A Novel Kernel Clustering Algorithm

Wesam M. Ashour
Islamic University of Gaza

ABSTRACT
K-means algorithm is one of the most famous clustering algorithms in data mining due to its simplicity.

Kernel K-means is an extension of K-means to cluster nonlinear separable data. However, it still has some limitations like sensitivity and convergence to the local optima. In this paper, we show how to implement a new novel kernel-clustering algorithm that is robust and converges to the global solution. We show using artificial and real data sets that the proposed kernel algorithm performs better than the standard kernel K-means algorithm.

Keywords
K-means, Kernel K-means, Clustering, global optima.

1. INTRODUCTION
Cluster analysis [1, 2] which is an unsupervised learning, is considered as one of the hot topics in data mining and machine learning. It is a process of dividing unlabeled data into groups 'clusters', so each cluster contains data that shares the same properties and differs from data of the other clusters. Clustering algorithms are used in several areas such as Marketing, bioinformatics, libraries, medicine, image processing, etc [3, 4]. While there is a vast number of clustering algorithms in the literature, there is no unique algorithm that is suitable for all different datasets. For example, current clustering algorithms do not address all requirements such as convergence to global solution, accuracy, time complexity, ability to deal with noise and outliers, scalability, usability, etc. Clustering algorithms could be divided into categories based on the type of the real datasets and applications e.g. partitional, hierarchical, and density based clustering algorithms [5].

2. RELATED WORK
K-means is considered as one of the most well known algorithm used to classify or group a set of data into K number of disjoint clusters, where K is a predefined value [6].

The operation of the iterative K-means is divided into two separated phases. The first phase is to select K initial centroids, one for each cluster. The next phase is to assign each point in data set to the closest centroid. Euclidean distance is used to determine the distance between data points and the centroids. The first step is completed when all the points are attached to some clusters. At this point, each cluster centroid is updated to be the average of data points allocated to each centriod. Once K new centroids are found, a new binding is to be created between the same data points and the closest new centroid, producing a loop. As a result of this loop, the K centroids may are repositioned in a step by step manner. The algorithm converges when the centroids do not move anymore [7].

[8] present the global K-means algorithm which is an incremental trend for clustering. In this agorithm, one cluster centroid is added dynamically at a time using a deterministic global search procedure cosisting of N executions of the K-means algorithm from suitable initial positions where N is the number os points in the data set.

[9] introduce a new algorithm based on density canopy to enhance the K- means' accuracy and stability and to optimize centroids initalization for K-means. This algorithm uses the clustering results of K-means, then, the result transformed by combining with hierarchial algorithm to find the better inital cluster centroids for K-means algorithm.

[10] introduce a new group of algorithms that solve the problem of sensitivity to initial conditions in K-means. The basic idea of these algorithms is that each centroid responds to positions of all other centroids' and to their locations with respect to the data points before they move to any new locations. As a result, it is possible for it to identify the free clusters that are not recognized by the other centroids.

S. Khan et al. [11] propose an algorithm for initial cluster centroids computation for K-means clustering based on individual attributes of the pattern, which may provide some information about initial cluster centers. The operation of the proposed algprithm is to apply the K-means for each attribute inorder to compute cluster centroids for individual attributes. This is achieved assuming that the attributes of the pattern space are normally distributed. Then the normal curve is divided into K segments, and the K-means algorithm is applied on this attribute. The previous cluster labels are allocated to every pattern, and K-means ia applied to the complete data set. Finally, a center of these classes must be found and used as centroid for K-means among the set of class labels in each pattern.

[12] enhance the K-means algorithm through using a seeding technique instead of substituting the random choosing of the centroids. There experimental results shows how their algorithm is better in term of time and accuracy.

[13, 14] develop a new K-Harmonic means algorithm which converges to a better solution than both traditional K-means or a group of experts trained using the EM algorithm. The output of this algorithm is less prone to findinga local minimum as a result of its bad initialization.

[15] develops a recursive algorithm for adaptation of fuzzy rule-based model structure using online clustering of the input-output data with a recursively calculated spatial proximity measure. The resulting evolving fuzzy rule-based models have high degree of transparency, compactform, and computational efficiency.

2.1 Kernel K-means Algorithm
Kernel K-means algorithm is an extension of the K-means clustering algorithm that identifies nonlinearly clusters. Kernel K-means maps data points to a new space called the feature space and then K-means is applied in the feature space [16].

The operation of the kernel K-means is as the same of the standard K-means, but with one difference in the calculation of distance. Kernel K-means algorithm uses kernel method instead of the Euclidean distance. The distance between each data point and the cluster centroid in the transformed space is calculated based on dot product and kernel functions e.g. polynomial, Guassian, etc. While this algorithm is able to identify the non-linear structures and it is best suited for real life data set, it has

many drawbacks as the number of cluster centers need to be predefined besides its high time complexity and convergence to the local optima [17].

Many kernel algorithms have been proposed in the recent years to improve the kernel K-means and extract clusters that are non linearly sperable [18, 19].

3. PROPOSED KERNEL CLUSTERING ALGORITHM

The performance function of K-means could be written as follows:

$$Perf_{km} = \sum_{i=1}^{n} \min_{j=1}^{k} \|x_i - p_j\|^2 \quad (1)$$

where x_i represents data points and m_j represents prototypes.

In K-means, each data point is assigned to the closest prototype, and then each prototype is updated based on this assignment. This makes both K-means and kernel K-means sensitive to the initial prototypes and lead to converge to the local optima.

In [20] we have implemented IWC clustering algorithms which is robust, insensitive to the initial conditions, and converges to the global solution. The performance function of IWC could be written as follows:

$$Perf_I = \sum_{i-1}^{n} \sum_{k=1}^{k} \frac{1}{\|x_i - p_k\|^y} \quad (2)$$

where x_i represents data points and p_k represents prototypes. y is any positive power. The idea in this performance function is to solve the problem of K-means and provides a similarity measures between each data point and all other prototypes (not only the closest one) without losing the ability to extract clusters. Before optimization process, we need each data point to respond to all prototypes, and to update each prototype iteratively based on the similarity measures between all data points and all prototypes.

We illustrate in this section how to extend IWC for kernel space and create new kernel clustering algorithm that is robust, converges to the global optima and cluster none linearly separable data.

For IWC we update prototypes iteratively using the following derived equation:

$$P_k = \frac{\sum_{i=1}^{n} r_{ik}\, x_i}{\sum_{i=1}^{n} r_{ik}} \quad (3)$$

where

$$r_{ik} = \frac{\|x_i - p_{k*}\|^2}{\|x_i - p_k\|^2} \quad (4)$$

For kernel space we need to map all data points, using none linear equations, in such a way that all none linearly separable data in original space become linearly separable in kernel space.

To update the prototypes iteratively in the kernel space we can rewrite equation (3) as follows:

$$P_k = \frac{\sum_{i=1}^{n} r'_{sk}\, f(x_i)}{\sum_{i=1}^{n} r'_{sk}} \quad (5)$$

where

$$r'_{ik} = \frac{\|f(x_i) - p_{k*}\|^2}{\|f(x_i) - p_k\|^2} \quad (6)$$

The distance between the data points and prototypes can be computed using the inner products and popular kernel functions e.g. polynomial kernel, Gaussian kernel, sigmoid kernel, etc.as follows:

$$\|\emptyset(x_i) - p_k\|^2 = \left\|\emptyset(x_i) - \frac{\sum_{s=1}^{n} r'_{sk}\, \emptyset(x_i)}{\sum_{s=1}^{n} r'_{sk}}\right\|^2$$

$$= h(x_i, x_i) - \frac{2}{\sum_{s=1}^{n} r'_{sk}} \left(\sum_{s=1}^{n} r'_{sk}\, h(x_i, x_s)\right)$$

$$+ \frac{1}{\sum_{s=1}^{n} r'_{sk}} \left(\sum_{s=1}^{n} \sum_{l=1}^{n} r'_{lk}\, r'_{sk}\, h(x_s, x_l)\right) \quad (7)$$

where $h(a, b)$ is a kernel function between points a and b.

The proposed kernel-clustering algorithm is implemented through the following steps:

1. Divide data randomly into a specified number of K clusters. The number of clusters K is an input parameter to the algorithm.
2. Initialize random weights b'_{ik} between all data points and prototypes
3. Compute the kernel matrix between all data points using one of the popular kernel functions.
4. In kernel space, compute the distances between all data points and prototypes using equation (6)
5. Update the weights between data points and prototypes using equation (5)
6. Repeat 3 and 4 until convergence.
7. Extract clusters by assigning every data point to the most similar prototype that gives the maximum weight.

4. EXPERIMENTAL RESULTS

We test and compare the improved kernel-clustering algorithm using artificial and real datasets.

4.1 Artificial data set

We have created an artificial datasets consisting of 150 data points divided into 7 clusters as shown in Figure1.

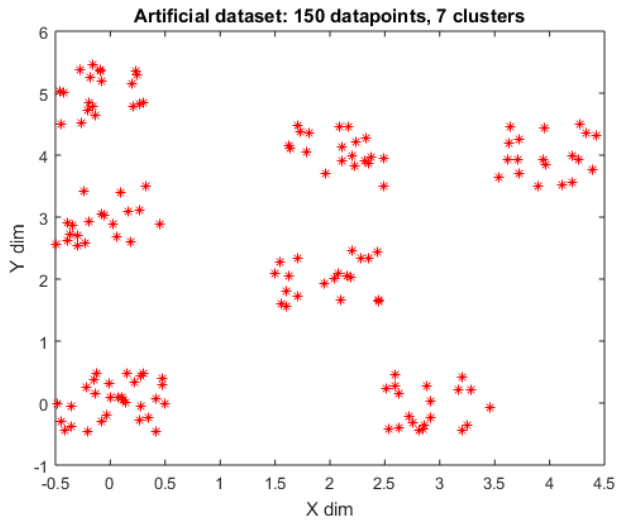
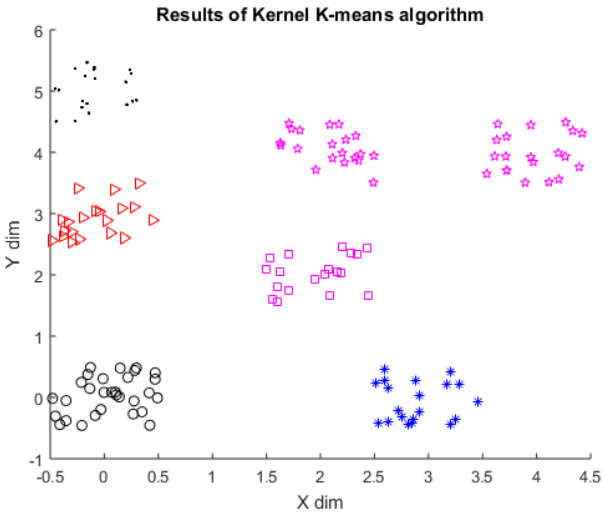


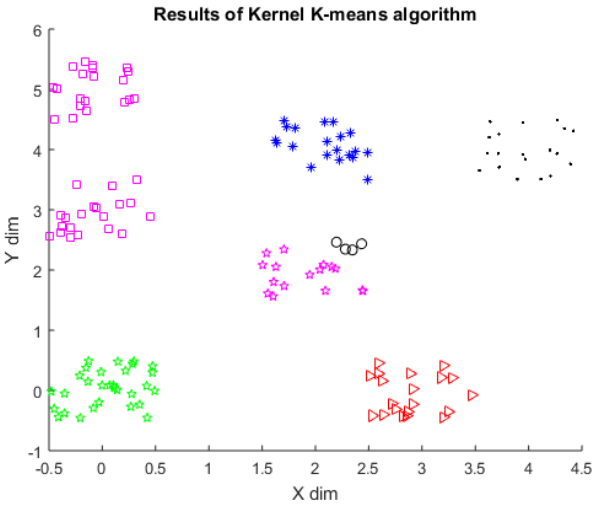
Figure 1: Artificial data set consisting of 7 clusters (150 data points)

Figure 2 (a-d) show the results after applying the Kernel K-means algorithm to the data in Figure 1 four times, each time with different initialization of the prototypes. The kernel K-means still sensitive to the initial parameters as we have different results. In addition, kernel K-means failed to extract

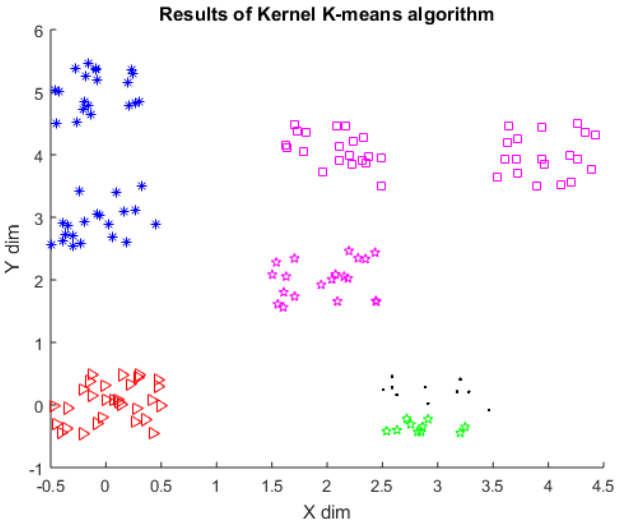
the correct clusters with some prototypes initializations. It also converged to the local optima.



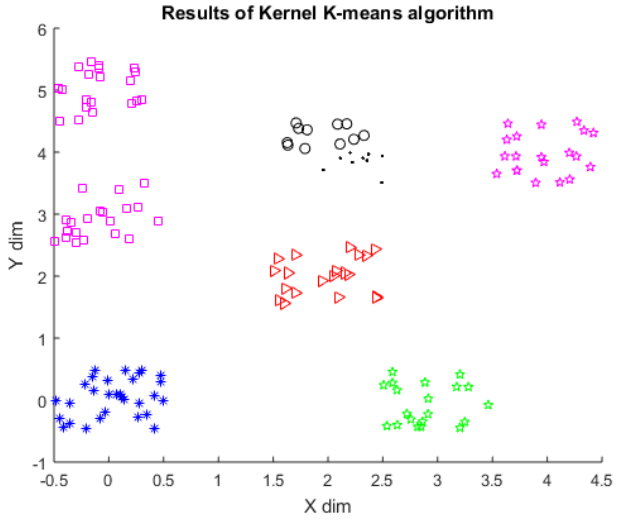
(a)



(b)



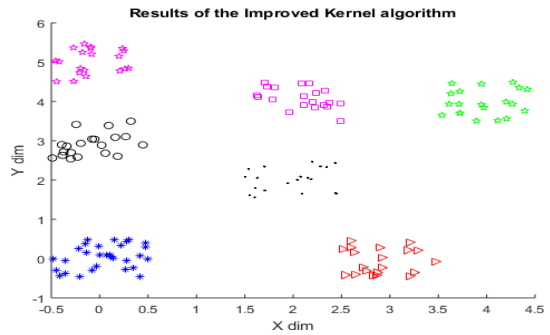
(c)



(d)

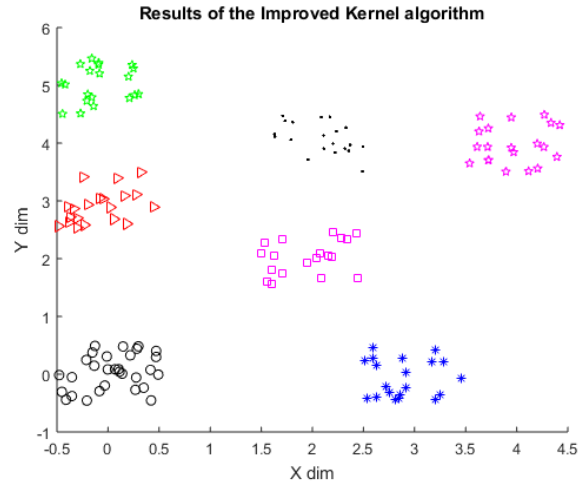
Figure 2 (a-d): Results of Kernel K-means algorithm (four times run with different prototypes initializations)

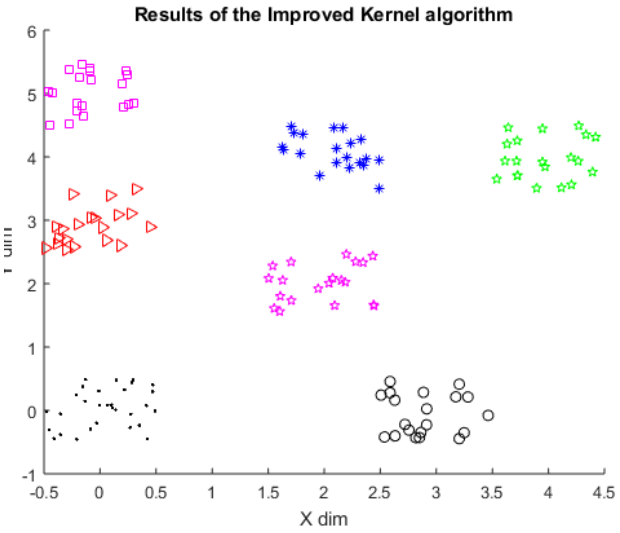
We repeat the experiment using the proposed kernel algorithm as shown in Figure 3 (a-d). The proposed algorithm succeeds to identify the clusters successfully even with poor initialization. It is robust and insensitive the initial condition.



(b)

(c)





(d)

Figure 3 (a-d): Results of the proposed Kernel algorithm (four times run with different prorotypes initializations)

4.2 Real datasets

We apply kernel k-means and the proposed kernel algorithm to the following data sets:

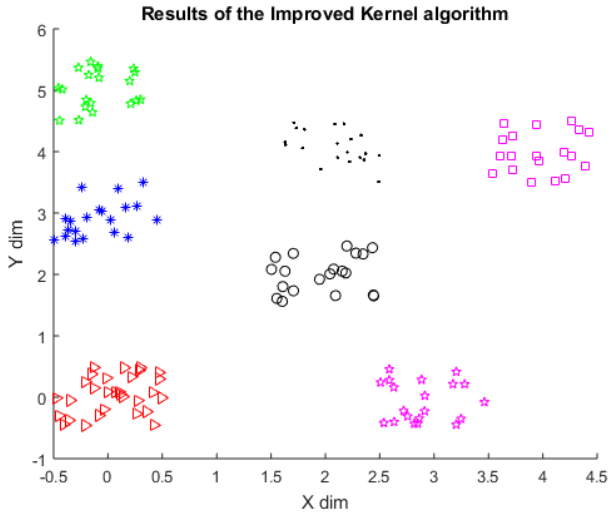
Iris dataset: 150 data points with 4 dimensions and 3 types.

Glass dataset: 214 data points with 10 dimensions and 6 types.

Algae1 dataset: 72 data points with 18 dimensions and 9 types.

Gene's dataset: 40 data points with 3036 dimensions and 3 types of bladder cancer.

In the experiment, we run each algorithm 10 times with different prototypes initializations, and compute the average of the resulted quantization errors. In addition, as another



measurement, we compute the number of errors in classifying the data points into the correct clusters. As shown in Table 1, we can see that the new proposed kernel algorithm outperforms the kernel K-means algorithm. It gives better quantization error and less number of errors for all real datasets.

5. CONCLUSION

In this paper, we have shown how to implement a new kernel-clustering algorithm that is insensitive to the prototypes initializations. In addition, it is robust and converges to the global optima. The proposed kernel algorithm has been evaluated and compared using artificial and real datasets. The experimental results showed that the proposed algorithm outperforms kernel K-means and gives better results. For evaluation, the measurements quantization error and the number of errors in classifying data points were used.

Table1: The results of applying Kernel K-means and the Proposed Kernel algorithm to the real datasets: Iris, Glass, Algae and Gene. Each row shows the quantization error and the number of errors after applying the algorithms 10 times. The average is computed at last column.

			1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**
IRIS	Kernel K-means	QE	124	110	97	115	99	123	97	112	121	99	109.7
		errors	50	16	19	50	19	16	17	50	16	19	27.2
	Proposed Kernel	QE	97	97	97	97	97	97	97	97	97	97	97
		errors	17	17	17	17	17	17	17	17	17	17	17
Glass	Kernel K-means	QE	1987	1990	1976	1990	1980	5761	1987	1976	5761	1987	2739.5
		errors	33	27	25	23	75	27	75	23	33	25	36.6
	Proposed	QE	1973	1973	1973	1973	1973	1973	1973	1973	1973	1973	1973

	Kernel	errors	21	21	19	21	19	19	21	19	21	19	20
Algae	Kernel K-means	QE	13	11	13	14	12	12	13	14	11	11	12.4
		errors	14	15	20	19	14	22	14	20	15	20	17.3
	Proposed Kernel	QE	9	9	10	9	10	10	10	9	10	9	9.5
		errors	10	14	10	10	11	14	14	9	9	11	11.2
Gene	Kernel K-means	QE	1190	1215	1220	1190	1213	1217	1195	1223	1198	1215	1207.6
		errors	6	14	9	6	14	11	14	9	6	9	9.8
	Proposed Kernel	QE	1187	1187	1187	1187	1187	1187	1187	1187	1187	1187	1187
		errors	6	6	6	6	6	6	6	6	6	6	6

6. REFERENCES

[1] Jain, A.K., M.N. Murty and P.J. Flynn, 1999. Data clustering: A review. ACM Comput. Surv., 31: 264-323.

[2] Xindong, Wu and et. al 2008. Top 10 Algorithms in Data Mining. Journal of Knowledge and Information Systems, 14(1):1-37, DOI: 10.1007/s10115-007-0114-2.

[3] Plant, C. ; Zherdin, A. ; Sorg, C. ; Meyer-Baese, A. ; Wohlschlager, A.M., 2014, Mining Interaction Patterns among Brain Regions by Clustering, IEEE Transactions on Knowledge and Data Engineering, 26(9): 2237-2249, DOI: 10.1109/TKDE.2013.61.

[4] Shuo Chen and Chengjun Liu, 2014, Clustering-Based Discriminant Analysis for Eye Detection, IEEE Transactions on Image Processing, 23(4):1629-1638, DOI: 10.1109/TIP.2013.2294548.

[5] Guojun Gan, Chaoqun Ma, and Jianhong Wu, 2007. Data Clustering: Theory, Algorithms, and Applications, ISBN: 978-0-898716-23-8, ASA-SIAM.

[6] Celebi, M., H Kingravi, and P. A. Vela, 2013, "A comparative study of efficient initialization methods for the k-means clustering algorithm." *Expert Syst. Appl*., vol. 40:200–210.

[7] Cao, F. Y., Liang, J. Y. and Jiang, G., 2009. An initialization method for the K-means algorithm using neighborhood model. Computers and Mathematics with applications, 58(3): 474-483.

[8] Likas, A., Vlassis,, M. and Verbeek, J., 2003, "The global k-means clustering algorithm," Pattern Recognition, vol. 36, pp. 451–461.

[9] Arai, Koheri and Ridho, Ali, 2007. Hierarchical K-means, an algorithm for Centroids initialization for K-means, Saga University, 36(1): 25-31.

[10] Ashour W., Fyfe, C., 2008, Local vs global interactions in clustering algorithms: advances over K-means, *International Journal of Knowledge-based and Intelligent Engineering Systems (KES)*, 12(2): 83-99, 2008. ISSN 1327-2314.

[11] Khan, S., Ahmad, A., 2004, Cluster center initialization algorithm for K-means clustering. Pattern Recognition Letters, vol. 25, pp.1293-1302.

[12] Arthur, D., and Vassilvitskii, S., 2006. K-means++: The advantages of careful seeding. *In Bay Area Theory Symposium,BATS06*. http://www.stanford.edu/~sergeiv/papers/kMeansPP-soda.pdf.

[13] Zhang, B., Hsu, M., and Dayal, U., 1999. K-harmonic means - a data clustering algorithm. Technical Report HPL-1999-124, HP Laboratories, Palo Alto.

[14] B. Zhang, 2001. Generalised k-harmonic means- dynamic weighting of data in unsupervised learning. In *First SIAM International Conference on Data Mining*. http://www.siam.org/meetings/sdm01/pdf/sdm01_06.pdf.

[15] Angelov, P., 2004. An approach for fuzzy rule-base adaptation using on-line clustering. International Journal of Approximate Reasoning, 35(3):275–289.

[16] Dhillon, S., Guan, Y., and Kulis, B, 2004. Kernel k-means, spectral clustering and normalized cuts. In *Proc. ACM SIGKDD Intl Conf. Knowledge Discovery and Data Mining, Seattle, W.*

[17] Girolami, M.., 2002, Mercer kernel based clustering in feature space *IEEE Transactions on Neural Networks*, 13(3):780- 784.

[18] Burges C. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

[19] Suykens, J. A. and J. Vandewalle, J. 1999. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300.

[20] Ashour, W., Wu, Y. and Fyfe, C, 2009. Non-standard parameter adaptation for exploratory data analysis, Springer.