into similar groups and then apply the y-rank algorithm. The grouping is performed using an unsupervised data mining technique known as k-means. The union of k-means and y-rank gives rise to a new methodology for data splitting, which we called 'k-rank' splitting method.

K-means is a popular clustering algorithm, which is categorized as unsupervised because it does not require information about the output variable. Clustering analysis can be understood as a technique that groups similar objects (Shamir et al., 2005; Thalamuthu et al., 2006; Raschka, 2015). The k-means algorithm aims to partition a set of data into *k* groups based on the similarity of the data.

The similarity between points is defined as the opposite of distance. The metric used by the algorithm is the square of the Euclidean distance between two points '*x*' and '*y*' m-dimensional. Equation 1 measures the multivariable Euclidian distance of 2 points (Raschka, 2015). The index *j* refers to the column of the dataset.

$$d(x,y)^2 = \sum_{j=1}^{m}(x_j - y_j)^2 = x - y_2^2 \quad (1)$$

Based on Euclidian distance, the k-means algorithm is understood as a simple optimization algorithm in which the minimization function is the sum of squared errors of points and centroids (Equation 2).

$$SSE = \sum_{i=1}^{n}\sum_{j=1}^{m} w^{(i,j)} x^{(i)} - ¼^{(j)}{}_2^2 \quad (2)$$

$\mu^{(j)}$ represents the centroid of cluster *j*, and if $w^{(i,\,j)} = 1$, the point $x^{(i)}$ belongs to the cluster. Otherwise $w^{(i,}$

$^{j)} = 0$. The k-means is an algorithm sensitive to the normalization of variables (Raschka, 2015).

It is worth mentioning that the number of groups must be informed to the algorithm; however, there are ways to evaluate the best number of clusters (Wang et al., 2009; Raschka, 2015). In this work, the silhouette analysis was adopted to evaluate the quality of the clusters generated by k-means. The analysis is done by calculating the silhouette coefficient, as follows (Raschka, 2015):

1. Calculate the cohesion of clusters as the average distance between the point and all other points in the same cluster.

2. Calculate the separation of clusters as the average distance between the point and all other points in the nearest cluster.

3. Calculate the silhouette coefficient as the difference between the cohesion and the separation of the clusters, divided by the maximum value of one of the two.

$$S^{(i)} = \frac{b^{(i)} - a^{(i)}}{\max\{b^{(i)} a^{(i)}\}} \quad (3)$$

The silhouette coefficient ranges between -1 and 1. If the coefficient is zero, the cohesion and the separation are equal, and the clusters will not be well separated, with clusters disordered and overlapping. However, when the coefficient approaches 1, the separation assumes a high value, while the cohesion assumes a value close to zero. In this case, clusters are well defined (Wang et al., 2009; Raschka, 2015).

Wang et al., (2009) suggested in his Cluster Validity Analysis Platform (Cluster Validation) that a higher value of the silhouette index indicates a better grouping of the data. In his work, he compared k-means, Hierarchical Clustering (HC), partitioning around medoids (PAM) and self-organizing maps (SOM) to group a set of yeast data. In their case study, the best method of separation was the k-means, chosen through the coefficient of silhouette, satisfactorily (Wang et al., 2009).

PROPOSED METHODOLOGY

The algorithm was developed in Python, mainly using the Scikit-learn package, an open source machine learning library that has a range of methods for classification, regression, covariance matrix estimation, dimensionality reduction, data preprocessing, among others (Kramer, 2016).

The proposed methodology can be visualized in the flowchart of Figure 2. The left-hand path explains the k-rank methodology, where we apply the y-rank to each cluster generated by k-means. It is important to remember that the implementation of y-rank used in this study makes a univariate decision on the splitting of the data: one output *y* must be selected as the main variable for the split. In addition, the selection of extremes will be made considering the values of this variable: both the highest and lowest values will always be selected as training, even inside each cluster. It then concatenates what has been allocated for training or testing in their specific sets. The right-hand path represents the normal y-rank approach, where the whole data is ranked and split based only on the output variable's values (*y*). In the end, we have training and tests sets of the same size for both paths, but with the selection of different data.

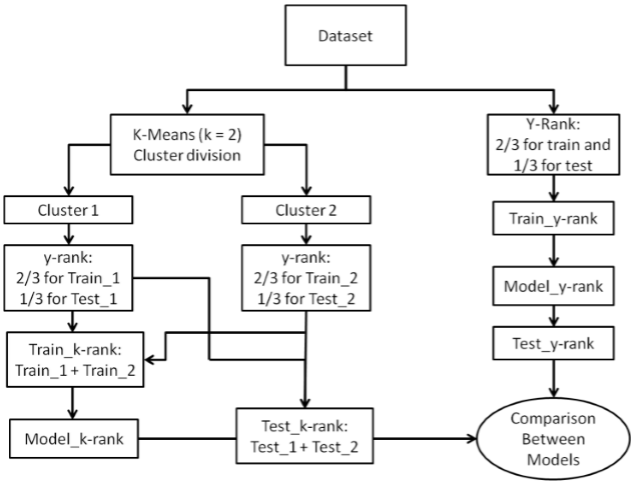

Figure 2. Simplified flowchart of the proposed methodology compared with the traditional y-rank approach.

For clustering, the number of clusters was calculated through the silhouette analysis and later used as input in the *KMeans* function, available in Scikit-learn. The next step was to train models using the y-rank data selection applied to the initial data set and the y-rank applied in the generated clusters (k-rank).

After training the models, they were compared, taking into account mainly their ability to estimate new data, since during training the linear regression forces the minimization of the adjusted curve error in relation to the training data used. Therefore, the greatest interest is in obtaining a model capable of generalizing new data. To compare the models, three standard metrics were used: MAPE (Mean Absolute Percentage Error), R^2 (coefficient of determination) and RMSE (Root Mean Squared Error) (Greene, 2002).

A second analysis was also made: the incidence of discrepant predicted points. When analyzing this incidence, we can discard models that do not fit well in any region of the curve, comparing predicted values with actual values in the test set. This analysis was done by analyzing the relative percentage error of each predicted point, according to Equation 4, where $y_{predicted}$

is the predicted output of the model, and y_{real} is the real output value:

$$e\% = \left| \frac{y_{predited} - y_{real}}{y_{real}} \right| \cdot 100\% \quad (4)$$

FIRST CASE STUDY: HEATING TANK

In this case study, simulation of a hypothetical heating system (Figure 3) is used as a source of non-linear data. Differential Equation 5 describes the system dynamics, where *T* is the temperature of the heating tank [°C], *Fin* the inlet flow [min^{-1}], *V* the tank volume [m^3], *Tin* the inlet flow temperature [°C], *Ta* the temperature of the heating fluid [°C] and *x* the valve opening. Stationary solutions for the tank temperature (*Tss)* can be obtained by Equation 6, where *Df* = *Fin* / *V*.

$$\frac{dT}{dt} = \frac{Fin}{V}(Tin - T) + \frac{10x}{1 + 20x^2}(Ta - T) \quad (5)$$

$$Tss = \frac{Df \cdot Tin + 20Df \cdot Tin \cdot x^2 + 10 \cdot x \cdot Ta}{Df + 20Df \cdot x^2 + 10x} \quad (6)$$

In this work, we assume *Df* = 2 min^{-1}, *Tin* = 10 ºC, *Ta* = 80 ºC. These values were selected for being commonly used in heating tank examples. The solutions for the stationary temperature (*Tss*) of the tank as a function of the valve opening can be visualized in Figure 4.

From Figure 4, it can be seen that, for some values of *Tss*, there may be two distinct values of *x*, and these values belong to different regions of operation. The goal of this work is to apply y-rank after the k-means algorithm identifies these distinct operation regions, minimizing the possibility of the y-rank to be unfair to a region.

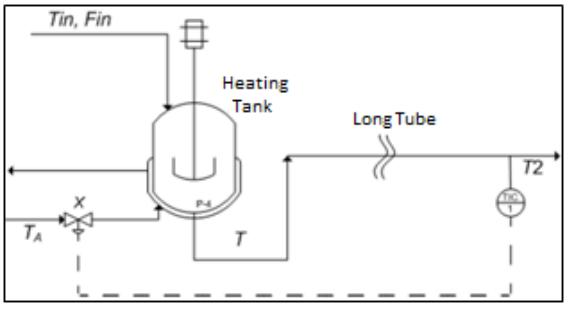

Figure 3. Process diagram of the heating tank case study.

The methodology applied to the Heating Tank Case Study

For the simulations of the heating tank, 21 random values were generated between 0 and 1 for the variable *x* and, based on Equation 6, 21 values were calculated for the tank temperature *T*. Subsequently, the temperature was normalized dividing the values by its maximum. The generated data set was a matrix with dimensions (21, 2), where the first column represents the opening of the valve '*x*' and the second column represents the normalized temperature of the tank '*T*'.

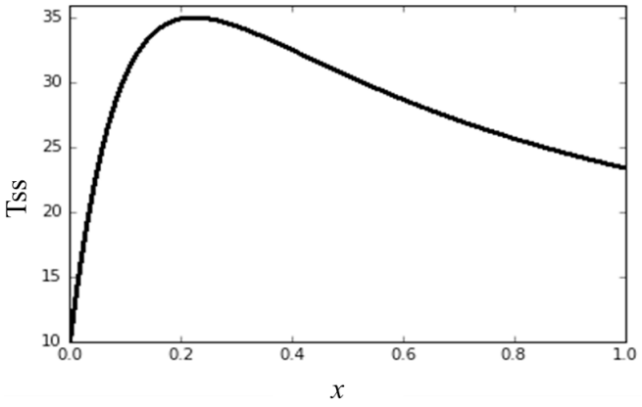

Figure 4. Stationary solutions for the heating tank system with *Df* = 2 min^{-1}, *Tin* = 10 ºC, *Ta* = 80 ºC.

It is worth to remember that this is a non-linear model. Therefore, for simplicity, we chose to expand the variable *x* into a fourth-order polynomial: New_*x* = [1, x, x^2, x^3, x^4]. Thus, it was possible to adjust five parameters with linear regression, and to fit them to the curve: T-hat (x) = $\alpha + \beta x + \gamma x^2 + \rho x^3 + \varphi x^4$.

For a comparison between the methodologies, a loop of ten thousand repetitions was made, generating ten thousand different data sets and ten thousand different models for each method. For each interaction, the models were compared using the metrics previously mentioned (MAPE, RMSE, R^2), as well as the analysis of the incidence of discrepant predicted points.

Results and Discussions of Case Study Heating Tank

In this case k = 2 was used to generate clusters (the data was divided into 2 clusters). This value was calculated through the silhouette coefficient, which obtained the best result with the value S_2 = 0.78. Then, y-rank was applied to the generated data and also individually to each cluster (our proposed methodology). An example of the selection of data by the two methods can be seen in Figure 5. The blue dots represent the data selected for testing and the red dots for training. Some observations and forecasts can already be made based on the selections of the data. Y-rank splits the data according to values of '*y'*, in ascending order. It can be seen in Figure 5 (left) that it selected the first left point for testing, making a serious mistake. By excluding the first left point from training, it requires the model to extrapolate when testing with those two data, which is not recommended for empirical models.

However, the selection made by k-rank was more robust. The algorithm was forced to select the first and last data points of each cluster for calibration to avoid extrapolations.

Then, models calibrated with both methodologies training sets were tested with their respective test sets. The best y-rank model's results are shown in Figure 6, and the results for the best k-rank model are shown in Figure 7.

As a final graphical comparison, Figure 8 shows the adherence of each of the fitted models to the complete original dataset.

From the fitted curves, it is possible to perceive the failure of the model calibrated by directly applying y-rank to the initial dataset. The main region of error is

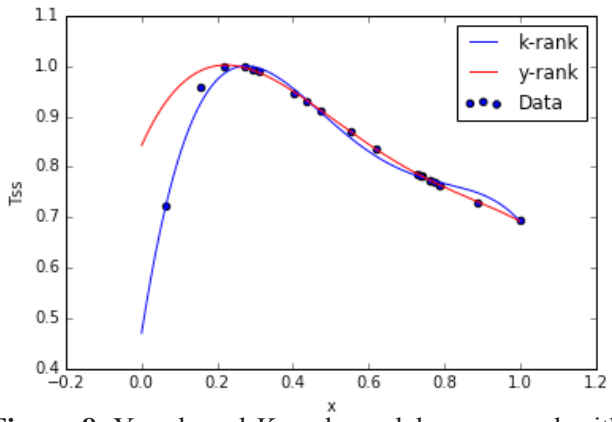

Figure 8. Y-rank and K-rank models compared with the original dataset.

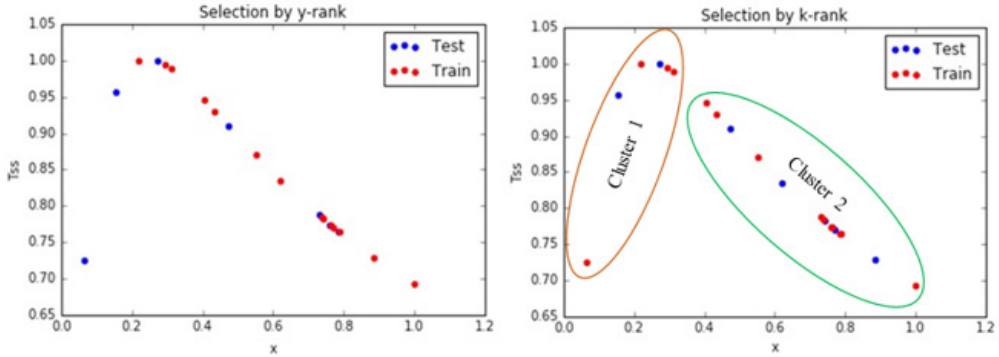

Figure 5. Data selected by y-rank (left) and k-rank (right).

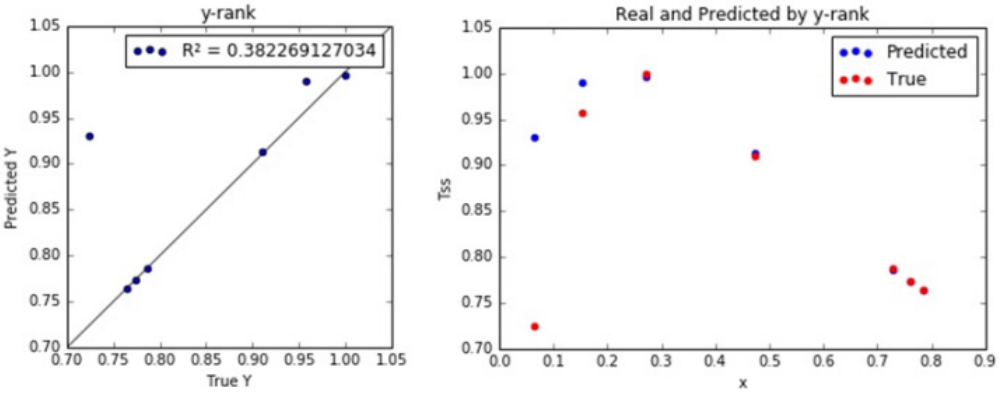

Figure 6. Best Y-rank model ($R^2 = 0.382$) and the predicted *vs* real representation.

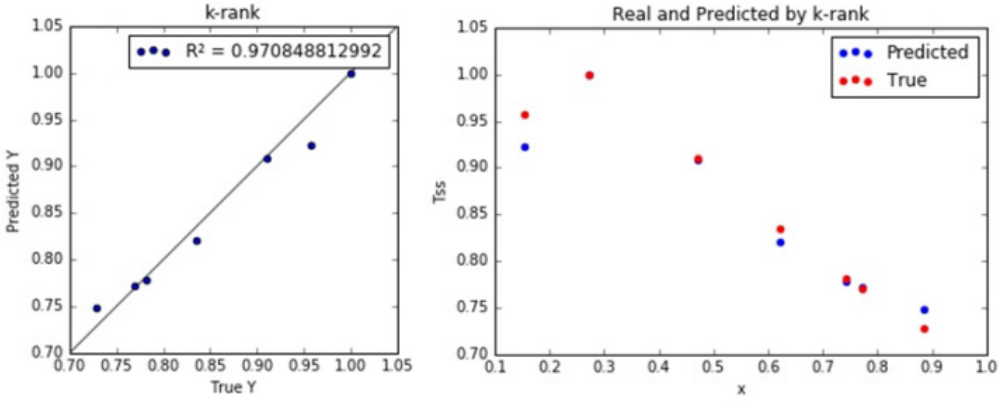

Figure 7. Best k-rank model ($R^2 = 0.971$) and the predicted *vs* real representation.

exactly located where it did not select data for training and needed to extrapolate (as expected).

Comparison between methods

To compare both methodologies, a loop of ten thousand repetitions was made. At each iteration, a new set of data was generated, split by y-rank and k-rank, and both trained models compared, according to the presented methodology. An unit was added to a counting variable for the model that obtained the best results in each of the three metrics (MAPE, R^2, RMSE) for the testing set. The model that achieved the best results in all three evaluated metrics was considered the 'winner' of that iteration. If at least one of the metrics was better in one model while the other model was superior in the other metrics, the iteration was considered a tie. Table 1 shows the results:

To evaluate the occurrence of disposable models, a maximum error of 15% was considered for each value predicted by the models. If the model estimated at least one value with a difference equal to or greater than 15% of the real value, according to Equation 6, the model was considered as disposable.

The value of 15% was selected so that models were not ruled out by the curve's - $T(x) = \alpha + \beta x + \gamma x^2 + \rho x^3 + \varphi x^4$ - inability to adhere perfectly to the data. Errors greater than 15% were notably caused by poor data selection, losing information on important regions of the curve. A loop of ten thousand repetitions was made and the disposable models were counted and presented in Table 2.

There was an incidence of approximately 12% of disposable y-rank models. This happened mainly because y-rank lost information by poorly selecting data for the model's adjustment. On the other hand, only 0.17% of the k-rank models were discarded, showing superiority in data selection when dealing with a multiplicity of solutions. Furthermore, of the 17 k-rank models discarded, 12 were also discarded with y-rank.

Table 1. Winners of the ten thousand-repetition comparison between y-rank and k-rank methods.

Total	y-rank	k-rank	Draws
10000	1764	5912	2324

SECOND CASE STUDY: PROPYLENE / PROPANE SEPARATION UNIT

A second case study is proposed to evaluate the efficacy of the method. It is based on an Aspen Plus simulation of a propane separation unit. The unit has the objective of producing a high purity (99.6%) propylene stream (C3-) from a liquefied petroleum gas (LPG) stream. The process consists of three distillation columns arranged in series, with the LPG being fed in the first column (T-01). In this column, the heavy (C4 +) compounds are removed from the bottom, while the propene-rich top stream feeds the second column (T-02). This column draws, at the top, a stream rich in ethane (C2) and, at the bottom, the stream rich in propane (C3 +) and propene (C3-) which will feed the third column (T-03). In the latter, in turn, the propene-rich chain is extracted from the top. The T-03 utilizes a heat pump, where the top stream is used as the heating fluid of the reboiler after passing through a compression step. The simplified process flow diagram is shown in Figure 9.

The constructive data of the columns were omitted due to the industrial secrecy of this information. For the stationary model of the unit, simulated in Aspen Plus version 7.2 (Figure 10), the following considerations were made: (I) Each distillation column has two steady-state degrees of freedom; (II) The unit disturbance consists of the feed stream with change in the inlet flow rate and the propene concentration, where the pressure and the temperature are controlled; (III) The top pressure of each column is held constant; (IV) The inlet temperature of each column is kept controlled; (V) The process vessels were not simulated, since they do not influence the result of the stationary simulation; (VI) The Peng-Robinson thermodynamic model was used to calculate the physicochemical properties of the currents, because the currents were composed of hydrocarbons; (VII) Load losses of the process pipes were disregarded; (VIII) The supply of each column has a constant pressure, controlled by a valve, which has been modeled to provide a specified outlet pressure; (IX) The heat exchangers, with the exception of P-05, were only modeled for the calculation of the necessary thermal exchange, disregarding the mechanical limits of the equipment; (X) The P-05 exchanger was modeled as a hull and tube with a constant global heat transfer coefficient of 932 kcal/(h.m².°C) and a total area of 2168 m²; (XI) The compressor was considered isentropic, calculating the energy required to maintain a specified discharge pressure.

The degrees of freedom used in the modeling of each column are described below:

- Column T-01: Reflux ratio (RR1) and the mass ratio between the distillate flow (stream 4) and the inlet flow (stream 1), and this new variable will be called D / F1.
- Column T-02: Reflux ratio (RR2) and the mass ratio between the backflow (current 10) and the input

Table 2. Disposable models for a loop of ten thousand repetitions - Heating Tank case study.

Total	y-rank Disposable models	k-rank Disposable models	Models discarded by both
10000	1187	17	12

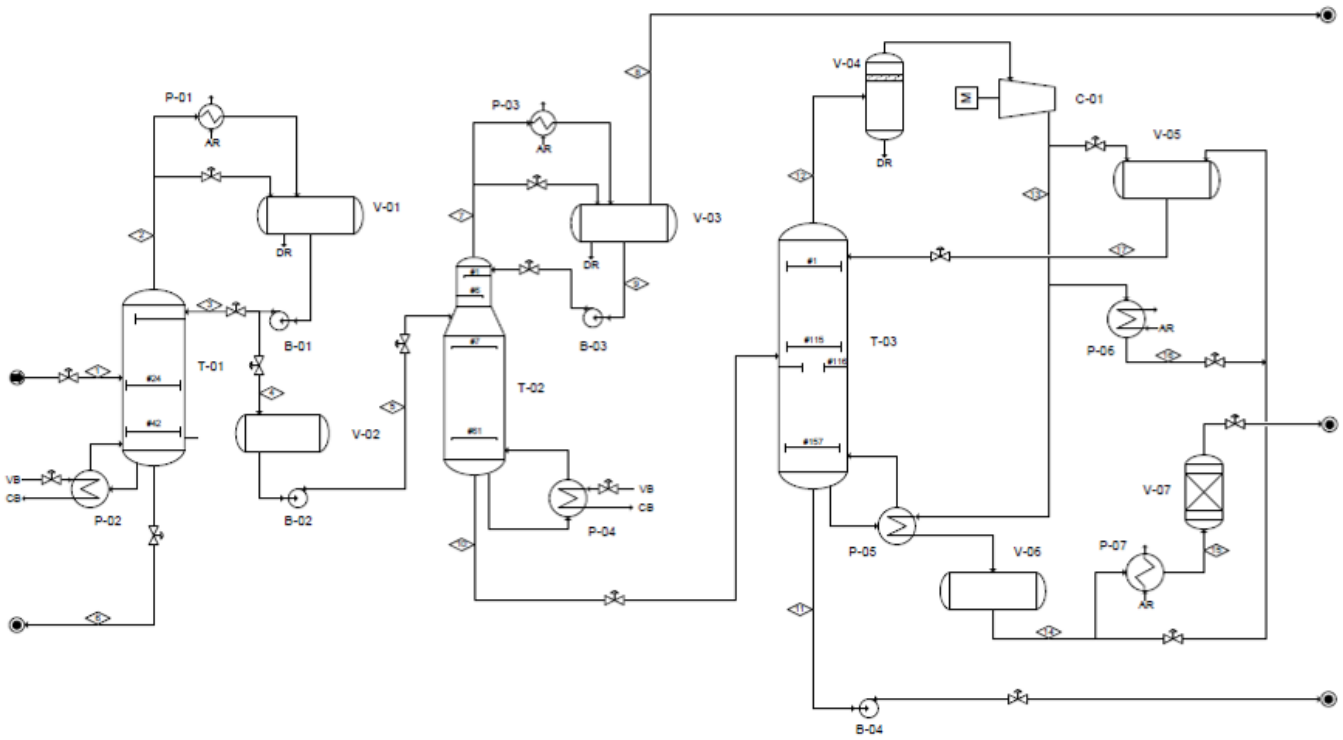

Figure 9. Process diagram of the propylene/propane separation unit. Source: Fleck et al. (2012).

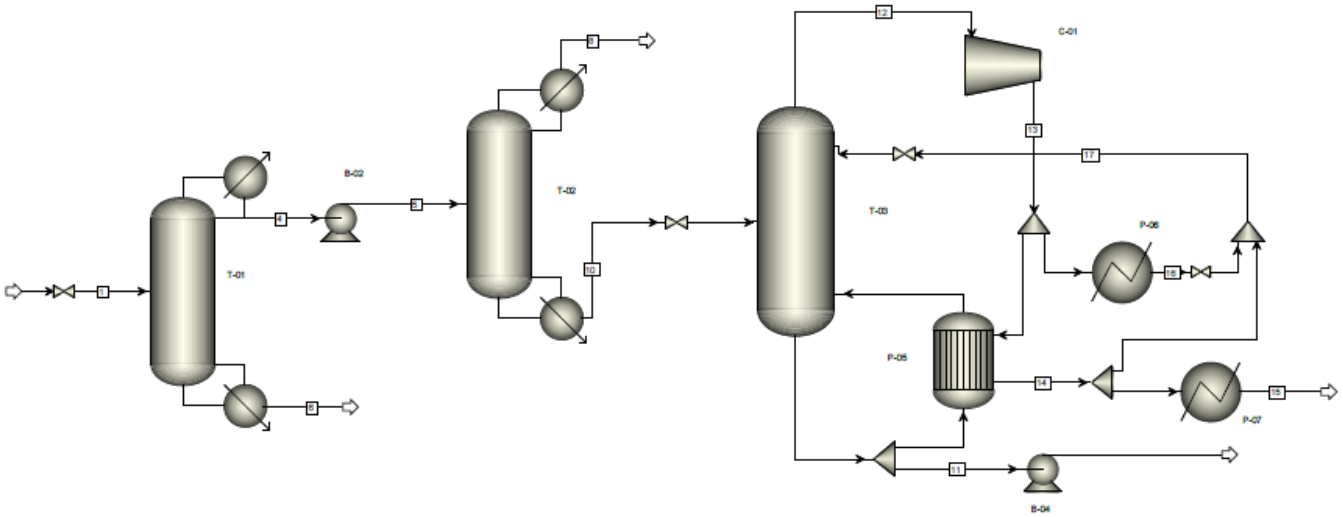

Figure 10. Model of the unit created in Aspen Plus for the propylene/propane separation. Source: Fleck et al. (2012).

flow of the column (current 5), and this new variable will be called B / F2.

- Column T-03: fraction of the current is coming out of the compressor that will be used as the heating fluid of the boiler, which will be called FA3. Also, the fraction of the stream leaving the referendum (stream 14) that returns to the column as reflux for the column, which will be called FR3, was used.

The methodology of Case Study: Propylene / Propane Separation Unit

For this study, we used simulation data from the T-01 column with the variation in the unit disturbances and the degrees of freedom of the column (RR_1 and D / F_1). As disturbances, the mass flow rate (F_1) and the mass fraction of propane (Z_{C3-1}) were considered. For the change in the molar fraction of propane, the same proportion of the nominal actual specification was kept for the other components.

Nine hundred sample points were generated for the T-01 column according to the ranges shown in Table 3.

With this data set, three scenarios will be studied: one well sampled (all 900 samples), another moderately sampled (50% of the original data – 450 samples), and a third with a reduced number of samples (2% of the original data – 18 points). The concentration of

Table 3. Variables and simulation data range for the Aspen Plus propylene/propane model.

Variable	Nominal Value	Limits	Number of points
F1	63000	59850-66150	5
ZC3-	0.355	0,319-0.390	6
RR1	1.98	1.00-4.00	6
D/F1	0.459	0.250-0.650	5

propene at the top of the T-01 column will be predicted. The available input variables, as in a practical case, are available in Table 4. These variables were previously selected according to the understanding of the process, and because they have high correlations with the output variable (the concentration of propene at the top of the column).

The idea is to compare the methodologies of data splitting and to evaluate the importance of selecting a good data set for the calibration of the model. Thus, for the evaluation of the three scenarios, linear models developed with linear regression will be fitted. Then, the comparisons will be made using the metrics already used for the previous case study (MAPE, R^2, RMSE). It is worth mentioning that the variables were normalized according to the z-normalization method, which places the data at a mean zero and unit standard deviation.

Table 4. Available variables from column T-01.

Variable	Description
D/F1	The mass ratio between distillate flow (stream 4) and inlet flow (stream 1)
RR1	Column reflux ratio T-01
Qref1	Heat exchanged in the column T-01
Qcond1	Heat exchanged in the T-01 column condenser
F4	T-01 column top-mass flow rate
T4	The temperature at the top of the T-01 column
DP1	The pressure difference in column T-01

Results and Discussions for the Propylene / Propane Separation Unit

As mentioned, three inferences will be made for the concentration of propene at the top of the T-01 column. The first will be done with the all the 900 samples available. The second with 50% of the samples (450), and a third with only 2% of the available samples (18).

Wide Sampling: 900 Samples

According to the proposed methodology, the same data set is separated in two ways: one using the k-rank methodology and the other using the y-rank methodology. Then a model is fitted through multivariate linear regression, for each methodology. For both methodologies, the train/test split chosen was 50% - 50% for calibration and 50% for testing.

For the k-rank methodology, the best number of clusters using k-means is initially measured. This is done through the silhouette coefficient, which analyzes the quality of the generated clusters. A scan was made using up to 20 clusters.

From Figure 11 we can see that the optimal number of clusters for this case would be k = 2, with a silhouette coefficient equal to 0.35 (the higher the silhouette coefficient, the better is the separation).

Two clusters were generated. Then, y-rank was applied in each cluster, to separate 50% of the data for calibration and 50% for testing. Meanwhile, conventional y-rank was also applied to the original set of samples, also separating 50% for calibration and 50% for testing. Thus, two distinct models were generated with linear regression using the input variables from Table 4. The metrics for the models can be seen in Table 5. Figure 12 shows the predicted values versus the real values for the test set of both methodologies.

Comparing the metrics for both models, it is possible to verify that both models are equivalent. On the other hand, when evaluating the prediction plot of the test set of both models, it is notable that the y-rank model adjusted better a specific region of the data (i. e. small error between 0.7 and 0.9 and large error between 0.1 and 0.6), whereas the k-rank model error was better distributed throughout the whole data.

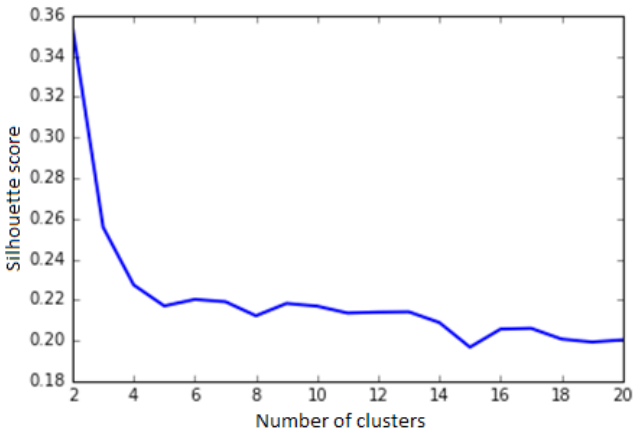

Figure 11. Silhouette coefficient for different numbers of clusters – cluster quality assessment.

Table 5. Values of the evaluation metrics for the models - ample sampling.

Method	Calibration			Test		
	R^2	RMSE	MAPE	R^2	RMSE	MAPE
k-rank	0.98	0.026	3.20 %	0.94	0.049	7.33 %
y-rank	0.97	0.034	4.75%	0.93	0.053	7.42 %

Average Sampling: 450 samples

The 450 samples were selected in such a way that the entire domain of the data was represented. The procedure was already done for the previous case (extensive sampling) was repeated. Again it was diagnosed that k = 2 is the number of clusters by

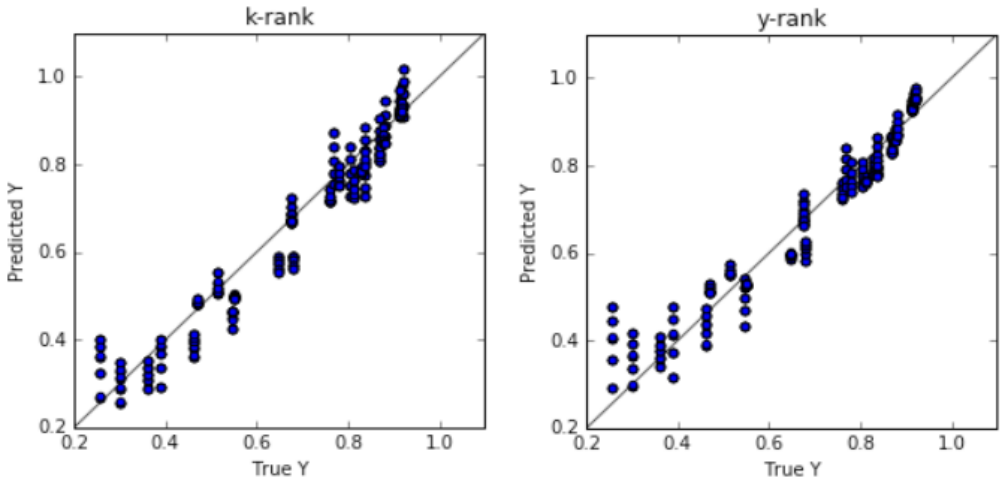

Figure 12. Test set predicted values versus real values - ample sampling.

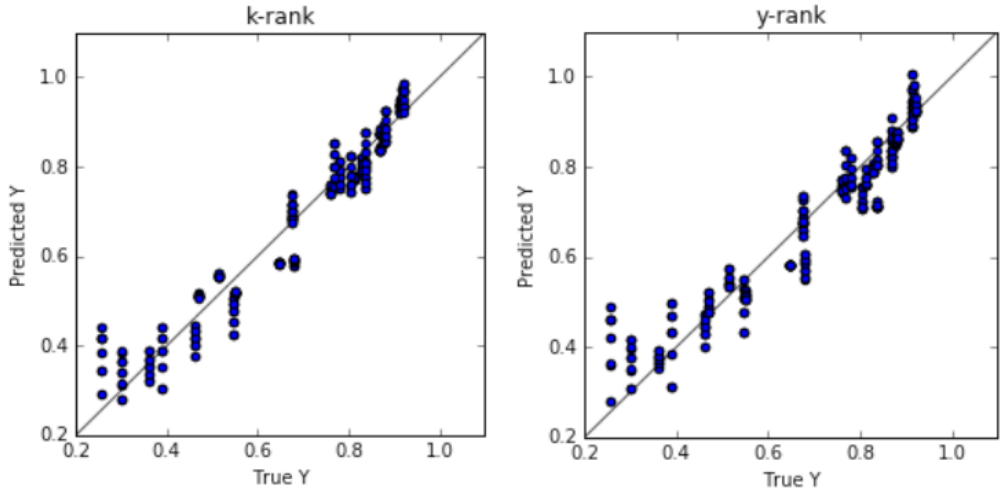

Figure 13. Test set predicted values versus real values - average sampling.

k-means, based on the silhouette analysis. The metrics for each model are shown in Table 6, and prediction plots for the test sets are illustrated in Figure 13.

Again, the generated models have very similar metrics and behaviors, as the previous case.

Table 6. Values of the evaluation metrics for the models - average sampling.

Method	Calibration			Test		
	R^2	RMSE	MAPE	R^2	RMSE	MAPE
k-rank	0.98	0.026	4.20 %	0.94	0.048	7.19 %
y-rank	0.98	0.024	3.15%	0.91	0.059	8.24 %

Minimum Sampling: 18 samples

The samples were strategically collected to represent the original data. Again, we have the highest silhouette score for k = 2. We followed the same methodology to generate models with the data selected with k-rank and y-rank. This time, 10 samples were selected for calibration and 8 samples for testing. Table 7 shows the values of the evaluation metrics for the two models. Also, Figure 14 shows the prediction plots of the test data for both models.

Table 7. The values of the evaluation metrics for the models - minimum sampling.

Method	Calibration			Test		
	R^2	RMSE	MAPE	R^2	RMSE	MAPE
k-rank	0.99	0.016	2.12 %	0.93	0.055	8.33 %
y-rank	0.98	0.024	3.53%	0.66	0.121	18.29 %

For the minimum sampling case, we can see expressive differences in the models. Although they were both able to fit well to the training data, the y-rank model had errors 3 times higher than the k-rank model in the test set. It means that the y-rank model was not able to deal satisfactorily with new unseen data, probably due to extrapolations in areas in which the y-rank did not select training data representatively.

On the other hand, the k-rank model with minimum sampling achieved similar results as the previous cases, using only 2% of the original data. The methodology was considered successful in splitting data in situations with multiplicity of solutions, especially when there is a limitation in data samples.

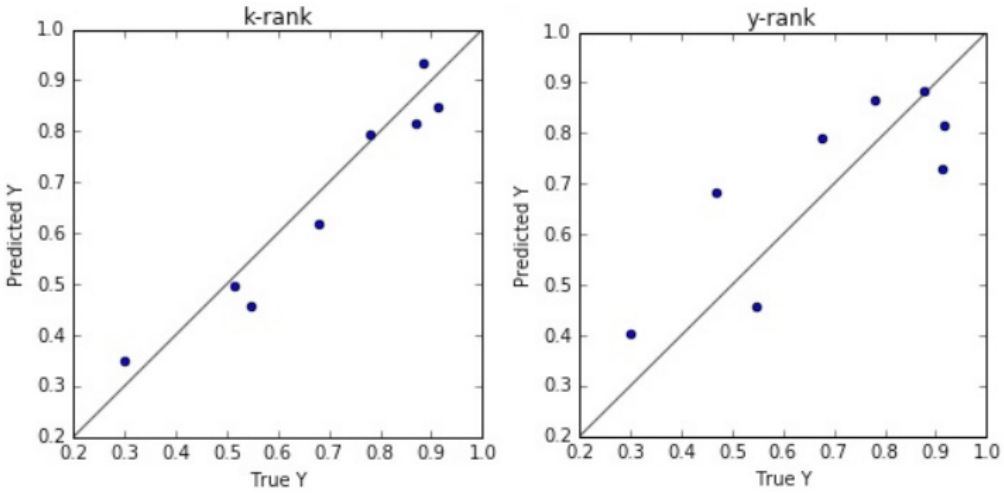

Figure 14. Test set predicted values versus real values - minimum sampling.

CONCLUSION

The results showed that y-rank applied directly to a dataset where there is multiplicity of solutions may result in loss of information, splitting training and testing sets with an unfair distribution of data. Such distribution may result in an unreliable model for certain regions.

K-rank proved efficient in first grouping the dataset into clusters and then running the y-rank in each cluster, improving the quality of data selection. It is worth mentioning that k-means can be used for multivariate functions, but it is an algorithm sensitive to the scaling of variables. Therefore, it is essential to normalize the data before applying the technique.

In the Heating Tank case study, in both comparisons studied in this work, k-rank was superior to y-rank. When comparing who obtained the best prediction for new data (test set) using three different metrics (MAPE, RMSE, R^2), the models calibrated with k-rank selection were superior in 59.12% of the cases; in 23.24% they were considered tied and only for 17.64% the y-rank achieved better results. Regarding the discarding of discrepant models, only 17 models out of 10,000 were discarded by the k-rank selection, while using y-rank 1187 were discarded.

In the Propylene / Propane case study, the importance of selecting the data for calibration and testing is clear. As the number of available samples becomes smaller, a better selection of data is required to fit the model properly. The y-rank models had errors almost 3 times higher than the k-rank models in the test subset with minimum sampling.

Considering that the splitting index selection is based on a single output *y*, the proposed methodology is scalable for systems with thousands of variables. Another interesting factor is that if there were a poorly made separation of clusters by k-means, the y-rank would still be applied in a similar way to the conventional method, meaning that our methodology, in the worst-case scenario, would achieve the same results as the traditional y-rank.

For cases with a large quantity of data and without computational limitations, the proposed methodology is not the most efficient for the construction of robust models. However, in cases where computational time must be taken into account or the amount of data is limited, especially when there is a limitation in certain regions indispensable to fit representative models, the methodology presented was efficient.

ACKNOWLEDGMENTS

The authors would like to thank the financial support by the National Petroleum, Natural gas and Biofuels Agency (ANP) and Petróleo Brasileiro S.A. – PETROBRAS.

ABBREVIATIONS

MAPE mean absolute percentage error
RMSE root mean squared error;
R^2 coefficient of determination;
SSE sum of squared error.

REFERENCES

Baliño, J. L., Modeling and Simulation of Severe Slugging in Air-Water Systems Including Inertial Effects. Journal of Computational Science, 5(3), 482-495 (2014). https://doi.org/10.1016/j.jocs.2013.08.006

Boullosa, D., Larrabe, J. L., Lopez, A., and Gomez, M. A., Monitoring Through T2 Hotelling of Cylinder Lubrication Process of Marine Diesel Engine. Applied Thermal Engineering, 110, 32-38 (2017). https://doi.org/10.1016/j.applthermaleng.2016.08.062

Browne, M. W., Cross-Validation Methods. Journal of Mathematical Psychology, 44(1), 108-132 (2000). https://doi.org/10.1006/jmps.1999.1279

Celisse, A.; and Robin, S., Nonparametric Density Estimation by Exact Leave-P-Out Cross-Validation. Computational Statistics & Data Analysis, 52(5), 2350-2368 (2008). https://doi.org/10.1016/j.csda.2007.10.002

Chi, Q., Fei, Z., Zhao, Z., Zhao, L., and Liang, J., A Model Predictive Control Approach with Relevant Identification in Dynamic PLS Framework. Control Engineering Practice, 22, 181-193 (2014). https://doi.org/10.1016/j.conengprac.2013.02.010

FABI. The Standard Toolbox for Matlab. Departament of Analytical Chemistry and Pharmaceutical Technology, Vrije Universiteit Brussel (1997).

Fleck, T. D., New Methodology for Data Inference Development. M.Sc. Thesis, Federal University of Rio Grande do Sul (2012).

Gautschi, W., Some Remarks on Systematic Sampling. Ann. Math. Statist, 28(2), 385-394 (1957). https://doi.org/10.1214/aoms/1177706966

Greene, W. H. Econometric Analysis. Upper Saddle River, N.J.: Pearson Education (2002).

Kao, F.-F.; Leu, C.-H.; and Ko, C.-H., Remainder Markov Systematic Sampling. Journal of Statistical Planning and Inference, 141(11), 3595-3604 (2011). https://doi.org/10.1016/j.jspi.2011.05.011

Kramer, O., Machine Learning for Evolution Strategies. Springer, Switzerland (2016). https://doi.org/10.1007/978-3-319-33383-0

Leu, C.-H.; and Kao, F.-F., Modified Balanced Circular Systematic Sampling. Statistics & Probability Letters, 76(4), 373-383 (2006). https://doi.org/10.1016/j.spl.2005.08.005

Massaron, L. A.; Boschetti, A. A. Regression Analysis with Python. Packt Publishing Ltd., Birmingham, UK (2016).

Ranzan, C., Strohm, A., Ranzan, L., Trierweiler, L. F., Hitzmann, B., and Trierweiler, J. O., Wheat Flour Characterization Using Nir And Spectral Filter Based on Ant Colony Optimization. Chemometrics and Intelligent Laboratory Systems, 132, 133-140 (2014). https://doi.org/10.1016/j.chemolab.2014.01.012

Raschka, S. A., Python Machine Learning. Packt Publishing Ltd., Birmingham, UK (2015).

Sampath, S.; and Uthayakumaran, N., Markov Systematic Sampling. Biometrical Journal, 40(7), 883-895 (1999). https://doi.org/10.1002/(SICI)1521-4036(199811)40:7%3C883::AID-BIMJ883%3E3.0.CO;2-4

Schultz, E. S., The Importance of Operating Point in Self-Optimizing Control Techniques. M.Sc. Thesis, Federal University of Rio Grande do Sul (2015).

Shamir, R., Maron-Katz, A., Tanay, A., Linhart, C., Steinfeld, I., Sharan, R.,Shiloh, Y. and Elkon, R., Expander--An Integrative Program Suite for Microarray Data Analysis. Bmc Bioinformatics, 6(1), 232 (2005). https://doi.org/10.1186/1471-2105-6-232

Srivastava, V., An Evaluation of Desulfurization Technologies for Sulfur Removal from Liquid Fuels. Rsc Advances, 2(3), 759-783 (2012). https://doi.org/10.1039/C1RA00309G

Storkaas, E.; and Skogestad, S., Controllability Analysis of Two-Phase Pipeline-Riser Systems at Riser Slugging Conditions. Control Engineering Practice, 15(5), 567-581 (2007). https://doi.org/10.1016/j.conengprac.2006.10.007

Thalamuthu, A., Mukhopadhyay, I., Zheng, X., and Tseng, G. C., Evaluation and Comparison of Gene Clustering Methods in Microarray Analysis. Bioinformatics, 22(19), 2405-12 (2006). https://doi.org/10.1093/bioinformatics/btl406

Wang, K.; Wang, B.; and Peng, L., Cvap: Validation for Cluster Analysis. Data Science Journal, 8, 88-93 (2009). https://doi.org/10.2481/dsj.007-020