Unsupervised Learning-Based Stock Keeping Units Segmentation

Ilya Jackson[1(✉)], Aleksandrs Avdeikins[1,2], and Jurijs Tolujevs[1]

[1] Transport and Telecommunication Institute, Lomonosova Iela 1, Riga, Latvia
{jackson.i,tolujevs.j}@tsi.lv
[2] Trialto Latvia LTD, "Dominante", Kekavas p., Kekavas n. 2123, Latvia

Abstract. This paper reports on the unsupervised learning approach for solving stock keeping units segmentation problem. The dataset under consideration contains 2279 observations with 9 features. Since the "ground truth" is not known, the research aims to compare such clustering algorithms as *K*-means, mean-shift and DBSCAN based only on the internal evaluation, thus, this research may be considered as descriptive cluster analysis. Besides that, several preprocessing techniques are utilized in order to improve the result.

Keywords: Clustering · Inventory segmentation · Inventory clustering · Data mining · Unsupervised machine learning · Principal component analysis

1 Introduction

An average inventory system contains immense number of stock keeping units (SKUs). In general case, it is computationally impossible to consider each item individually and manage it under individual inventory policy. As far back as late 80s, an essentially important question has arisen: "how to aggregate stock units into groups so that the resulting inventory policies are sufficiently close to those policies that would have been generated if every unit was treated individually?" [1].

Nowadays the development of the efficient methodology for defining SKU's groups is still relevant. In previous research we managed to develop a metaheuristic technique for finding a nearly-optimal reorder policy in stochastic multiproduct inventory systems [2]. However, dimensionality of real-world problems requires a segmentation of an assortment in such a way that each segment is relatively homogeneous and may be treated under a common inventory policy. Thus, it becomes an extremely tempting opportunity to take advantage on the state-of-the-art unsupervised machine learning approaches in order to tackle this long-standing problem.

Early attempts to group SKUs by cluster analysis may be tracked back to the study of Srinivasan and Moon [3]. The researchers introduced a hierarchical clustering-based methodology for supporting inventory systems in supply chains. Several heuristics were applied to identify the relationships between items and take into account product features with a significant impact on supply chain. The Calinski-Harabasz index was used to validate the result of the average linkage clustering.

I. Kabashkin et al. (Eds.): RelStat 2018, LNNS 68, pp. 603–612, 2019.
https://doi.org/10.1007/978-3-030-12450-2_58

In 2007 the *K*-means-based SKU segmentation methodology was proposed [4]. The research aimed to reduce the time required to compute the inventory-control parameters in large-scale multi-echelon inventory system. The segmentation methodology was tested on different multi-echelon inventory systems in order to understand an effect of resulting penalty costs. Three years later the similar *k*-means-based approach by Egas and Masel was applied to determine storage assignments [5]. The paper concludes with the statement that the method managed to reduce the number of aisles to retrieve orders by 20–30% compared to a demand-based assignment strategy.

It is also important to emphasize the recent paper [6]. The authors conducted a study on the application of a constrained clustering method reinforced with principal component analysis. According to the research, the proposed method is able to provide significant compactness among item clusters.

This study discusses an application of various clustering algorithms to solve the SKU-aggregation problem. Since the "ground truth" is not known, such algorithms as *K*-means, mean-shift and DBSCAN are compared based only on the internal evaluation. In this regard, this study may be considered as a descriptive clustering analysis. The research utilizes dataset provided by the "Trialto Latvia LTD", the third-party logistics operator. Since SKU's groups should take into account all attributes with a sufficient impact on the certain inventory operation, considered features include information beyond the inventory cost and volume that are used in classical ABC analysis. Besides, the work pays special attention to feature-scaling, anomaly-detection and validation approaches. In order to perform all necessary calculations and data transformations we wrote a script in Python 3.6 [7]. The script uses a free machine learning library "scikit-learn" [8] as a core complementing it with several algorithms.

2 Dataset Description

The initial dataset consists of 2279 SKUs (observations) with 9 features. Selected features include only numerical data and comprise a lot of information beyond that utilized by a classical ABC analysis (Table 1).

Table 1. Selected features.

ID	Unit price (EUR)	Expire date (days)	Total outbound (units)	Number of outbound orders	Pallet weight (kg)	Pallet height (cm)	Units per pallet
1	0.058	547	2441	9	105.6	1.56	1920
2	0.954	547	0	0	207.68	1.00	384
3	2.385	547	23	12	165.78	1.02	108
…	…	…	…	…	…	…	…
2278	2.02	730	710	354	322.56	1.19	288
2279	1.99	730	765	363	322.56	1.19	288

All the features have an undeniable impact on the inventory management and constitute two core groups: handling-related and turnover-related. Such features as expire date, pallet weight, pallet height and number of units per pallet determine the speed and subtlety of handling. On the other hand, total outbound and number of outbound orders indicate how tradable a particular SKU is. The total outbound and the number of outbound orders is represented as different attributes despite the fact of sharing some mutual information. It is done on purpose, since both the demand size and the demand frequency are important for the research. It is also worth to note that the feature "number of outbound orders" is calculated based on arisen demand from 2017-02-06 to 2018-02-13 (537,791 orders in total).

3 Methodology

3.1 Data Preprocessing

The first relevant problem we noticed was the excessive presence of missing data. Such features as "unit price", "pallet gross weight", "pallet height" and "units per pallet" contain 710(31.1%), 371(16.3%), 787(34.5%) and 295(12.9%) missing values respectively. Moreover, 1070(47%) observations include at list one missing value and all four features are missing in 208(9.1%) observations. Since each observation stands for a real tradable product, we cannot afford to drop such precious information. Since such naive methods for treating missing data as replacement by the feature mean or may insert a bias [9], we have decided to impute missing values using the nearest neighbor imputation (NNI). NNI, justifying its name, imputes missing values using values calculated from the k nearest neighbors. The nearest neighbors are found by minimizing a distance function, in our case the Euclidean distance [10].

According to Chen and Shao, NNI has two pivotal benefits [11]. Firstly, the method provides asymptotically unbiased and consistent estimators for population means. Secondly, the NNI method is expected to be more robust against model violations than methods based on parametric models, such as ratio imputation and regression imputation. An important parameter for the NNI method is the value of k. In this study we used k of 10 which is suggested by Batista and Monard [12].

Since anomaly detection and clustering algorithms incorporated to this research utilize distances between points as a metric, attribute scaling becomes a necessary prerequisite. This research applies Z-score attribute standardization relying on the research by Mohamad and Usman, which concludes with the statement that the application of Z-score standardization prior to K-means clustering leads to more accurate result compared to decimal scaling and min-max normalization [13].

In light of the fact that clustering algorithms, which use distances between points as a metric, such as k-means, are extremely sensitive to presence of anomalies in dataset, it is crucial for quality of cluster analysis to identify and remove such anomalies. For this purpose, the local outlier factor (LOF) was applied. Despite the fact that LOF is a relatively old algorithm, Campos, et al. conclude with the statement that even after 16 years of active research LOF remain the state of the art, especially in datasets with possibly larger amounts of outliers [14].

The distinguishing feature of LOF is utilization of reachability distance as an additional measure. Let k-distance (A) be the distance of the A to the k-th nearest neighbor. Based on that the local reachability density (lrd) is defined as follows:

$$lrd(A) = \frac{|N_k(A)|}{\sum_{B \in N_k(A)} \max\{k - distance(B), d(A,B)\}}, \tag{1}$$

where $N_k(A)$ is the set of k nearest neighbors. Eventually the local reachability densities are compared with those of the neighbors in order to calculate LOF:

$$LOF(A) = \frac{\sum_{B \in N_k(A)} \frac{lrd(B)}{lrd(A)}}{|N_k(A)|}. \tag{2}$$

The LOF should be interpreted the following way, a value around 1 or less indicates an inliner, while values significantly larger than 1 clearly indicate an outlier.

3.2 Cluster Analysis

In this research we take advantage on the pivotal mechanics of mean-shift clustering to discover the number of "clots" and potential clusters respectively. In fact, mean-shift is a centroid-based algorithm, which works by iteratively sorting out candidates for centroids to be the mean of the points in the considered region. It starts with an initial estimate x. Then the kernel function $K(x_i - x)$ calculates the weight of nearby observations for further reestimation of the mean. After that such candidate-centroids are filtered in order to eliminate duplicates forming the final set of centroids. The weighted mean of the density determined by K is Eq. (3) [15].

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) x_i}{\sum_{x_i \in N(x)} K(x_i - x)}, \tag{3}$$

where $N(x)$ is the neighborhood of x. The algorithm iteratively assigns $m(x)$ to x repeating the estimation until $m(x)$ converges. That is important to note that the flat kernel was used in the research Eq. (4).

$$K(x) = \begin{cases} 1, & if\ x \leq 1 \\ 0, & if\ x > 1 \end{cases}. \tag{4}$$

The key advantage of mean-shift is application-independence and clustering without assumptions on predefined shape of clusters.

The number of discovered "clots" is further used as an initial value of parameter k in k-Means clustering algorithm; one of the oldest and, nevertheless, commonly used methods for partitioning the observations. The algorithm attempts to clusters data separating a set of data observations into k clusters minimizing the Euclidean distance-based objective function. It repeatedly proceeds two pivotal steps, namely assignment each data point to the cluster with closest centroid and recalculation of the centroids as

the mean of all the observations in that cluster until the algorithm converges forming the Voronoi diagram.

Since *k*-means cannot find non-convex clusters, DBSCAN (density-based spatial clustering of applications with noise) is also incorporated to the research for an extra insurance. DBSCAN is a density-based data clustering algorithm that finds a number of clusters by estimation of density distribution of corresponding observations. Unlike *k*-means, DBSCAN determines clusters based on the concepts of "ε-neighborhood" and "density reachability" viewing clusters as areas of high density separated by areas of low density [16].

3.3 Clustering Validation

Since the ground truth is not known, abovementioned algorithms are compared using internal cluster validation tests. Namely, silhouette index, Calinski-Harabasz index and Dunn index were calculated in order to compare clustering results based on such properties of clusters as density, shape and separability.

Silhouette validity index refers to a method of internal clustering consistency validation. The index reflects cohesion and separation of clustered data and may be defined as follows Eq. (5).

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{5}$$

where $a(i)$ is the average distance between an observation i and all the other observations, $b(i)$ is the lowest average distance between i and observations in clusters of which i does not belong to [17]. The values of index are bounded in range $-1 \leq s(i) \leq 1$. The score is generally higher for dense and well separated clusters, which corresponds to a concept of a cluster.

Calinski-Harabasz (*CH*-index), also known as pseudo *F*-statistics, is a traditional and computationally fast method for internal clustering validation. For k clusters the *CH*-index $s(k)$ is defined as the ratio of the between-clusters dispersion mean and the within-cluster dispersion:

$$s(k) = \frac{tr(B_k)}{tr(W_k)} \times \frac{N - k}{k - 1}, \tag{6}$$

where $tr(B_k)$ is the trace of the between-clusters dispersion matrix and $tr(W_k)$ is the trace of within-cluster dispersion matrix [18]. As a side note, the value of CH-index is generally higher for convex clusters which make it biased towards DBSCAN and other density-based clustering techniques.

Dunn index (*DI*) measures a compactness and well-separation of clusters. Namely, the score is higher if clustering produces a small variance between observations within one cluster keeping mean values of different clusters sufficiently far apart:

$$DI = \min_{i=1,\ldots,n_c} \left\{ \min_{j=i+1,\ldots,n_c} \left(\frac{d(c_i, c_j)}{\max_{l=1,\ldots,n_c} diam(c_k)} \right) \right\}, \tag{7}$$

where $d(c_i, c_j) = \min_{x \in c_i, y \in c_j} d(x, y)$ stands for the intercluster distance between clusters c_i and c_j, $d(x, y)$ is the Euclidean distance between observations x and y, $diam(c) = \max_{x,y \in c} d(x, y)$ is the diameter of a cluster [19]. The *DI* values lie in the interval from 0 to infinity such that the higher values correspond to better clustering.

3.4 Dimensionality Reduction

In this research PCA gave hope to improve the current result for two main reasons. Firstly, some attributes are interrelated and correlate with each other sharing some mutual information (Table 2). Secondly, it is quite natural for a real-world data to contain some portion of noise, for instance, due to the measurement error. Additionally, some noise may be inserted during NNI procedure. Based on that PCA could be useful as a noise-reduction tool.

Table 2. Pearson correlation coefficient of attributes.

	Expire date	Total outbound	Number of outbound orders	Pallet weight	Pallet height	Units per pallet
Unit price	−0.08	−0.07	−0.09	−0.09	−0.09	−0.04
Expiredate	1.00	0.08	0.07	−0.35[a]	–0.36[a]	0.04
Total outbound	–	1.00	0.86[b]	0.04	–0.04	−0.03
Number of orders	–	–	1.00	−0.04	0.01	0.00
Pallet weight	–	–	–	1.00	0.28[a]	0.06
Pallet height	–	–	–	–	1.00	−0.04

[a] indicates moderate linear relationship
[b] indicates strong linear relationship

PCA may be defined as a common statistical procedure that maps the data to a new coordinate system such that the first coordinate contains the greatest variance, the second coordinate contains the second greatest variance and so on. This study incorporates PCA for two major purposes. Firstly, PCA is applied after all preprocessing and clustering for data-visualization. For this purpose, exactly two principal components are derived, such that each principal component corresponds to an axis on 2-d plot. Secondly, we tried to improve the result obtained from *k*-means relying on noise-reduction and feature-extraction properties of PCA.

4 Experiment

4.1 Data Preprocessing

Applying the NNI we varied the value of k in range from 3 to 10 and did not observe a significant difference, thus, it was concluded to use k of 10 as suggested by Batista and Monard [12]. As the result, all the missing values were successfully imputed. Right after that the data was transformed using Z-score standardization.

Since clustering algorithms that use distances between points as a metric, such as k-means, are extremely sensitive to presence of anomalies in dataset, the LOF algorithm was applied iteratively varying k in reasonable range from 5 to 30. What was interesting, in each run exactly 114 outliers (5.1%) have been detected. Roughly speaking, an outlier is an observation that deviates too much from other observations as to arouse suspicion that it is generated by a completely different mechanism. However, for inventory management outliers are the most precious pieces of data. In this particular case identified outliers stand for small SKUs with very high unit price, SKUs with high total demand, but low demand frequency, SKUs with extremely high or extremely low pallet weight and so on. These findings were reported to the representative of the "Trialto Latvia LTD", however, they were also dropped from the dataset prior to cluster analysis for the purpose of clustering validity. As the result 2165 observation remained.

Right after that mean-shift with a flat kernel is applied to estimate the number of blobs and potential clusters respectively. The algorithm detected 26 potential clusters, however, the vast majority of these clusters (18) consists of 3 or less observations. These observations are, in fact, outliers that were able to survive the anomaly detection with LOF. We trimmed 35 new-found outliers and repeated the procedure iteratively. In the second iteration 28 more outliers were removed and 10 potential clusters were discovered with no single cluster containing less than 6 observations. Eventually, 63 additional anomalies missed by LOF (2.9%) were detected and trimmed. As the result 2102 observation remain in the dataset.

4.2 Cluster Analysis

10 clusters defined by mean-shift (Fig. 1) were a starting point. Mean-shift served the research well discovering existing blobs and potential outliers, however, the cluster analysis by itself is quite poor, according to all the indexes (Table 2).

DBSCAN showed even worse results due to the fact that the dataset contained quite convex clusters with large differences in densities. This also implies the problem that DBSCAN considers a significant portion of the observations as noise. For instance, the algorithm with ε (local radius for expanding clusters) of 0.747 defined 10 clusters classifying 173 observations (8.2%) as noise (Fig. 1). Since each observation stands for a product and we were already forced to trim 177 outliers, we cannot afford to lose any more precious data. Besides that, the clustering result is generally very poor based on all the indexers.

Since k-means clustering requires some prior guessing about the number of clusters [20], we take key advantage on mean-shift assuming that the number of found "clots" corresponds to a nearly-optimal value of k parameter in k-means clustering. In our case

this guess turned out to be correct, namely *k*-means with k of 10 results the best segmentation based on the silhouette index and DI, holding one of the largest value of the CH-index at the same time. It may also be observed that k of 9 demonstrates a quite promising result (Fig. 2).

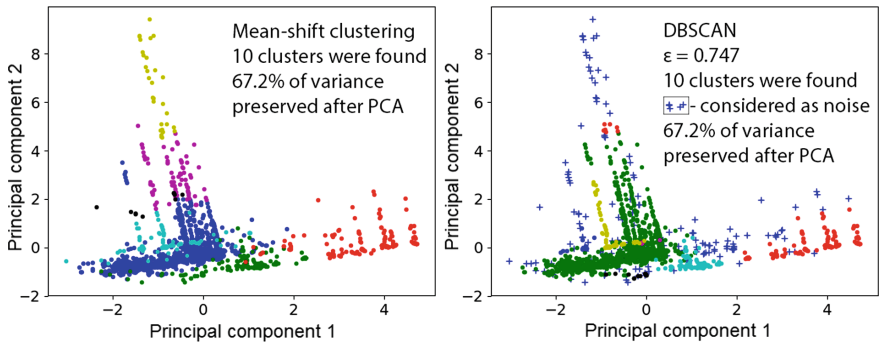

Fig. 1. Cluster analysis by mean-shift and DBSCAN visualized via PCA.

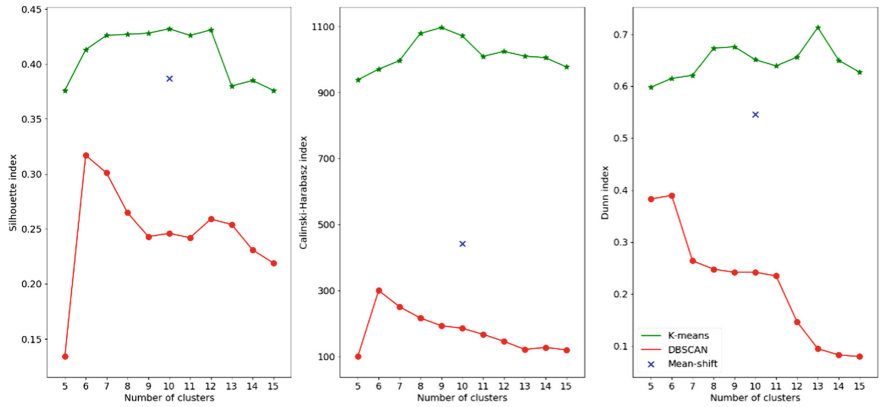

Fig. 2. Values of validity indexes depending on the number of clusters.

4.3 Further Improvements via PRINCIPAL Component Analysis

Eventually, the incorporation of PCA was indeed a right decision, since gradually lowering the number of principal components to 4 we managed to improve the initial result on 11% according to silhouette validity index preserving more than 95% of variance (Table 3, Fig. 3). It is also important to emphasize that such indexes as *CH* and *DI* rose even more. However, these figures are less representative due to excessive sensitivity of such indexes to dimensionality.

Table 3. 10-means after PCA.

Number of components	Silhouette	*CH*	*DI*	Variance preserved (%)
Prior to PCA	0.432	1072.0	0.639	100
6	0.441	1120.1	0.662	99.10
5	0.452	1163.5	0.667	97.82
4	**0.475**	**1368.3**	**0.750**	**95.11**
3	0.456	2173.1	0.746	82.09
2	0.454	3641.7	0.581	67.21

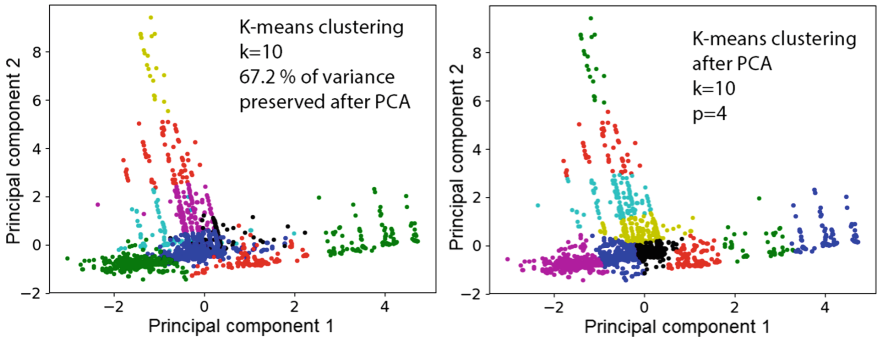

Fig. 3. *K*-means clustering prior to and after PCA with 4 principal components.

5 Conclusion

In conclusion, it is worth to mention that with the combination of PCA and *k*-means we managed to achieve decent SKUs' segmentation, according to internal validity tests. On the other hand, due to the fact that the dataset comprises quite convex clusters with large differences in densities, DBSCAN was quite inefficient.

Taking into account the fact that the cluster analysis incorporated features with undeniable impact on the inventory management beyond that utilized by a classical ABC approach, each cluster is homogeneous enough to be treated under a common inventory policy. Thus, the proposed methodology is expected to be efficient for real-world inventory control problems of high dimensionality.

References

1. Ernst, R., Cohen, M.A.: Operations related groups (ORGs): a clustering procedure for production/inventory systems. J. Oper. Manage. **9**(4), 574–598 (1990)
2. Jackson, I., Tolujevs, J., Reggelin, T.: The combination of discrete-event simulation and genetic algorithm for solving the stochastic multi-product inventory optimization problem. Transp. Telecommun. J. **19**(3), 233–243 (2018)

3. Srinivasan, M., Moon, Y.B.: A comprehensive clustering algorithm for strategic analysis of supply chain networks. Comput. Ind. Eng. **36**(3), 615–633 (1999)
4. Egas, C., Masel, D.: Determining warehouse storage location assignments using clustering analysis. In: 11th IMHRC Proceedings on Progress in Material Handling Research, Milwaukee, Wisconsin, USA, pp. 22–33 (2010)
5. Desai, V.L.: Evaluating Clustering methods for multi-echelon (r, Q) policy setting. In: Proceedings of the IIE Annual Conference, p. 352. Institute of Industrial and Systems Engineers (2007)
6. Yang, C.L., Nguyen, T.P.Q.: Constrained clustering method for class-based storage location assignment in warehouse. Ind. Manage. Data Syst. **116**(4), 667–689 (2016)
7. GitHub repository "pumpy". https://github.com/Jackil1993/pumpy. Accessed 10 July 2018
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**(1), 2825–2830 (2011)
9. Zloba, E., Yatskiv, I.: Statistical methods of reproducing of missing data. Comput. Model. New Technol. **6**(1), 51–61 (2002)
10. Jonsson, P., Wohlin, C.: An evaluation of k-nearest neighbour imputation using likert data. In: 10th International Symposium on Software Metrics, pp. 108–118 (2004)
11. Chen, J., Shao, J.: Jackknife variance estimation for nearest-neighbor imputation. J. Am. Stat. Assoc. **96**(453), 260–269 (2001)
12. Batista, G.E., Monard, M.C.: A study of K-nearest neighbour as an imputation method. HIS **87**, 251–260 (2002)
13. Mohamad, I.B., Usman, D.: Standardization and its effects on K-means clustering algorithm. Res. J. Appl. Sci. Eng. Technol. **6**(17), 3299–3303 (2013)
14. Campos, G.O., Zimek, A., Sander, J., Campello, R.J., Micenková, B., Schubert, E., Assent, I., Houle, M.E.: On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Min. Knowl. Disc. **30**(4), 891–927 (2016)
15. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
16. Tran, T.N., Drab, K., Daszykowski, M.: Revised DBSCAN algorithm to cluster data with dense adjacent clusters. Chemometr. Intell. Lab. Syst. **120**, 92–96 (2013)
17. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
18. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Commun. Stat. -Theory Methods **3**(1), 1–27 (1974)
19. Yatskiv, I., Gusarova, L.: The methods of cluster analysis results validation. Transp. Telecommun. **6**(1), 19–26 (2005)
20. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell. **17**(8), 790–799 (1995)